Peter Wriggers
Fadi Aldakheel
Blaž Hudobivnik

Virtual Element Methods in Engineering Sciences

Springer

Virtual Element Methods in Engineering Sciences

Peter Wriggers · Fadi Aldakheel · Blaž Hudobivnik

Virtual Element Methods in Engineering Sciences

Springer

Peter Wriggers
Institute for Continuum Mechanics
Leibniz University Hannover
Garbsen, Germany

Blaž Hudobivnik
Institute of Continuum Mechanics
Leibniz University Hannover
Garbsen, Germany

Fadi Aldakheel
Institute of Mechanics and Computational Mechanics
Leibniz University Hannover
Hannover, Germany

ISBN 978-3-031-39254-2 ISBN 978-3-031-39255-9 (eBook)
https://doi.org/10.1007/978-3-031-39255-9

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Paper in this product is recyclable.

Preface

The idea to write this book, related to engineering applications of the new virtual element method, occurred to us when the lock-down during the Coronavirus pandemic started. There was suddenly time to concentrate at home on an aggregation of work that we had done together with my group and colleagues. It all started in 2016 after having heard several inspiring talks by Franco Brezzi, his wife Donatella Marini and Lourenco Beirão da Veiga on the new virtual element method. It was immediately clear that this methodology might have advantages for numerical solution schemes in different applications, especially in the nonlinear range.

In 2016, we had first discussions on the basis of the virtual element method and how to implement it with Daya Reddy who stayed at the institute with an award from the Alexander von Humboldt Foundation. The idea was to use virtual elements for contact with the advantage that even for non-matching meshes a node to node formulation of contact was possible. Together with Wilhelm Rust we derived a contact discretization and algorithm based on virtual elements, first for frictionless and later for frictional contact in large strain applications. A work to be continued with our colleagues from Italy, Lourenco Beirão da Veiga and Edoardo Artioli for curved virtual elements. A contribution to three-dimensional contact is due to Mertcan Cihan who has developed in his dissertation the complex projection procedures needed in contact discretizations for three-dimensions when using virtual elements. Also Alfredo Gay Neto from the University of Sao Paulo, a former Humboldt fellow in our institute, contributed to contact formulations within the virtual element method by integrating three-dimensional virtual elements for finite elastic strains as single flexible particles into his discrete element code.

With Daya we tried to understand finite strain problems in 2017 and got some new ideas about the stabilization of the method for low order approximations. After that we worked on a quite general implementation that could be applied using two- and three-dimensional meshes with virtual elements of arbitrary shape. With these tools at hand it was only a small step to apply the new virtual element scheme also to finite strain plasticity problems and extend the application range to anisotropic materials which resulted in joint work with Jörg Schröder. The virtual element method was further extended to large strain dynamical problems by Mertcan Cihan who applied his developments to vibration problems in elasticity and impact problems undergoing finite elasto-plastic deformations.

Fracture mechanics is another application where the virtual element method offers some advantages. To cover different possible approaches, we tried the phase field method for brittle and ductile fracture with Fadi Aldakheel. But also a new cutting scheme, based on linear fracture mechanics, was introduced within the virtual element method in the Ph.D. thesis of Ali Hussein. Additionally, phase field (for fracture detection) and the cutting scheme were put together in this thesis with adaptivity and provided a new efficient tool for crack propagation problems.

When Maria Laura de Bellis came to our institute as a Humboldt fellow, we continued our work on the virtual element method in the area of damage mechanics. Here virtual elements were used in a non-local form. Additionally, Laura worked on a serendipity formulation of virtual elements for finite strain problems.

With another Humboldian, Michele Marino, we tackled homogenization problems where the vitual element method has big advantages due to the possibility to define elements with arbitrary shape. Here, metals and ceramics are described using a direct discretization of a real microstructure by one virtual element per grains with non-convex polyhedral shape. This provides a very efficient tool for homogenization, especially since only averages have to be computed. This is also demonstrated in the Ph.D. work by Christoph Böhm for crystalline microstructures of steel and magnetoelectro-mechanical materials.

Lately, we applied the virtual element method to Kirchhoff plates where C^1-continuous ansatz functions are needed. Based on the pioneering work of Franco Brezzi and Donatella Marini we were able to construct together with Olivier Allix, who visited our institute as a Humboldt laureate, virtual plate elements that provide stable solutions for thin plates with isotropic and anisotropic material.

This book is a combined effort of the three of us to allow graduate students and engineers working in industry to familiarize themselves with the new virtual element method. Due to that examples are included in the text for deeper understanding of the formulations and associated algorithms. Additionally, a simplified code that allows to solve two-dimensional elastic problems with the software tools *AceGen* and *AceFEM* was created and can be found in the shared library *AceShare* of *AceGen*.

We are very grateful to all Ph.D. students, post-docs and colleagues mentioned above with whom we had extremely fruitful discussions and collaborations. But we also have to mention and thank Jože Korelc, Professor in Ljubljana, who provided his symbolic software tool *AceGen* and the related analysis code *AceFEM* and changed it during the last five years considerably in order to accommodate the need for more flexibility when using a discretization scheme like the virtual element method with arbitrary and different number of nodes within elements and meshes.

Last but not least, we like to thank Volker Meine for drawing many of the figures in the book.

Hannover, Germany
March 2023

Peter Wriggers
Fadi Aldakheel
Blaž Hudobivnik

Contents

Chapter 1
Introduction

Many different approaches exist for the approximate solution of problems involving partial differential equations. These include finite difference schemes, finite elements, finite volume techniques, boundary elements, and particle methods, just to name some of the most prominent discretization techniques. Within these methods there exist various significant developments. As an example, in the finite element method, see e.g. Hughes (1987), Zienkiewicz and Taylor (2000), Wriggers (2008) and Oñate (2009), mixed, enhanced and reduced order methods and discretization schemes were designed for different types of partial differential equations using low and high order ansatz spaces. A relative new variant is connected to isogeometric approaches with high order of continuity, see Cottrell et al. (2009). Research on numerical solution methods continues to be motivated by the goal of developing stable, efficient and robust discretization schemes for finite deformation applications in solid mechanics.

While the finite element method is well established as a tool to handle problems in nonlinear solid mechanics, see for example the texts by Bathe (1996), Belytschko et al. (2000), Wriggers (2008) and the encyclopedia (Stein et al. 2004), it is nevertheless of interest to explore new methods that have potential advantages such as flexibility with regard to mesh generation and choice of element shapes. The discontinuous Galerkin method provides flexibility in the sense that meshing does not have to take into account hanging nodes: this can be an advantage in adaptive mesh procedures, see the works by ten Eyck and Lew (2006) and Noels and Radovitzky (2006) for applications to nonlinear elasticity.

There is a wide literature on polygonal and polyhedral elements, with applications to several important fields in engineering since there are situations in which it is advantageous not to be restricted in the choice of element shapes. In this regard there exist interesting developments in the use of polygonal or polyhedral elements. Representative work in this area can be found in Kuznetsov and Repin (2003), Sukumar (2004), Sukumar and Malsch (2006), Martin et al. (2008), Milbradt and Pick (2008), Bishop (2014) and Manzini et al. (2014); for finite deformation problems including contact, see Biabanaki and Khoei (2012), Biabanaki et al. (2014) and

P. Wriggers et al., *Virtual Element Methods in Engineering Sciences*,
https://doi.org/10.1007/978-3-031-39255-9_1

Chi et al. (2015). This list is not complete, but underlines the importance and need to use elements with more general shapes.

Finite difference methods are seldom applied in solid mechanics and associated engineering applications. However, they have a long history in physics, magnetodynamics and computational fluid dynamics. A generalization, the mimetic finite differences, allows discretizations with general decompositions and thus attracted a lot of attention. Some recent papers were contributed by Brezzi et al. (2005, 2009), Beirão da Veiga and Manzini (2008), Beirão da Veiga et al. (2011, 2013c), Beirão da Veiga (2010) and Droniou et al. (2010). However this methodology is not based on the framework of the Galerkin method, like the finite element method.

1.1 History and Recent Developments of Virtual Elements

An evolution of the mimetic finite difference methods led to the virtual element methods (VEM). Virtual elements maintain the generality related to element shapes which can have arbitrary geometrical forms. Thus virtual elements allow very complex meshes within a Galerkin type approximation, see Fig. 1.1. Meshes like this open up a high flexibility for the discretization and also the design of algorithms for low and higher order ansatz spaces.

The virtual element method was introduced in the seminal work Beirão da Veiga et al. (2013b). Some early contributions in the area of mathematical analysis and

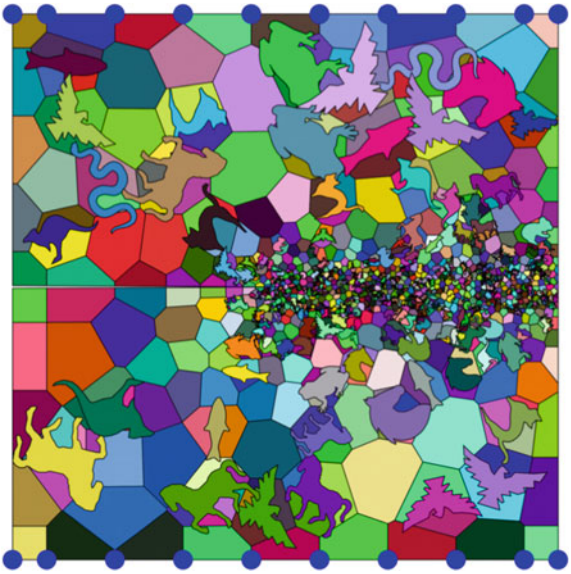


Fig. 1.1 A virtual element mesh with arbitrary shaped objects, from Aldakheel et al. (2019)

engineering can be found in Ahmad et al. (2013), Beirão da Veiga et al. (2013a, 2014), Brezzi and Marini (2013), Gain (2013) and Gain et al. (2014).

The above mentioned contributions describe the mathematical background of the recently virtual element method. The method permits the use of polygonal elements for problems in two dimensions and polyhedral elements in three dimensions. Furthermore, there is no need for a restriction to convex elements, nor is it necessary to avoid degeneracies such as element sides having an interior angle close to π radians, see the mesh in Fig. 1.1, which consists of elements of different shapes from convex to non-convex forms. All kind of shapes are possible for a discretization using virtual elements, e.g. all animals are just single virtual elements. Despite this variety of different element geometries the mesh provides a continuous C^0 discretization. Thus the method permits the direct use of Voronoi meshing tools, and as an example, crystals can be represented by single elements in a polycrystalline material.

Key examples of the method in elasticity can be found in Beirão da Veiga et al. (2013a), Gain et al. (2014) and Artioli et al. (2017). Extensions to contact and fracture mechanics can be found in Wriggers et al. (2016), Benedetto et al. (2018) and Aldakheel et al. (2018). Kirchhoff plates with C^1-continuous ansatz spaces were discussed in Brezzi and Marini (2013) and Chinosi and Marini (2016) while Reissner–Mindlin plates were treated in e.g. Beirão da Veiga et al. (2019). Applications to engineering problems, where solids undergo finite strains, are presented in Beirão da Veiga et al. (2015), Chi et al. (2017), Wriggers et al. (2017), Wriggers and Hudobivnik (2017), Hudobivnik et al. (2018) and De Bellis et al. (2019). Optimization problems were treated as well by employing the virtual element method. Developments in this direction are due to Antonietti et al. (2017), Chi et al. (2020) and Zhang et al. (2020). Applications of the virtual element method, besides solid mechanics, can be found in the area of Navier–Stokes flow, see e.g. Beirão da Veiga et al. (2018), of parabolic problems in Vacca and Beirão da Veiga (2015) and of magneto statics in Beirão da Veiga et al. (2018), to name just a few.

Despite being less than a decade under development, the range of virtual elements in engineering has been quickly widened to a bunch of applications. The following cases highlight some of the advantages of the virtual element method when applied to specific problems:

- Many applications in engineering include a combination of different materials or different layers. The mathematical modeling leads to interfaces between the layers or materials. In a discretization, parts related to one material or layer might be meshed in various ways and with dissimilar methods. This leads to non-matching meshes at the interface, see Fig. 1.2a. Virtual elements allow arbitrary number of nodes and thus coupling can be done in a continuous manner, even for non-matching meshes. It was shown that such coupling fulfills the patch test and thus yields a stable discretization scheme.

  It is of course possible to couple the virtual element discretization in a continuous way when the finite element method has the same ansatz order, see Fig. 1.2b. The reason for this is that the shape functions of the finite element method are matched at the boundary by the virtual element method. Thus the virtual element method

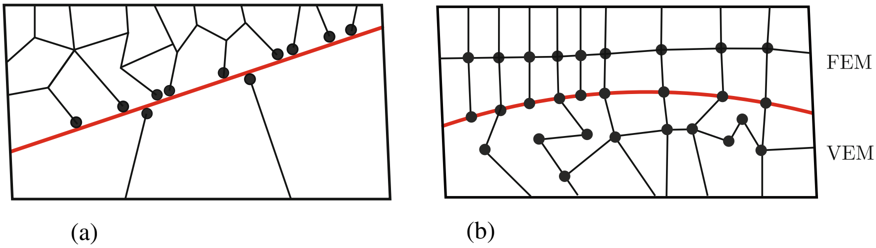


Fig. 1.2 Interface: **a** non-matching mesh, **b** coupling of virtual and finite element meshes

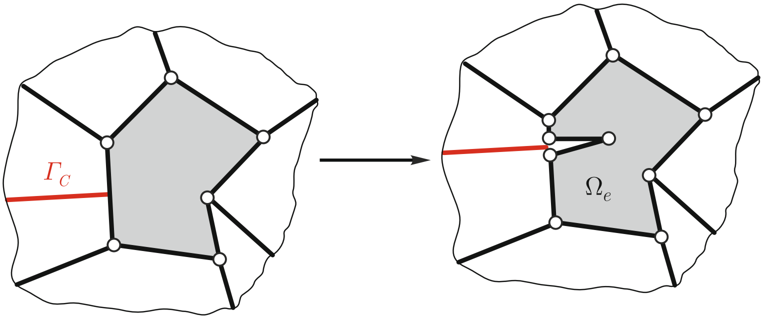


Fig. 1.3 Insertion of a crack within a virtual element

can be employed in combination with the finite element method where virtual elements have an advantage with respect to geometry and flexibility while finite elements are employed for the remaining discretization. More details regarding the ansatz functions for the virtual element method are provided in Chap. 3.

- In crack propagation problems, it is possible to insert directly a crack into a virtual element, see Fig. 1.3. The crack starts at the ending of the previous crack (in red) on the left side and proceeds to the middle of the element, see right side of the figure. This insertion produces a virtual element that has additional vertices (in this case three). Such adding of nodes is not permitted in the finite element method but does not provide any problem within the virtual element technology. The idea and algorithmic aspects of this method is presented in Sect. 10.2, see also Hussein et al. (2019) for the treatment of crack-propagation for 2D elastic solids at small strains. As depicted in Fig. 1.4 it is also possible to split virtual elements (here Ω_1) into two elements while maintaining the mesh structure of the existing discretization. This cutting technique for virtual elements can be combined with a phase field approach, see Aldakheel et al. (2018) and Hussein et al. (2019, 2020).
- Discretizations of contact problems lead generally to non-matching meshes at the interface, see e.g. Fig. 1.5a. Due to that, numerous investigations and formulations were designed within finite element methodologies. Among them are node-to-segment, segment-to-segment or mortar methods, see e.g. early works of Hallquist (1984), Wriggers and Simo (1985), Simo et al. (1985), Belgacem et al. (1997) and

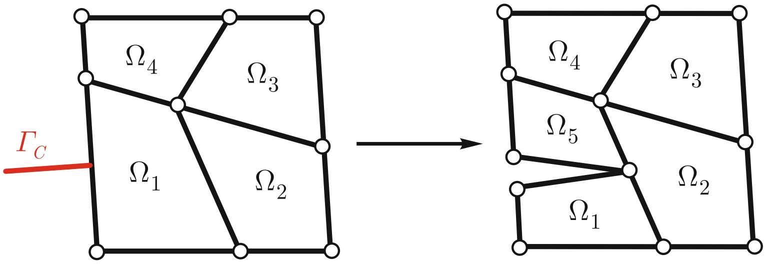


Fig. 1.4 Splitting of a virtual element

Wohlmuth (2000). Node-to-segment approaches are simple, but usually do not fulfill the contact patch test. Mortar methods fulfill the contact patch test, but have a high complexity with regard to coding.

The virtual element allows to have arbitrary number of nodes within an element. This feature can be exploited to introduce additional nodes in a contact area, see Fig. 1.5b. Here nodes $\hat{k}$, $\hat{i}$ and $\widehat{k+1}$ are inserted depending on a projection of nodes k, i and $k+1$ onto the opposite surface, respectively. Hence a very simple node-to-node contact formulation is achieved, even for non-matching meshes. Node-to-node contact simplifies on one hand the coding and on the other hand fulfills the contact patch test. Applications of the virtual element method in the area of contact mechanics can be found in Wriggers et al. (2016) for small, in Wriggers and Rust (2019) for large strains, and in Aldakheel et al. (2020) for virtual elements with curved boundaries. More details related to the use of virtual elements in contact mechanics are provided in Chap. 11.

- An advantage of the virtual element methods comes into play when microstructures of certain materials have to be modeled. This is the case in homogenization problems of crystalline anisotropic materials or polycrystals in metal plasticity. As an example, Fig. 1.6 shows cuts through polycrystalline microstructures. The left microstructure is modeled directly by a finite element mesh inside each of the grains, see Fig. 1.6a. Such discretization leads to a huge number of elements and thus needs considerable computing power for its numerical solution. Contrary, in Fig. 1.6b, a three-dimensional polycrystal is discretized using virtual elements, see e.g. Marino et al. (2019) and Böhm et al. (2021). Here only one virtual element per grain is needed, having arbitrary number of nodes and faces. This approach reduces the number of elements drastically and with this the total number of unknowns. Meshes for polycrystals can either be directly obtained from CT-scans, see Proudhon et al. (2018), or can be generated using special Voronoi tessellation techniques, see e.g. Falco et al. (2017). Details of the related homogenization approach are provided in Chap. 12 and illustrate the efficiency of a discretization using virtual elements.
- Fourth order partial differential equations describe e.g. Kirchhoff plate bending and gradient elasticity. C^1-continuous ansatz functions have to be introduced for such problems. The development of associated functions—especially for finite element

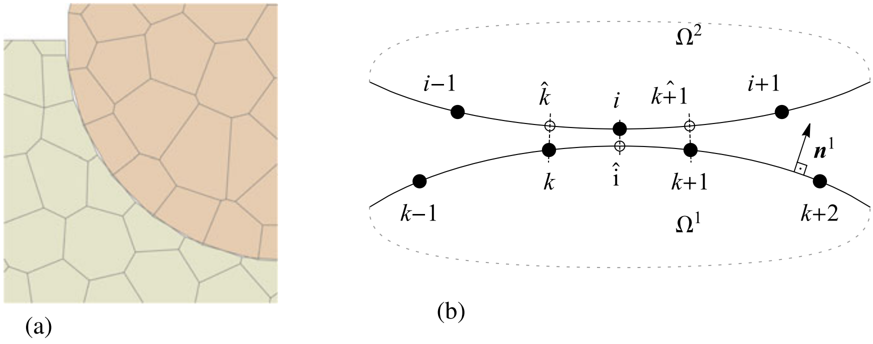
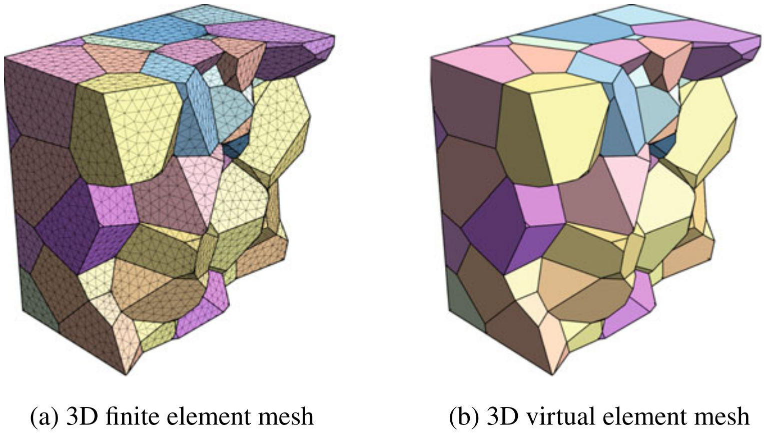


Fig. 1.5 Contact interface: **a** non-matching mesh, **b** insertion of additional nodes


(a) 3D finite element mesh

(b) 3D virtual element mesh

Fig. 1.6 Polycrystalline microstructures

analysis—is nontrivial and often leads to degrees of freedom that have no direct physical interpretation, see e.g. Argyris et al. (1968) and Bell (1969). The virtual element method which only defines the approximation functions at the element edges allows a straightforward and simple construction of even C^n-continuous ansatz functions, see e. g. Brezzi and Marini (2013), Chinosi and Marini (2016) and Wriggers et al. (2021). Further details are provided in Chap. 13.

All these applications demonstrate the high flexibility of the virtual element method in terms of discretization and algorithmic treatment for special applications like contact or crack propagation.

In general, the structure of the VEM typically comprises a term in the weak formulation or energy functional in which the quantity u_h being sought is replaced by its projection $u_\pi = \Pi u_h$ onto a polynomial space. Thus the shape functions of the virtual element method cannot be formulated directly, unlike in the finite element method. They depend via the projection on the nodal variables at the vertices along with some moments. Those have to be computed for each of the elements individually since the number of vertices for an element varies in a discretization. Details will be discussed in Chap. 3. Insertion of this projected ansatz space into the weak form results in a rank-deficient structure for arbitrary virtual elements. Hence it is necessary to add a stabilization term to the formulation. The stabilization term is generally a function of the difference $u_h - u_\pi$ between the original variable and its projection. In order to adhere to a fundamental aspect of the virtual element method in which all integrations take place on element boundaries, the stabilization term proposed, for example, in Beirão da Veiga et al. (2013a, b) takes the form of a sum of a function of nodal variables. Chi et al. (2017) adopted this approach and modified it for nonlinear problems such that the scalar stabilization parameter depends on the deformation. Another stabilization for the virtual element method uses a different technique which approximates the difference $u_h - u_\pi$ by an internal mesh, see Wriggers et al. (2017) for the two-dimensional case and Hudobivnik et al. (2018) for three-dimensional problems, see also Chap. 5. Other stabilization procedures for the virtual element method that are using hour-glass stabilization, well known from the work of Belytschko et al. (1984), are described in Cangiani et al. (2015) for linear Poisson problems. Lately some new enhanced scheme have been introduced to overcome stabilization issues, see e.g. D' Altri et al. (2021) and Chen and Sukumar (2023).

1.2 Introductory Examples

Of course, the virtual element method is a discretization scheme for two- and three-dimensional problems and not meant for one-dimensional problems. But to convey the general idea, a virtual element formulations for a truss with linear and quadratic ansatz functions will be considered and compared to finite truss elements.

1.2.1 Virtual Element Formulation of a Truss Using a Linear Ansatz

The truss or bar is depicted in Fig. 1.7. Its behaviour is mathematically modeled by the ordinary differential equation, written here for a constant cross section A,

$$EA\, u''(x) = -f(x) \tag{1.1}$$

where $u(x)$ is the longitudinal displacement of the bar, EA the stiffness, $f(x)$ the loading along the bar and x the coordinate. This differential equation can be recast in the weak form

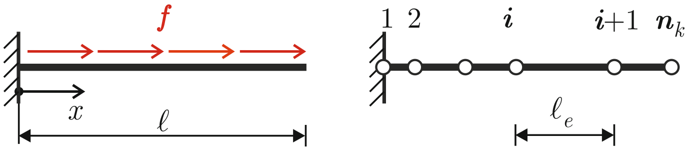


Fig. 1.7 Simple bar problem, geometry, data and discretization

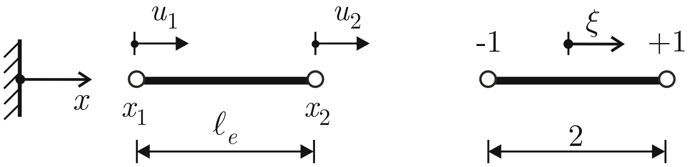


Fig. 1.8 Element in initial and reference configuration

$$\int_0^l EA\,u'(x)\,\eta'(x)\mathrm{d}x - \int_0^l f(x)\,\eta(x)\mathrm{d}x = 0\,. \tag{1.2}$$

where $\eta(x)$ is a test function which usually is discretized with the same ansatz as $u(x)$.

Equivalently a potential can be formulated

$$\frac{1}{2}\int_0^l EA[u'(x)]^2\mathrm{d}x - \int_0^l f(x)\,u(x)\mathrm{d}x \longrightarrow MIN\,. \tag{1.3}$$

To solve the bar problem (1.1) with a discretization scheme[1] the bar of length l will be subdivided in n_e elements of length l_e such that $\sum_{i=1}^{n_e} l_e = l$, see Fig. 1.7. Either the weak form (1.2) or the potential (1.3) can be starting point for a discretization scheme. Here we will employ the potential (1.3) for linear ansatz functions.

- **Finite element method.** The ansatz functions are classically defined on the basis of an isoparametric mapping such that for an element Ω_e a linear ansatz is defined in a reference space $-1 \leq \xi \leq 1$, see Fig. 1.8,

$$u_h(\xi) = \sum_{k=1}^{2} N_k(\xi)u_k \quad \text{with } N_1 = \frac{1}{2}(1-\xi) \text{ and } N_2 = \frac{1}{2}(1+\xi)\,. \tag{1.4}$$

[1] In this simple example the finite element and the virtual element method yield the same matrices and hence produce the same results. However the mathematical framework to obtain the related stiffness matrices is different.

This ansatz approximates the displacement field in the element Ω_e and thus can be inserted e.g. in (1.3) for all n_e elements leading to

$$\mathop{\mathsf{A}}_{e=1}^{n_e} \left[\frac{1}{2} \int_{-1}^{1} EA\, [u_h'(\xi)]^2\, J_e\, \mathrm{d}\xi - \int_{-1}^{1} f(\xi)\, u_h(\xi)\, J_e\, \mathrm{d}\xi \right] \longrightarrow MIN \tag{1.5}$$

where J_e is the Jacobian of the isoparametric mapping $J_e = \frac{\mathrm{d}x}{\mathrm{d}\xi} = \frac{l_e}{2}$. Both integrals can be evaluated using numerical integration. In the special case of a linear ansatz the first integral is with

$$u_h'(\xi) = \frac{\mathrm{d}u_h(\xi)}{\mathrm{d}x} = \frac{\mathrm{d}u_h(\xi)}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x} = \frac{\mathrm{d}u_h(\xi)}{\mathrm{d}\xi}\frac{2}{l_e} = \frac{u_2 - u_1}{l_e} \tag{1.6}$$

simply given by a constant term for an element e

$$U_e(u_i) = \frac{1}{2} \int_{-1}^{1} EA\, [u_h'(\xi)]^2\, J_e\, \mathrm{d}\xi = \frac{EA}{2} \left[\frac{u_2 - u_1}{l_e} \right]^2 l_e\,. \tag{1.7}$$

By introducing the vector of unknowns $\mathbf{u}_e = \langle u_1\,,\, u_2 \rangle^T$, the term in the square bracket can be written as $(u_2 - u_1)/l_e = \frac{1}{l_e}\langle -1\,,\,1\rangle \mathbf{u}_e = \mathbf{B}\,\mathbf{u}_e$ which yields the matrix form of (1.7)

$$U_e(\mathbf{u}_e) = \frac{1}{2} \int_{-1}^{1} EA\, [u_h'(\xi)]^2\, J_e\, \mathrm{d}\xi = \frac{EA}{2}\, \mathbf{u}_e^T \mathbf{B}^T \mathbf{B}\, \mathbf{u}_e\, l_e \tag{1.8}$$

and the element stiffness matrix for the linear ansatz

$$\mathbf{K}_l^{FEM} = \frac{\partial^2 U_e(\mathbf{u}_e)}{\partial \mathbf{u}_e \partial \mathbf{u}_e} = \frac{EA}{l_e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \tag{1.9}$$

The second integral in (1.5) which relates to the potential of the distributed load f can be evaluated at element level using numerical integration, like the Gauss quadrature. The number of integration point depends then on the polynomial degree of $f(\xi)$, taking into account that $u(\xi)$ is a linear function.

- **Virtual element method.** This methodology relies on an ansatz space in which the displacements are known at the nodal points and on the element edges (which do not exist in a one-dimensional problem). But the approximation u_h is *not known* within the domain Ω_e. This is totally different from the finite element method, see above, where the ansatz for the displacement field (1.4) is defined in the entire domain Ω_e.

The question is now: how can we derive an ansatz for the displacement field? The idea is to use a projection $\Pi(u_h)$ of the displacement u_h onto a polynomial space. The projected part will be called $u_\pi = \Pi(u_h)$. With this definition one can write $u_h = \Pi(u_h) + (u_h - \Pi(u_h))$ which splits the approximation u_h into the projected part $\Pi(u_h)$ and a remainder $(u_h - \Pi(u_h))$.

With a linear ansatz[2] $u_\pi = a_1 + a_2 x$ one can compute the gradient of the projected part, here the derivative $u'_\pi = a_2$, from the orthogonality condition within an element

$$\int\limits_0^{l_e} p' \, (u'_h - u'_\pi) \, \mathrm{d}x = 0 \rightarrow \int\limits_0^{l_e} p' \, u'_\pi \, \mathrm{d}x = \int\limits_0^{l_e} p' \, u'_h \, \mathrm{d}x \tag{1.10}$$

where p is a weighting function that has the same polynomial degree as the ansatz u_π. Since u_h is not known within the element, the integral on the right side of (1.10) cannot be computed. However, based on the identity $(p' \, u_h)' = p' \, u'_h + p'' \, u_h$ (integration by parts) (1.10) can be reformulated

$$\int\limits_0^{l_e} p' \, u'_\pi \, \mathrm{d}x = (p' \, u_h)\Big|_0^{l_e} - \int\limits_0^{l_e} p'' \, u_h \, \mathrm{d}x \, . \tag{1.11}$$

For the chosen linear ansatz p' is constant as well as u'_π, thus it follows that $p'' = 0$. With $u_h(0) = u_1$ and $u_h(l_e) = u_2$, see Fig. 1.8, we obtain the projected gradient from (1.11)

$$a_2 = u'_\pi = \frac{u_2 - u_1}{l_e} \tag{1.12}$$

which is now a function of the nodal displacements u_1 and u_2.

Thus it is possible to compute the gradient u'_π without knowing the function u_h inside the element Ω_e. In this simple case, the result u'_π matches exactly u'_h provided by the finite element method and hence the remainder is zero: $u'_h - u'_\pi = 0$. But in general the results are different which will be shown in the subsequent chapters and virtual elements allow a more general discretization as has been demonstrated in Sect. 1.1. Finally, u'_π can be inserted in (1.3) which provides the same result as given in (1.7). Thus the stiffness matrix of the virtual element $\mathbf{K}^{VEM}$ is exactly the same as $\mathbf{K}^{FEM}$ in (1.9).

A problem is now to compute the loading term in (1.3) since u_h is not known inside the element. A possibility is to compute an approximation of u_h by the projected displacement u_π using the linear ansatz.[3] This approximation follows from the

[2] In the standard isoparametric formulation used in the finite element method an additional coordinate system is employed in the reference configuration, see (1.4). Contrary the virtual element method is formulated directly with respect to the coordinate system in the initial configuration.

[3] Mathematically it can be shown that this approximation of u_h by u_π in the loading term is appropriate and will even result in optimal error error estimates, see e.g. Ahmad et al. (2013).

assumption that the average displacement u_π is equal to the average displacement u_h at the vertices of the element. In more detail we obtain for the two nodes of the bar element with $x_1 = 0$ and $x_2 = l_e$

$$\sum_{n=1}^{2} u_\pi(x_n) = \sum_{n=1}^{2} u_h(x_n) \rightarrow (a_1 + a_1 + a_2 l_e) = (u_1 + u_2) \tag{1.13}$$

and together with (1.12) $a_1 = u_1$. Thus $u_\pi = u_1 + \frac{u_2 - u_1}{l_e} x$. This function can be employed to evaluate the integral associated with the loading potential in (1.3). As an example we compute the potential energy U_e^f of the loading term for $f(x) = f_c = const.$

$$U_e^f = \int_0^{l_e} f_c\, u_\pi(x)\, \mathrm{d}x = \int_0^{l_e} f_c(u_1 + \frac{u_2 - u_1}{l_e} x)\, \mathrm{d}x = \frac{f_c\, l_e}{2}(u_1 + u_2) \tag{1.14}$$

which yields with the definition $\mathbf{u}_e = \langle u_1\,, u_2 \rangle^T$ the matrix form

$$\mathbf{f} = \frac{\partial U_e^f}{\partial \mathbf{u}_e} = \frac{f_c\, l_e}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} . \tag{1.15}$$

This loading term is exactly the same as the one for the finite element formulation in (1.5).

1.2.2 Quadratic Ansatz for a One-Dimensional Virtual Truss Element

To demonstrate the difference between the virtual element and finite element method a higher order ansatz function will now be used to derive a virtual element for the bar. For this purpose we discuss a quadratic ansatz for the differential equation $EAu''(x) = -f(x)$, see Fig. 1.8.

Analogous to the case of the linear projection we select a quadratic function (ansatz order $n = 2$)

$$u_\pi = a_1 + a_2\, x + a_3\, x^2 \tag{1.16}$$

This projection function has three unknown parameters and the question is: how will it be possible to derive a virtual element since we have only two points at the edges of the element with nodal displacements u_2 and u_1?

The solution follows by looking at the least square projection (1.11)

$$\int_0^{l_e} p'\,u'_\pi\,\mathrm{d}x = (p'\,u_h)\big|_0^{l_e} - \int_0^{l_e} p''\,u_h\,\mathrm{d}x\,, \tag{1.17}$$

where again p has the same polynomial degree as u_π in (1.16) and thus is a quadratic polynomial with $p'' = const$. Hence the last integral $\int u_h\,\mathrm{d}x$ does not vanish. But it is also not computable since u_h is not known inside the element. The way out is to define a new (internal) variable[4]

$$m_0 = \frac{1}{l_e}\int_0^{l_e} u_h\,\mathrm{d}x \tag{1.18}$$

which is scaled by the element length such that m_0 has the same dimensions as u_1 and u_2. With this new variable the projection in (1.17) is computable, as we will see next.

It is convenient to introduce a matrix formulation in order to shorten notation. This leads to the derivative of u_π

$$u'_\pi = a_2 + 2a_3\,x = \langle 1\ 2x\rangle \begin{Bmatrix} a_2 \\ a_3 \end{Bmatrix}\,. \tag{1.19}$$

With the matrix form $p' = \langle 1\ 2x\rangle^T$ the left hand side of (1.17) yields

$$\int_0^{l_e} p'\,u'_\pi\,\mathrm{d}x = \int_0^{l_e} \begin{Bmatrix} 1 \\ 2x \end{Bmatrix} \langle 1\ 2x\rangle\,\mathrm{d}x \begin{Bmatrix} a_2 \\ a_3 \end{Bmatrix} = \int_0^{l_e} \begin{bmatrix} 1 & 2x \\ 2x & 4x^2 \end{bmatrix} \mathrm{d}x \begin{Bmatrix} a_2 \\ a_3 \end{Bmatrix}\,. \tag{1.20}$$

The polynomials can be integrated exactly leading to

$$\int_0^{l_e} p'\,u'_\pi\,\mathrm{d}x = \begin{bmatrix} l_e & l_e^2 \\ l_e^2 & \frac{4}{3}l_e^3 \end{bmatrix} \begin{Bmatrix} a_2 \\ a_3 \end{Bmatrix} = \mathbf{G}\,\mathbf{a}\,. \tag{1.21}$$

The right hand side in (1.17) can be evaluated with $p'' = \langle 0\ 2\rangle^T$ resulting in

$$(p'\,u_h)\big|_0^{l_e} - \int_0^{l_e} p''\,u_h\,\mathrm{d}x = \left[\begin{Bmatrix} 1 \\ 2x \end{Bmatrix} u_h\right]_0^{l_e} - \begin{Bmatrix} 0 \\ 2 \end{Bmatrix} \int_0^{l_e} u_h\,\mathrm{d}x\,. \tag{1.22}$$

[4] This variable is not associated with any node and is called moment in the virtual element literature since for higher order ansatz functions ($n \geq 2$) integrals $\int x^{n-2}\,u_h\,\mathrm{d}x$ up to the order $n-2$ appear. These can be associated with moments in mechanics.

With the unknown defined in (1.18) the right hand side of (1.17) follows as

$$(p'\,u_h)\Big|_0^{l_e} - \int\limits_0^{l_e} p''\,u_h\,\mathrm{d}x = \begin{Bmatrix} u_2 - u_1 \\ 2\,l_e u_2 \end{Bmatrix} - \begin{Bmatrix} 0 \\ 2\,l_e m_0 \end{Bmatrix} = \mathbf{r}(u_i, m_0) \tag{1.23}$$

The projection Eq. (1.17) can now be solved for the unknowns **a** by combining (1.21) and (1.23)

$$\mathbf{G}\,\mathbf{a} = \mathbf{r}(u_i, m_0) \rightarrow \begin{Bmatrix} a_2 \\ a_3 \end{Bmatrix} = \begin{Bmatrix} -\frac{2}{l_e}(2u_1 + u_2 - 3m_0) \\ \frac{3}{l_e^2}(u_1 + u_2 - 2m_0) \end{Bmatrix}. \tag{1.24}$$

Furthermore the constant a_1 can be obtained by the condition that the average of the projection u_π is equal to the average of the ansatz u_h over the element

$$\int\limits_0^{l_e} u_\pi \mathrm{d}x = \int\limits_0^{l_e} u_h \mathrm{d}x\,. \tag{1.25}$$

This leads with the definition of the unknown m_0, the ansatz (1.16) and the result (1.24) after some algebra to

$$a_1 l_e + \frac{1}{2} a_2\, l_e^2 + \frac{1}{3} a_3\, l_e^3 = l_e\, m_0 \quad \rightarrow a_1 = u_1 \tag{1.26}$$

Equations (1.24) and (1.26) determine the projection and its first derivative in terms of the unknowns u_1 , u_2 and m_0

$$u_\pi = \langle 1\ x\ x^2 \rangle \begin{Bmatrix} u_1 \\ -\frac{2}{l_e}(2u_1 + u_2 - 3m_0) \\ \frac{3}{l_e^2}(u_1 + u_2 - 2m_0) \end{Bmatrix} \quad \text{and } u'_\pi = \langle 1\ 2x \rangle \begin{Bmatrix} -\frac{2}{l_e}(2u_1 + u_2 - 3m_0) \\ \frac{3}{l_e^2}(u_1 + u_2 - 2m_0) \end{Bmatrix} \tag{1.27}$$

This result can be written in a more compact matrix form by introducing the matrices

$$\mathbf{N}_\pi(x) = \langle 1\ x\ x^2 \rangle \quad \text{and } \nabla\mathbf{N}_\pi(x) = \langle 1\ 2x \rangle \tag{1.28}$$

together with the vector of unknowns $\mathbf{u}_e = \langle u_1\,, u_2\,, m_0 \rangle^T$ and the projection operators

$$\mathbb{P} = \frac{1}{l_e^2} \begin{bmatrix} l_e^2 & 0 & 0 \\ -4l_e & -2l_e & 6l_e \\ 3 & 3 & -6 \end{bmatrix} \quad \text{and } \mathbb{B} = \frac{1}{l_e^2} \begin{bmatrix} -4l_e & -2l_e & 6l_e \\ 3 & 3 & -6 \end{bmatrix}, \tag{1.29}$$

leading to

$$u_\pi = \mathbf{N}_\pi(x)\,\mathbb{P}\,\mathbf{u}_e \quad \text{and } u'_\pi = \nabla\mathbf{N}_\pi(x)\,\mathbb{B}\,\mathbf{u}_e\,. \tag{1.30}$$

The derivative u'_π from (1.30) can now be inserted into the potential energy and integrated using the result from (1.21)

$$U_e = \frac{1}{2}\int_0^{l_e} E A\,(u'_\pi)^2\,\mathrm{d}x = \frac{1}{2}\mathbf{u}_e^T\mathbb{B}^T\int_0^{l_e}[\nabla\mathbf{N}_\pi(x)]^T\,\nabla\mathbf{N}_\pi(x)\,\mathrm{d}x\,\mathbb{B}\,\mathbf{u}_e = \frac{1}{2}\mathbf{u}_e^T\mathbb{B}^T\mathbf{G}\,\mathbb{B}\,\mathbf{u}_e\,. \tag{1.31}$$

Now the stiffness matrix of the virtual truss/bar element follows for the quadratic ansatz by differentiation with respect to $\mathbf{u}_e$ as

$$\mathbf{K}_q^{VEM} = \frac{\partial^2 U_e}{\partial\mathbf{u}_e\partial\mathbf{u}_e} = \mathbb{B}^T\mathbf{G}\,\mathbb{B} = \frac{2\,EA}{l_e}\begin{bmatrix} 2 & 1 & -3\\ 1 & 2 & -3\\ -3 & -3 & 6\end{bmatrix} \tag{1.32}$$

It is interesting to note that the VEM stiffness matrix is not equivalent to the FEM matrix for a quadratic element which can be computed from the potential (1.5) using the ansatz functions in (3.6)

$$\mathbf{K}_q^{FEM} = \frac{2\,EA}{l_e}\begin{bmatrix} \frac{7}{6} & \frac{1}{6} & -\frac{4}{3}\\ \frac{1}{6} & \frac{7}{6} & -\frac{4}{3}\\ -\frac{4}{3} & -\frac{4}{3} & \frac{8}{3}\end{bmatrix}. \tag{1.33}$$

However, when looking at the eigenvalues ω of both stiffness matrices we obtain $\omega_q^{VEM} = \frac{2\,EA}{l_e}\{0\,,1\,,9\}$ and $\omega_q^{FEM} = \frac{2\,EA}{l_e}\{0\,,1\,,4\}$. These have the same subset $\frac{2\,EA}{l_e}\{0\,,1\}$ which are actually the two eigenvalues of the stiffness matrix, see (1.9), for the virtual and the finite truss element with linear ansatz function. Hence both stiffness matrices have the correct rank. The zero eigenvalue is associated with the rigid body translation. Again, a stabilization for the quadratic virtual element is not necessary since the eigenvalues ω_q^{VEM} depict no rank-deficiency of the stiffness matrix.

The loading term follows from the potential, see (1.14),

$$U_f = \int_0^{l_e} f(x)\,u_h\,\mathrm{d}x\,. \tag{1.34}$$

Interestingly, for the case of $f(x) = f_c = const.$ the integral $\int u_h\,\mathrm{d}x$ can be approximated directly by the variable m_0 introduced in (1.18) leading to

$$U_f = f_c\int_0^{l_e} u_h\,\mathrm{d}x = f_c\,m_0\,l_e\,. \tag{1.35}$$

Thus the matrix form of the loading term is

$$\mathbf{f}_q^{VEM} = \frac{\partial U_f}{\partial \mathbf{u}_e} = f_c l_e \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} \tag{1.36}$$

which is counterintuitive since no load term is assigned to the nodal degrees of freedom u_1 and u_2. The loading term for finite elements is given for a constant load by

$$\mathbf{f}_q^{FEM} = \frac{f_c l_e}{6} \begin{Bmatrix} 1 \\ 1 \\ 4 \end{Bmatrix} \tag{1.37}$$

which is clearly different from (1.36).

In order to verify the correctness of the virtual element formulation leading to (1.32) and (1.36), a simple example of a bar under constant load f_c is considered that is fixed at the left end ($u_1 = 0$), see Fig. 1.7. Using only one virtual element for the discretization yields with (1.32) and (1.36)

$$\frac{2\,EA}{l_e} \begin{bmatrix} 2 & -3 \\ -3 & 6 \end{bmatrix} \begin{Bmatrix} u_2 \\ m_0 \end{Bmatrix} = f_c l_e \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \quad \rightarrow \quad \begin{Bmatrix} u_2 \\ m_0 \end{Bmatrix} = \frac{f_c L_e^2}{6\,EA} \begin{Bmatrix} 3 \\ 2 \end{Bmatrix}. \tag{1.38}$$

This result can be introduced in (1.30) to compute the displacement u_π and normal force $N = EAu_\pi'$ which leads after some simple algebra to

$$u_\pi = \frac{f_c}{2\,EA}(2l_e - x)x \quad \text{and } N = f_c(l_e - x)\,. \tag{1.39}$$

These results are equivalent to the exact analytical solution of the differential equation $EAu'' = -f_c$. Thus the projected displacement u_π of the quadratic virtual truss element delivers an exact solution in this special case. Actually, the exact analytical solution is also recovered when using the finite element formulation in (1.33) with the load vector in (1.37). But we observe that stiffness matrix and load vector of the virtual element are different from the finite element formulation.

The basic differences between finite element and virtual element formulations were shown in this simple example. They are related to the different methodologies, which for virtual elements are based on the real geometry and on projections $u_h \approx \Pi u_h = u_\pi$ where u_h is not known in the element Ω_e. The virtual element formulation can lead to the same matrices as obtained using the finite element method. However the second example reveiled that this is only true for special cases.[5]

[5] Even for problems in solid mechanics the equivalence between FEM and VEM is true for triangular elements with three vertices in two dimensions and in three dimensions for tetrahedral elements with four vertices.

1.3 Contents of the Book

The book is subdivided in twelves further chapters which include the following subject areas:

- Chapter 2 contains an introduction to continuum mechanics which provides the theoretical basis in solid mechanics for all formulations in the subsequent chapters.
- Ansatz spaces and functions are discussed for the virtual element method when applied to solid mechanics problems in Chap. 3. This establishes a basis for an unified treatment of virtual element formulations and includes the computations of projections $u_\pi = \Pi u_h$ onto a polynomial space that replace the sought approximation u_h of the solution.
- The same ansatz functions can be applied to problems related to the Poisson equation which is discussed in Chap. 4.
- Chapter 5 discusses the general construction of a virtual element method based on a weak form or a potential including methodologies for stabilization of a virtual element formulation. The chapter closes with an example of the virtual element method when applied to solve the Poisson equation.
- Chapter 6 describes the application field of the method to elasticity in solid mechanics. Linear and finite strain deformations of elastic solids are considered. The treatment includes compressible and incompressible deformations as well as applications for anisotropic materials.
- Explicit and implicit algorithms for time dependent problems are described in Chap. 7. Here different ways of generating mass matrices are investigated. The application to nonlinear problems demonstrates that virtual elements can be successfully used in nonlinear dynamics. The presented algorithms can be applied to problems with small and large strains.
- The treatment of finite strain plasticity problems is subject of Chap. 8. Here two- and three-dimensional applications are presented.
- An extension of finite strain plasticity to the analysis of coupled problems is provided in Chap. 9. Thermo-mechanical coupling is formulated on the basis of the virtual element method. Examples depict that the two-field analysis can be successful treated with the virtual element method.
- Advantages of using virtual elements were exemplarily demonstrated in Sect. 1.1. Among them is the treatment of fracture problems which is subject of Chap. 10. Different methods are presented and discussed, among them are approaches based on stress intensity factors, phase-field concepts, damage mechanics, cutting technologies and combinations of different techniques.
- Contact mechanics is another field where virtual element technologies can be applied profitably. The description of the related formulation and algorithms for small and finite deformations is contained in Chap. 11.
- Chapter 12 discusses the use of virtual elements in the area of homogenization. Here the advantages of a direct discretization of grains with only one virtual element are investigated and compared to finite element approaches.

- Chapter 13 presents virtual elements for beams and plates. Advantages of the virtual element method enable a simple Kirchhoff formulation of C^1-continuous ansatz functions. This possibility is explored, leading to virtual elements with arbitrary number of nodes and shape. The resulting elements can then be specialized to triangular and quadrilateral elements and provide a new class of C^1-continuous elements for engineering analysis.

The book is written as a textbook for the virtual element method as a tool to solve engineering problems in solid mechanics. Due to that a study of Chaps. 3–5, containing the general background of virtual element formulation, might be advantageous before reading Chaps. 6–13 on applications in engineering.

Several software packages are available that include virtual element formulations for the two- and three-dimensional analysis of the Laplace equation as well as elasticity and plate problems using C^0- and C^1-continuous ansatz spaces. The packages are based on *MATLAB*, see e.g. Sutton (2017), Yu (2022) and Herrera et al. (2023), and on the other hand on the multi-language and multi-environment numerical code generation package *AceGen* that is an extension of *Mathematica*, see Korelc (2000) and Korelc and Wriggers (2016). The related software for virtual elements can be found in Korelc (2023).

References

Ahmad, B., A. Alsaedi, F. Brezzi, L. Marini, and A. Russo. 2013. Equivalent projectors for virtual element methods. *Computers and Mathematics with Applications* 66: 376–391.

Aldakheel, F., B. Hudobivnik, E. Artioli, L. Beirão da Veiga, and P. Wriggers. 2020. Curvilinear virtual elements for contact mechanics. *Computer Methods in Applied Mechanics and Engineering* 372: 113394.

Aldakheel, F., B. Hudobivnik, A. Hussein, and P. Wriggers. 2018. Phase-field modeling of brittle fracture using an efficient virtual element scheme. *Computer Methods in Applied Mechanics and Engineering* 341: 443–466.

Aldakheel, F., B. Hudobivnik, and P. Wriggers. 2019. Virtual element formulation for phase-field modeling of ductile fracture. *International Journal for Multiscale Computational Engineering* 17: 181–200.

Antonietti, P.F., M. Bruggi, S. Scacchi, and M. Verani. 2017. On the virtual element method for topology optimization on polygonal meshes: A numerical study. *Computers & Mathematics with Applications* 74 (5): 1091–1109.

Argyris, J.H., I. Fried, and D.W. Scharpf. 1968. The TUBA family of plate elements for the matrix displacement method. *The Aeronautical Journal* 72 (692): 701–709.

Artioli, E., L. Beirão da Veiga, C. Lovadina, and E. Sacco. 2017. Arbitrary order 2d virtual elements for polygonal meshes: Part I, elastic problem. *Computational Mechanics* 60: 355–377.

Bathe, K.J. 1996. *Finite element procedures*. Englewood Cliffs, New Jersey: Prentice-Hall.

Beirão da Veiga, L., and G. Manzini. 2008. A higher-order formulation of the mimetic finite difference method. *SIAM, Journal of Scientific Computing* 31: 732–760.

Beirão da Veiga, L., F. Brezzi, and L. Marini. 2013a. Virtual elements for linear elasticity problems. *SIAM, Journal of Numerical Analysis* 51: 794–812.

Beirão da Veiga, L., F. Brezzi, A. Cangiani, G. Manzini, L. Marini, and A. Russo. 2013b. Basic principles of virtual element methods. *Mathematical Models and Methods in Applied Sciences* 23 (01): 199–214.

Beirão da Veiga, L., F. Brezzi, F. Dassi, L.D. Marini, and A. Russo. 2018. A family of three-dimensional virtual elements with applications to magnetostatics. *SIAM Journal on Numerical Analysis* 56 (5): 2940–2962.

Beirão da Veiga, L., F. Brezzi, L.D. Marini, and A. Russo. 2014. The hitchhiker's guide to the virtual element method. *Mathematical Models and Methods in Applied Sciences* 24 (8): 1541–1573.

Beirão da Veiga, L., K. Lipnikov, and G. Manzini. 2011. Arbitrary-order nodal mimetic discretizations of elliptic problems on polygonal meshes. *SIAM Journal of Numerical Analysis* 49: 1737–1760.

Beirão da Veiga, L., K. Lipnikov, and G. Manzini. 2013c. *The mimetic finite difference method*. Modeling, simulations and applications, vol. 11, 1st ed. Berlin: Springer.

Beirão da Veiga, L., C. Lovadina, and D. Mora. 2015. A virtual element method for elastic and inelastic problems on polytope meshes. *Computer Methods in Applied Mechanics and Engineering* 295: 327–346.

Beirão da Veiga, L., C. Lovadina, and G. Vacca. 2018. Virtual elements for the Navier-Stokes problem on polygonal meshes. *SIAM Journal on Numerical Analysis* 56 (3): 1210–1242.

Beirão da Veiga, L., D. Mora, and G. Rivera. 2019. Virtual elements for a shear-deflection formulation of Reissner-Mindlin plates. *Mathematics of Computation* 88 (315): 149–178.

Beirão da Veiga, L. 2010. A mimetic discretization method for linear elasticity. *M2AN Mathematical Modeling and Numerical Analysis* 44: 231–250.

Belgacem, F.B., P. Hild, and P. Laborde. 1997. Approximation of the unilateral contact problem by the mortar finite element method. *Comptes Rendus de l'Académie des Sciences, Paris, Series* I (324): 123–127.

Bell, K. 1969. A refined triangular plate bending finite element. *International Journal for Numerical Methods in Engineering* 1 (1): 101–122.

Belytschko, T., J.S.J. Ong, W.K. Liu, and J.M. Kennedy. 1984. Hourglass control in linear and nonlinear problems. *Computer Methods in Applied Mechanics and Engineering* 43: 251–276.

Belytschko, T., W.K. Liu, and B. Moran. 2000. *Nonlinear finite elements for continua and structures*. Chichester: Wiley.

Benedetto, M.F., A. Caggiano, and G. Etse. 2018. Virtual elements and zero thickness interface-based approach for fracture analysis of heterogeneous materials. *Computer Methods in Applied Mechanics and Engineering* 338: 41–67.

Biabanaki, S., and A. Khoei. 2012. A polygonal finite element method for modeling arbitrary interfaces in large deformation problems. *Computational Mechanics* 50: 19–33.

Biabanaki, S.O.R., A.R. Khoei, and P. Wriggers. 2014. Polygonal finite element methods for contact-impact problems on non-conformal meshes. *Computer Methods in Applied Mechanics and Engineering* 269: 198–221.

Bishop, J.E. 2014. A displacement-based finite element formulation for general polyhedra using harmonic shape functions. *International Journal for Numerical Methods in Engineering* 97: 1–31.

Böhm, C., B. Hudobivnik, M. Marino, and P. Wriggers. 2021. Electro-magneto-mechanically response of polycrystalline materials: Computational homogenization via the virtual element method. *Computer Methods in Applied Mechanics and Engineering* 375: 113775.

Brezzi, F., and L.D. Marini. 2013. Virtual element methods for plate bending problems. *Computer Methods in Applied Mechanics and Engineering* 253: 455–462.

Brezzi, F., K. Lipnikov, and V. Simoncini. 2005. A family of mimetic finite difference methods on polygonal and polyhedral meshes. *Mathematical Models and Methods in Applied Science* 15: 1533–1553.

Brezzi, F., A. Buffa, and K. Lipnikov. 2009. Mimetic finite differences for elliptic problems. *M2AN Mathematical Modeling and Numerical Analysis* 43: 277–295.

Cangiani, A., G. Manzini, A. Russo, and N. Sukumar. 2015. Hourglass stabilization and the virtual element method. *International Journal for Numerical Methods in Engineering* 102: 404–436.

Chen, A., and N. Sukumar. 2023. Stabilization-free virtual element method for plane elasticity. *Computer Methods in Applied Mechanics and Engineering* 404: 115784.

Chi, H., L. Beirão da Veiga, and G. Paulino. 2017. Some basic formulations of the virtual element method (VEM) for finite deformations. *Computer Methods in Applied Mechanics and Engineering* 318: 148–192.

Chi, H., C. Talischi, O. Lopez-Pamies, and G. Paulino. 2015. Polygonal finite elements for finite elasticity. *International Journal for Numerical Methods in Engineering* 101 (4): 305–328.

Chi, H., A. Pereira, I.F.M. Menezes, and G.H. Paulino. 2020. Virtual element method (VEM)-based topology optimization: An integrated framework. *Structural and Multidisciplinary Optimization* 62 (3): 1089–1114.

Chinosi, C., and L.D. Marini. 2016. Virtual element method for fourth order problems: L2-estimates. *Computers & Mathematics with Applications* 72 (8): 1959–1967.

Cottrell, J.A., T.J.R. Hughes, and Y. Bazilevs. 2009. *Isogeometric analysis: Toward integration of CAD and FEA*. New York: Wiley.

D' Altri, A.M., S. de Miranda, L. Patruno, and E. Sacco. 2021. An enhanced VEM formulation for plane elasticity. *Computer Methods in Applied Mechanics and Engineering* 376: 113663.

De Bellis, M., P. Wriggers, and B. Hudobivnik. 2019. Serendipity virtual element formulation for nonlinear elasticity. *Computers & Structures* 223: 106094.

Droniou, J., R. Eymard, T.R. Gallouet, and R. Herbin. 2010. A unified approach to mimetic finite difference, hybrid finite volume and mixed finite volume methods. *Mathematical Models and Methods in Applied Science* 20: 265–295.

Falco, S., J. Jiang, F. De Cola, and N. Petrinic. 2017. Generation of 3d polycrystalline microstructures with a conditioned laguerre-voronoi tessellation technique. *Computational Materials Science* 136: 20–28.

Gain, A. L. 2013. Polytope-based topology optimization using a mimetic-inspired method. Dissertation, University of Illinois at Urbana-Champaign.

Gain, A.L., C. Talischi, and G.H. Paulino. 2014. On the virtual element method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes. *Computer Methods in Applied Mechanics and Engineering* 282: 132–160.

Hallquist, J. O. 1984. Nike 2d: An implicit, finite deformation, finite element code for analyzing the static and dynamic response of two-dimensional solids. Technical Report Rept. UCRL-52678, Lawrence Livermore National Laboratory, University of California, Livermore, CA.

Herrera, C., R. Corrales-Barquero, J. Arroyo-Esquivel, and J.G. Calvo. 2023. A numerical implementation for the high-order 2d virtual element method in MATLAB. *Numerical Algorithms* 92 (3): 1707–1721.

Hudobivnik, B., F. Aldakheel, and P. Wriggers. 2018. Low order 3d virtual element formulation for finite elasto-plastic deformations. *Computational Mechanics* 63: 253–269.

Hughes, T.J.R. 1987. *The finite element method*. Englewood Cliffs, New Jersey: Prentice Hall.

Hussein, A., F. Aldakheel, B. Hudobivnik, P. Wriggers, P.A. Guidault, and O. Allix. 2019. A computational framework for brittle crack propagation based on an efficient virtual element method. *Finite Elements in Analysis and Design* 159: 15–32.

Hussein, A., B. Hudobivnik, and P. Wriggers. 2020. A combined adaptive phase field and discrete cutting method for the prediction of crack paths. *Computer Methods in Applied Mechanics and Engineering* 372: 113329.

Korelc, J. 2000. Automatic generation of numerical codes with introduction to AceGen 4.0 symbolc code generator. http://www.fgg.uni-lj.si/Symech

Korelc, J. 2023. Acegen, AceFEM and AceShare. http://symech.www.fgg.uni-lj.si

Korelc, J., and P. Wriggers. 2016. *Automation of finite element methods*. Berlin: Springer.

Kuznetsov, Y., and S. Repin. 2003. New mixed finite element method on polygonal and polyhedral meshes. *Russian Journal of Numerical Analysis and Mathematical Modelling* 18: 261–278.

Manzini, G., A. Russo, and N. Sukumar. 2014. New perspectives on polygonal and polyhedral finite element methods. *Mathematical Models and Methods in Applied Science* 24: 1665–1700.

Marino, M., B. Hudobivnik, and P. Wriggers. 2019. Computational homogenization of polycrystalline materials with the virtual element method. *Computer Methods in Applied Mechanics and Engineering* 355: 349–372.

Martin, S., P. Kaufmann, M. Botsch, M. Wicke, and M. Gross. 2008. Polyhedral finite elements using harmonic basis functions. *Computer Graphics Forum* 27: 1521–1529.

Milbradt, P., and T. Pick. 2008. Polytope finite elements. *International Journal for Numerical Methods in Engineering* 73: 1811–1835.

Noels, L., and R. Radovitzky. 2006. A general discontinuous Galerkin method for finite hyperelasticity, formulation and numerical applications. *International Journal for Numerical Methods in Engineering* 68: 64–97.

Oñate, E. 2009. *Structural analysis with the finite element method, Vol. 1: Basis and solids*. Berlin: Springer.

Proudhon, H., N. Guéninchault, S. Forest, and W. Ludwig. 2018. Incipient bulk polycrystal plasticity observed by synchrotron in-situ topotomography. *Materials* 11 (10).

Simo, J.C., P. Wriggers, and R.L. Taylor. 1985. A perturbed Lagrangian formulation for the finite element solution of contact problems. *Computer Methods in Applied Mechanics and Engineering* 50: 163–180.

Stein, E., R. De Borst, and T.J. Hughes. 2004. *Encyclopedia of computational mechanics*. New York: Wiley.

Sukumar, N. 2004. Construction of polygonal interpolants: A maximum entropy approach. *International Journal for Numerical Methods in Engineering* 61 (12): 2159–2181.

Sukumar, N., and E.A. Malsch. 2006. Recent advances in the construction of polygonal finite element interpolants. *Archives of Computational Methods in Engineering* 13 (1): 129.

Sutton, O.J. 2017. The virtual element method in 50 lines of matlab. *Numerical Algorithms* 75 (4): 1141–1159.

ten Eyck, A., and A. Lew. 2006. Discontinuous Galerkin methods for non-linear elasticity. *International Journal for Numerical Methods in Engineering* 67: 1204–1243.

Vacca, G., and L. Beirão da Veiga. 2015. Virtual element methods for parabolic problems on polygonal meshes. *Numerical Methods for Partial Differential Equations* 31 (6): 2110–2134.

Wohlmuth, B.I. 2000. A mortar finite element method using dual spaces for the lagrange multiplier. *SIAM, Journal of Numerical Analysis* 38: 989–1012.

Wriggers, P. 2008. *Nonlinear finite elements*. Berlin, Heidelberg, New York: Springer.

Wriggers, P., and B. Hudobivnik. 2017. A low order virtual element formulation for finite elasto-plastic deformations. *Computer Methods in Applied Mechanics and Engineering* 327: 459–477.

Wriggers, P., and W. Rust. 2019. A virtual element method for frictional contact including large deformations. *Engineering Computations* 36: 2133–2161.

Wriggers, P., and J. Simo. 1985. A note on tangent stiffnesses for fully nonlinear contact problems. *Communications in Applied Numerical Methods* 1: 199–203.

Wriggers, P., W. Rust, and B. Reddy. 2016. A virtual element method for contact. *Computational Mechanics* 58: 1039–1050.

Wriggers, P., B. Reddy, W. Rust, and B. Hudobivnik. 2017. Efficient virtual element formulations for compressible and incompressible finite deformations. *Computational Mechanics* 60: 253–268.

Wriggers, P., B. Hudobivnik, and F. Aldakheel. 2021. NURBS-based geometries: A mapping approach for virtual serendipity elements. *Computer Methods in Applied Mechanics and Engineering* 378: 113732.

Yu, Y. 2022. mVEM: A MATLAB software package for the virtual element methods. arXiv:2204.01339.

Zhang, X.S., H. Chi, and G.H. Paulino. 2020. Adaptive multi-material topology optimization with hyperelastic materials under large deformations: A virtual element approach. *Computer Methods in Applied Mechanics and Engineering* 370: 112976.

Zienkiewicz, O.C., and R.L. Taylor. 2000. *The finite element method*, vol. 1, 5th ed. Oxford, UK: Butterworth-Heinemann.

Chapter 2
Continuum Mechanics Background

This chapter summarizes the basic relation needed to formulate the deformation of solids in the linear and nonlinear range. It is subdivided into the sections kinematics, balance laws, variational formulations and constitutive equations. This part of the book is not meant for studying continuum mechanics, it only summarizes results that provide essential background and notation for understanding the discretization techniques related to the virtual element method in the following chapters. For an in depth background on continuum mechanics, we refer to e.g. Truesdell and Toupin (1960), Truesdell and Noll (1965), Eringen (1967), Malvern (1969), Gurtin et al. (2010), Chadwick (2012), Tadmor et al. (2012), Shabana (2018) and Holzapfel (2000).

2.1 Basic Equations

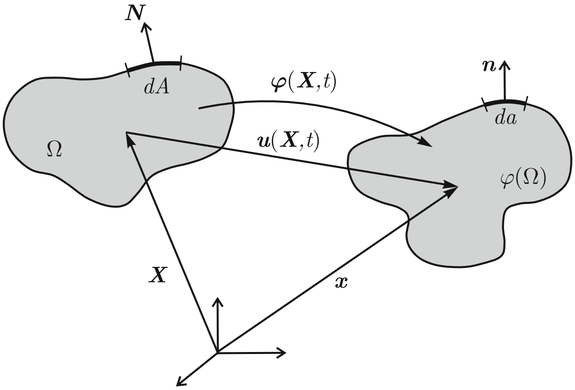
All basic equations are written in terms of a direct notation of tensors and vectors. For more classical notation using indices we refer to Eringen (1967). In general a vector $\boldsymbol{v}$ can be written as $\boldsymbol{v} = v_i\, \boldsymbol{E}_i$ where $\boldsymbol{E}_i$ are the unit basis vectors in a Cartesian coordinate system. In the same way a tensor $\boldsymbol{T}$ can be written as $\boldsymbol{T} = T_{ik}\, \boldsymbol{E}_i \otimes \boldsymbol{E}_k$. The tensor maps between to vector spaces $\boldsymbol{v} = \boldsymbol{T}\, \boldsymbol{w}$ which is in index notation

$$v_i\, \boldsymbol{E}_i = (T_{ik}\, \boldsymbol{E}_i \otimes \boldsymbol{E}_k) w_m\, \boldsymbol{E}_m = T_{ik}\, w_m \boldsymbol{E}_i (\boldsymbol{E}_k \cdot \boldsymbol{E}_m) = T_{ik}\, w_m \boldsymbol{E}_i \delta_{km} = T_{im}\, w_m \boldsymbol{E}_i\,.$$

Note that this multiplication between a tensor and a vector or between a tensor and a tensor is written without $''\cdot''$ in this book. The $''\cdot''$ is only used for a scalar product between vectors and tensors, e.g. $\boldsymbol{v} \cdot \boldsymbol{w} = v_i w_i$ and $\boldsymbol{T} \cdot \boldsymbol{U} = T_{ik} U_{ik}$.

P. Wriggers et al., *Virtual Element Methods in Engineering Sciences*,
https://doi.org/10.1007/978-3-031-39255-9_2

Fig. 2.1 Deformation states of a solid



2.1.1 Kinematics

Consider a body that occupies the bounded domain $\Omega \subset \mathbb{R}^n$ with $n = 2\,, 3$. It can undergo finite motions which deform the body from its initial configuration Ω into a current configuration $\varphi(\Omega)$. The following relations define deformation and strain states under unconstraint and constraint motions (Fig. 2.1).

The position $\boldsymbol{x}$ of a material point in the current configuration $\varphi(\Omega)$, initially at $\boldsymbol{X}$ in Ω, is given by the motion

$$\boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X}, t) = \boldsymbol{X} + \boldsymbol{u}(\boldsymbol{X}, t) \tag{2.1}$$

where $\boldsymbol{u}$ is the displacement and t denotes the time.[1] We define the deformation gradient $\boldsymbol{F}$ by

$$\boldsymbol{F} = \text{Grad}\,\boldsymbol{\varphi} = \boldsymbol{1} + \text{Grad}\,\boldsymbol{u} \tag{2.2}$$

being evaluated with respect to $\boldsymbol{X}$.[2] It links line elements in the current and initial configuration: $\mathrm{d}\boldsymbol{x} = \boldsymbol{F}\,\mathrm{d}\boldsymbol{X}$. The determinant of $\boldsymbol{F}$, known as the *Jacobian*,

$$J = \det \boldsymbol{F} \tag{2.3}$$

defines a measure for the change in volume: $\mathrm{d}v = J\,\mathrm{d}V$ where v is the volume in the current and V the volume in the initial configuration. Additionally one can compute the change of the surface elements a in the current and A in the initial configuration by Nansons's formula

$$\boldsymbol{n}\,\mathrm{d}a = J\boldsymbol{F}^{-T}\boldsymbol{N}\mathrm{d}A \tag{2.4}$$

[1] In the following the argument of the introduced physical fields, like the displacement field is omitted to shorten notation. Thus instead of $\boldsymbol{u}(\boldsymbol{X}, t)$ only $\boldsymbol{u}$ is used.

[2] Generally, operatores, like Grad$\,(\bullet)$ and Div$\,(\bullet)$, which start with a capital letter denote differentiation with respect to the coordinates $\boldsymbol{X}$ in the reference configuration. Contrary, operators, , like grad$\,(\bullet)$ and div$\,(\bullet)$, denote differentiation with respect to the current coordinates $\boldsymbol{x}$.

where $\boldsymbol{n}$ is the outward normal vector of the surface element in the current configuration and $\boldsymbol{N}$ the outward normal vector of the surface element in the initial configuration.

As a strain measure one can introduce the right Cauchy-Green tensor $\boldsymbol{C}(\boldsymbol{u})$ which is defined as

$$\boldsymbol{C} = \boldsymbol{F}^T \boldsymbol{F} \tag{2.5}$$

and acts with respect to the initial configuration. Its counterpart is the left Cauchy-Green tensor

$$\boldsymbol{b} = \boldsymbol{F}\boldsymbol{F}^T \,. \tag{2.6}$$

which is related to the current configuration. Based on (2.5) the Green-Lagrange strain tensor can be defined

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{1}) = \frac{1}{2}(\boldsymbol{F}^T \boldsymbol{F} - \boldsymbol{1}) \tag{2.7}$$

where $\boldsymbol{1} = \delta_{ik} \boldsymbol{E}_i \otimes \boldsymbol{E}_k$ is the second order identity tensor. With the right part of (2.2) the Green-Lagrange strains can be written in terms of the displacement field

$$\boldsymbol{E} = \frac{1}{2}(\mathrm{Grad}\,\boldsymbol{u} + \mathrm{Grad}\,\boldsymbol{u}^T + \mathrm{Grad}\,\boldsymbol{u}^T \mathrm{Grad}\,\boldsymbol{u}) \,. \tag{2.8}$$

This nonlinear strain measure can be simplified for the case of geometrically linear problems leading to the linear strain tensor

$$\boldsymbol{\varepsilon} = \frac{1}{2}(\mathrm{Grad}\,\boldsymbol{u} + \mathrm{Grad}\,\boldsymbol{u}^T) = \nabla_s \boldsymbol{u} \tag{2.9}$$

where $\nabla_s(\bullet) = \frac{1}{2}(\nabla(\bullet) + [\nabla(\bullet)]^T)$ is a symmetric gradient operator.

A deformation can be subjected to constraints. These are mostly volume constraints, leading to the notion of incompressibility, and directional constraints that are related to fiber reinforced materials when the fiber stiffnees is much larger than the matrix material. Both cases are discussed and formulated next.

- **Incompressibility.** Deformations that preserve the volume of a solid are associated with special materials that prevent volumetric extensions or contractions. If we recall the change of volume equation $\mathrm{d}v = J\,\mathrm{d}V$ then the volume constraint can be formulated by $\mathrm{d}v = \mathrm{d}V$ leading with (2.3) to

$$J - 1 = \det \boldsymbol{F} - 1 = 0 \tag{2.10}$$

for finite deformations. This constraint can often only be introduced in an approximative way. Then it is meaningful to split the deformation gradient in a multiplicative way: $\boldsymbol{F} = J^{\frac{1}{3}}\,\hat{\boldsymbol{F}}$, where due to construction $\det \hat{\boldsymbol{F}} = 1$. This result can be employed to define an isochoric strain tensor based on (2.5)

$$\hat{\boldsymbol{C}} = J^{-\frac{2}{3}}\,\boldsymbol{C} = (\det \boldsymbol{C})^{-\frac{1}{3}}\,\boldsymbol{C}\,. \tag{2.11}$$

In case of small strains the volume change is given by the trace of the linear strain tensor $\boldsymbol{\varepsilon}$ which has to be zero

$$\operatorname{tr}\boldsymbol{\varepsilon} = \operatorname{Div}\boldsymbol{u} = 0\,. \tag{2.12}$$

Also here it is possible to split the strain tensor in a volume preserving part $\boldsymbol{\varepsilon}_D$, called the deviatoric part, and a part that is associated with the volume change $\frac{1}{3}\operatorname{tr}\boldsymbol{\varepsilon}\,\boldsymbol{1}$

$$\boldsymbol{\varepsilon} = \frac{1}{3}\operatorname{tr}\boldsymbol{\varepsilon}\,\boldsymbol{1} + \boldsymbol{\varepsilon}_D = \frac{1}{3}\operatorname{Div}\boldsymbol{u}\,\boldsymbol{1} + \boldsymbol{\varepsilon}_D\,. \tag{2.13}$$

- **Directional constraint.** For the description of anisotropic materials the direction of fibers can be introduced by the unit vector $\boldsymbol{a}_a$ which can point in arbitrary directions. The strain in fiber direction follows from the Green-Lagrange strain tensor (2.7): $E_a = \boldsymbol{a}_a \cdot \boldsymbol{E}\boldsymbol{a}_a$. This is equivalent with

$$E_a = \frac{1}{2}(\operatorname{tr}[\boldsymbol{C}\,\boldsymbol{M}_a] - 1) \tag{2.14}$$

with the structure tensor $\boldsymbol{M}_a = \boldsymbol{a}_a \otimes \boldsymbol{a}_a$. For an inextensible fiber in direction $\boldsymbol{a}$ the strain E_a has to vanish: $E_a = 0$. Thus the constraint in fiber direction $\boldsymbol{a}$ is given by

$$\operatorname{tr}[\boldsymbol{C}\,\boldsymbol{M}_a] - 1 = 0\,. \tag{2.15}$$

In case of small strains this constraint can be formulated as

$$\varepsilon_a = \boldsymbol{a}_a \cdot \boldsymbol{\varepsilon}\boldsymbol{a}_a = \operatorname{tr}[\boldsymbol{\varepsilon}\,\boldsymbol{M}_a] = 0 \tag{2.16}$$

where the strain tensor (2.9) has been used.

2.1.2 Balance Laws

There are several balance equations that govern the motion of a solid. Relevant for problems in solid mechanics are the balance of mass, the balance of linear and angular momentum and the first and second law of thermodynamics.

2.1.2.1 Mass Balance

The first balance equation is the mass balance that states a conservation of the mass m of a solid during the motion and can be expressed as $\dot{m} = 0$ which implies $\varrho_0 \, \mathrm{d}\,V = \varrho \, \mathrm{d}\,v$. With $\mathrm{d}v = J\,\mathrm{d}V$ the mass balance in the initial configuration yields

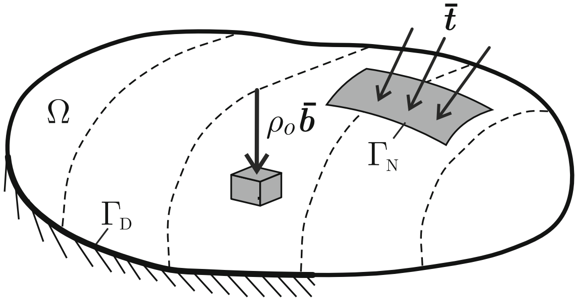


Fig. 2.2 Boundary and loading condition of body Ω

$$\varrho_0 = \varrho\, J\,. \tag{2.17}$$

The equivalent form with respect to the spatial configuration is provided by

$$\dot{\varrho} + \varrho\, \operatorname{div} \boldsymbol{v} = 0 \tag{2.18}$$

where $\boldsymbol{v}$ is the velocity.

2.1.2.2 Balance of Momentum

The body Ω has a boundary Γ which comprises non-overlapping sections Γ_D and Γ_N such that $\Gamma_D \cup \Gamma_N = \Gamma$ (Fig. 2.2). The body satisfies in Ω the momentum equation

$$\operatorname{Div} \boldsymbol{P} + \varrho_0\, \bar{\boldsymbol{b}} = \varrho_0\, \ddot{\boldsymbol{x}}\,, \tag{2.19}$$

with the body force $\bar{\boldsymbol{b}}$, the first Piola-Kirchhoff stress $\boldsymbol{P}$ and the inertia term $\varrho_0\, \ddot{\boldsymbol{x}}$ with the density ϱ_0. All quantities in this equation are related to the initial configuration and the divergence is evaluated with respect to $\boldsymbol{X}$.

The Dirichlet and Neumann boundary conditions are, respectively,

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \qquad \text{on } \Gamma_D, \tag{2.20}$$

$$\boldsymbol{P}\boldsymbol{N} = \bar{\boldsymbol{t}} \qquad \text{on } \Gamma_N. \tag{2.21}$$

with $\boldsymbol{N}$ being the outward unit normal vector with respect to the initial configuration, $\bar{\boldsymbol{u}}$ the prescribed displacement, and $\bar{\boldsymbol{t}}$ the surface traction on Γ_N.

The momentum equation can be defined as well in the current configuration $\varphi(\Omega)$

$$\operatorname{div} \boldsymbol{\sigma} + \varrho\, \bar{\boldsymbol{b}} = \varrho\, \ddot{\boldsymbol{x}}\,, \tag{2.22}$$

with density ϱ, the Cauchy stress $\boldsymbol{\sigma}$ and the inertia term $\varrho\, \ddot{\boldsymbol{x}}$, being all related to the current configuration. In this case the Dirichlet and Neumann boundary conditions are given by

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \qquad \text{on } \varphi(\Gamma_D), \tag{2.23}$$

$$\boldsymbol{\sigma n} = \bar{\boldsymbol{t}} \qquad \text{on } \varphi(\Gamma_N). \tag{2.24}$$

with $\boldsymbol{n}$ being the outward unit normal vector with respect to the current configuration, $\bar{\boldsymbol{u}}$ the prescribed displacement, and $\bar{\boldsymbol{t}}$ the surface traction on $\varphi(\Gamma_N)$ in the current configuration.

It is possible to express the Cauchy stresses in terms of the first Piola-Kirchhoff stresses via

$$\boldsymbol{\sigma} = \frac{1}{J}\,\boldsymbol{P}\boldsymbol{F}^T\,. \tag{2.25}$$

Furthermore, for later reference, we introduce the second Piola-Kirchhoff stress tensor

$$\boldsymbol{S} = \boldsymbol{F}^{-1}\boldsymbol{P} \tag{2.26}$$

and the Kirchoff stress tensor

$$\boldsymbol{\tau} = J\boldsymbol{\sigma} = \boldsymbol{P}\boldsymbol{F}^T = \boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T\,. \tag{2.27}$$

The balance of angular momentum yields the symmetry of the Cauchy stress tensor as well as of the Kirchhoff and second Piola-Kirchhoff tensor

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^T \quad \boldsymbol{\tau} = \boldsymbol{\tau}^T \quad \boldsymbol{S} = \boldsymbol{S}^T\,. \tag{2.28}$$

For the first Piola-Kirchhoff tensor one obtains the condition

$$\boldsymbol{P}\boldsymbol{F}^T = \boldsymbol{F}\boldsymbol{P}^T\,. \tag{2.29}$$

2.1.2.3 First Law of Thermodynamics

The first law of thermodynamics postulates the conservation of energy in a thermodynamical process

$$\dot{E} = P + Q\,. \tag{2.30}$$

The mechanical power due to volume and surface loads is given by

$$P = \int\limits_{\varphi(\Omega)} \varrho\,\bar{\boldsymbol{b}}\dot{\boldsymbol{x}}\,\mathrm{d}v + \int\limits_{\varphi(\Gamma)} \boldsymbol{t}\cdot\dot{\boldsymbol{x}}\,\mathrm{d}a\,. \tag{2.31}$$

where $\dot{\boldsymbol{x}}$ is the velocity. The heat supply

$$Q = -\int\limits_{\varphi(\Gamma)} \boldsymbol{q}\cdot\boldsymbol{n}\,\mathrm{d}a + \int\limits_{\varphi(\Omega)} \varrho\, r\,\mathrm{d}v \tag{2.32}$$

consists of a conduction through the surface of the body which is described by the heat flux vector $\boldsymbol{q}$ and the surface normal $\boldsymbol{n}$ and a distributed inner heat source r (specific heat supply).

The total energy E is composed of the kinetic energy

$$K = \int\limits_{\varphi(\Omega)} \frac{1}{2} \varrho \dot{\boldsymbol{x}} \cdot \dot{\boldsymbol{x}} \, \mathrm{d}v \tag{2.33}$$

and the internal energy

$$U = \int\limits_{\varphi(\Omega)} \varrho u \, \mathrm{d}v \, . \tag{2.34}$$

where u is the specific internal energy. Inserting all these relations into equation $\dot{E} = P + Q$ yields after several manipulations the local form of the first law of thermodynamics

$$\varrho \, \dot{u} = \boldsymbol{\sigma} \cdot \boldsymbol{d} + \varrho \, r - \operatorname{div} \boldsymbol{q} \, . \tag{2.35}$$

The term $\boldsymbol{\sigma} \cdot \boldsymbol{d}$ is called specific stress power where $\boldsymbol{d} = \frac{1}{2} \left[\operatorname{grad} \dot{\boldsymbol{x}} + (\operatorname{grad} \dot{\boldsymbol{x}})^T \right]$ is the symmetric velocity gradient.

In the framework of constitutive theory the free HELMHOLTZ energy ψ is often introduced by the relation

$$\psi = u - \eta \, \theta \, . \tag{2.36}$$

Here η denotes the entropy of the system and θ is the absolute temperature. With this definition the first law of thermodynamics can be recast as

$$\varrho \, \dot{\psi} = \boldsymbol{\sigma} \cdot \boldsymbol{d} + \varrho \, r - \operatorname{div} \boldsymbol{q} - \dot{\eta} \, \theta - \eta \, \dot{\theta} \, . \tag{2.37}$$

The transformation of the first law of thermodynamics to the initial configuration leads to

$$\varrho_0 \, \dot{u} = \boldsymbol{S} \cdot \dot{\boldsymbol{E}} - \operatorname{Div} \boldsymbol{Q} + \varrho_0 \, R \, . \tag{2.38}$$

Here the heat source R and the heat flux vector $\boldsymbol{Q}$ are referred to the initial configuration. The stress power in (2.35) can be written with (2.7) and (2.27) as

$$\boldsymbol{S} \cdot \dot{\boldsymbol{E}} = \frac{1}{2} \, \boldsymbol{S} \cdot \dot{\boldsymbol{C}} = \boldsymbol{P} \cdot \dot{\boldsymbol{F}} = \boldsymbol{\tau} \cdot \boldsymbol{d} \, . \tag{2.39}$$

In this equivalence, the first three terms are related to the initial configuration whereas the last term is referred to the current configuration.

Some special cases of thermodynamical processes can now be stated:

1. Supply of heat energy is excluded as well in the interior as over the surface of the body ($r = 0 \, , \boldsymbol{q} = \boldsymbol{0}$) such process is called *adiabatic*.

2. A process in which the temperature in the body is kept constant ($\theta = const.$) is known as *isothermal* process.
3. A process where neither heat is supplied to an elastic body nor external forces act on the body leads to conservation of total energy

$$\dot{E} = \dot{K} + \dot{U} = 0 \;\Leftrightarrow\; E = K + U \equiv \text{const.}\,. \tag{2.40}$$

On the other hand, if we assume a rigid body with $\boldsymbol{S} \cdot \dot{\boldsymbol{E}} = 0$ and a stationary process with $\varrho_0 \, \dot{u} = 0$ then (2.38) reduces to the energy equation for stationary heat conduction

$$-\,\text{Div}\,\boldsymbol{Q} + \varrho_0 \, R = 0\,. \tag{2.41}$$

With Fourier's law, stating $\boldsymbol{Q} = -k\,\text{Grad}\,\theta$, the partial differential equation for the temperature θ follows as

$$\text{Div}\,(k\,\text{Grad}\,\theta) + \varrho_0 \, R = 0\,. \tag{2.42}$$

2.2 Constitutive Equations

There exist many constitutive relations for different engineering materials like e.g. steel, concrete, aluminum and composites. Purely elastic material behaviour is discussed in this section which is related to the name hyper elasticity, see e.g. Ogden (1984). This description is valid for many materials—like e.g. foam or rubbers—which undergo finite deformations. In case of small strains these constitutive equations reduce to the classical law of Hooke. Furthermore elasto-plastic material behaviour is summarized. To discuss constitutive relations in depth is not possible in this book. Thus only the most frequently used constitutive equations are introduced in this section. More specific details can be found in the chapters where the virtual element method is applied to different engineering problems.

2.2.1 Linear Elasticity

In linear elasticity the constitutive response of a material is governed by the so called Hooke's law. It can be obtained by using the strain energy function

$$\Psi = \frac{\Lambda}{2}\,[\text{tr}(\boldsymbol{\varepsilon})]^2 + \mu\,\text{tr}(\boldsymbol{\varepsilon}\,\boldsymbol{\varepsilon}) \tag{2.43}$$

where Λ and μ are the Lamé constants which can be expressed in terms of the Young's modulus E and the Poisson ratio ν

$$\Lambda = \frac{E\,\nu}{(1-2\nu)(1+\nu)} \qquad \mu = \frac{E}{2(1+\nu)}\,. \tag{2.44}$$

The second Lamé constant μ is equivalent to the shear modulus. From the strain energy (2.43) the stress tensor $\boldsymbol{\sigma}$ follows by differentiation with respect to the strains

$$\boldsymbol{\sigma} = \frac{\partial \Psi}{\partial \boldsymbol{\varepsilon}} = \Lambda\ \mathrm{tr}(\boldsymbol{\varepsilon})\boldsymbol{1} + 2\mu\,\boldsymbol{\varepsilon} \tag{2.45}$$

where $\boldsymbol{1}$ is the second order unit tensor. The stress strain relation is often written in matrix form using the Voigt notation. In that case the second order tensors like stresses and strains reduce to vectors and the fourth order tensor that denotes the constitutive matrix becomes a matrix. For the plane strain case in two-dimensions this leads to

$$\boldsymbol{\sigma} = \mathbb{C}_{pl\varepsilon}\,\hat{\boldsymbol{\varepsilon}} \Longleftrightarrow \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} = \begin{bmatrix} \Lambda + 2\,\mu & \Lambda & 0 \\ \Lambda & \Lambda + 2\,\mu & 0 \\ 0 & 0 & \mu \end{bmatrix} \begin{Bmatrix} u_{x,x} \\ u_{y,y} \\ u_{x,y} + u_{y,x} \end{Bmatrix} \tag{2.46}$$

which is equivalent to (2.45). In the same way the stress strain relation for the plane stress case follows as

$$\boldsymbol{\sigma} = \mathbb{C}_{pl\sigma}\,\hat{\boldsymbol{\varepsilon}} \Longleftrightarrow \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{Bmatrix} u_{x,x} \\ u_{y,y} \\ u_{x,y} + u_{y,x} \end{Bmatrix}. \tag{2.47}$$

The three-dimensional case leads to

$$\boldsymbol{\sigma} = \mathbb{C}\,\boldsymbol{\varepsilon} \Longleftrightarrow \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{Bmatrix} = \begin{bmatrix} \Lambda + 2\,\mu & \Lambda & \Lambda & 0 & 0 & 0 \\ \Lambda & \Lambda + 2\,\mu & \Lambda & 0 & 0 & 0 \\ \Lambda & \Lambda & \Lambda + 2\,\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \begin{Bmatrix} u_{x,x} \\ u_{y,y} \\ u_{z,z} \\ u_{x,y} + u_{y,x} \\ u_{y,z} + u_{z,y} \\ u_{x,z} + u_{z,x} \end{Bmatrix}. \tag{2.48}$$

In case of nearly incompressible elastic material it is advantageous to split the stress tensor in a hydrostatic $\frac{1}{3}\mathrm{tr}\boldsymbol{\sigma}\,\boldsymbol{1}$ and deviatoric stress $\boldsymbol{s}$ by

$$\boldsymbol{\sigma} = \frac{1}{3}\mathrm{tr}\,\boldsymbol{\sigma}\,\boldsymbol{1} + \boldsymbol{s} \tag{2.49}$$

then (2.45) can be recast in the form

$$\mathrm{tr}\,\boldsymbol{\sigma} = 3\,K\ \mathrm{tr}\,\boldsymbol{\varepsilon} \qquad \boldsymbol{s} = 2\mu\boldsymbol{\varepsilon} \tag{2.50}$$

where K ist the modulus of compression or bulk modulus

$$K = \Lambda + \frac{2}{3}\mu = \frac{E}{3(1-2\,\nu)}\,. \tag{2.51}$$

Note that the modulus of compression approaches for the case of an incompressible material ($\nu \rightarrow 0.5$) infinity. Since the hydrostatic stress is constant the volumetric strain $\operatorname{tr}\boldsymbol{\varepsilon}$ has to go to zero. In this case it might be advantageous to introduce as hydrostatic pressure the variable p which leads instead of (2.49) to

$$\boldsymbol{\sigma} = p\,\boldsymbol{1} + \boldsymbol{s}\,. \tag{2.52}$$

In that case the pressure is a Lagrangian parameter that can be computed using the potential or weak form. When employing the potential, the constraint (2.12) is then added as $\int_\Omega p\,\mathrm{Div}\boldsymbol{u}\,\mathrm{d}\Omega$ to (2.82).

2.2.2 Finite Elasticity

By introducing a strain energy function $\Psi(\boldsymbol{F})$ for elastic problems the first Piola-Kirchhoff stresses follow from differentiation with respect to the deformation gradient $\boldsymbol{F}$

$$\boldsymbol{P}(\boldsymbol{F}) = \frac{\partial\Psi(\boldsymbol{F})}{\partial\boldsymbol{F}}\,. \tag{2.53}$$

Due to the equivalence of the work conjugate stress strain relations, see (2.39), the second Piola-Kirchhoff and the Kirchhoff stresses follow with (2.5), (2.6) and (2.7) as

$$\boldsymbol{S} = \frac{\partial\Psi(\boldsymbol{E})}{\partial\boldsymbol{E}}\,, \quad \boldsymbol{S} = 2\frac{\partial\Psi(\boldsymbol{C})}{\partial\boldsymbol{C}} \quad \text{and} \quad \boldsymbol{\tau} = 2\boldsymbol{b}\,\frac{\partial\Psi(\boldsymbol{b})}{\partial\boldsymbol{b}}\,. \tag{2.54}$$

Different forms of the strain energy Ψ can be used. A few of them, which are employed in the following chapters are summarized in the next sections.

2.2.2.1 Compressible Materials

Many different strain energies have been proposed to model the finite strain response of elastic solids, see e.g. Ogden (1984), Holzapfel (2000). Here we just use the most simple relations knowing that the accurate prediction of the response of rubber or foam materials requires often a more sophisticated approach.

For a homogeneous compressible isotropic hyperelastic material we adopt the neo-Hookean strain energy function for the three-dimensional case

$$\Psi(\boldsymbol{C}) = \frac{\Lambda}{4}(J^2 - 1 - 2\ln J) + \frac{\mu}{2}(\operatorname{tr}\boldsymbol{C} - 3 - 2\ln J) \tag{2.55}$$

in which Λ and μ are the Lamé constants. The right Cauchy-Green tensor $\boldsymbol{C}$ is defined in (2.5) and the Jacobian J of the deformation in (2.3). Note that the right Cauchy-Green tensor as well as the Jacobian depend on the deformation and with that on the displacement field $\boldsymbol{u}$.

For a homogeneous compressible isotropic elastic material, the Neo-Hookean strain energy function can also be defined in terms of variables that act in the current configuration. This leads for example with (2.6) to

$$\Psi_b(\boldsymbol{b}) = \frac{\Lambda}{4}[\det \boldsymbol{b} - 1 - \ln(\det \boldsymbol{b})] + \frac{\mu}{2}[\operatorname{tr} \boldsymbol{b} - 3 - \ln(\det \boldsymbol{b})]\,. \tag{2.56}$$

which is equivalent to (2.55) In Eq. (2.56) μ and Λ are the Lamé constants.

Further constitutive relations to model large strain elastic deformations are given by the Mooney-Rivlin model, see Mooney (1940),

$$\Psi(\boldsymbol{C}) = c_1\,(\operatorname{tr} \boldsymbol{C} - 3) + c_2(\operatorname{tr}[\operatorname{cof}\boldsymbol{C}] - 3) \tag{2.57}$$

where c_1 and c_2 are constitutive parameters and the cofactor of the right Cauchy-Green tensor is given by $\operatorname{cof} \boldsymbol{C} = \det \boldsymbol{C}\,\boldsymbol{C}^{-T}$. This model predicts the behaviour of rubber materials up to 50% strain well.

Often the Ogden model is used to fit complex stress-strain relations in finite deformation elasticity. This material is formulated in the eigenvalues λ_i of the deformation gradient. These eigenvalues are associated with the principal stretches of the deformation. The material model depends usually on three parameter sets $(\mu_i\,, \alpha_i)$ which have to be selected carefully in order to fulfill the second law of thermodynamics, see e.g. Ogden (1984),

$$\Psi(\boldsymbol{u}) = \sum_{i=1}^{3} \frac{\mu_i}{\alpha_i} \left(\lambda_1^{\alpha_i}(\boldsymbol{u}) + \lambda_2^{\alpha_i}(\boldsymbol{u}) + \lambda_3^{\alpha_i}(\boldsymbol{u}) - 3\right)\,. \tag{2.58}$$

The principal stretches λ_i follow from the eigenvalue problem $(\boldsymbol{C} - \lambda^2 \boldsymbol{1})\,\boldsymbol{m} = \boldsymbol{0}$ where , $\boldsymbol{m}_i$ are the directions of the principal stretches.[3]

A more general material model can be formulated by including all three invariants $I_{1C} = \operatorname{tr} \boldsymbol{C}$, $I_{2C} = \operatorname{tr}(\operatorname{cof}\boldsymbol{C})$ and $I_{3C} = \det \boldsymbol{C}$ of the strain tensor

$$\Psi(\boldsymbol{C}) = \frac{c_1}{2}(\operatorname{tr} \boldsymbol{C})^2 + \frac{c_2}{2}[\operatorname{tr}(\operatorname{cof}\boldsymbol{C})]^2 + c_3\, g(\det \boldsymbol{C})\,. \tag{2.59}$$

This general constitutive equation fulfils the polyconvexity conditions, see e.g. Marsden and Hughes (1983), Schröder (2009), needed to establish existence of solutions.

[3] The eigenvalues in (2.58) need to be computed within numerical simulations. Problems occur when eigenvalues are repeated. This can be overcome by a smooth and accurate approximation around singularities related to repeated eigenvalues. The key idea is that the sum of eigenvalues can be replaced by a closed form solution of the sum, see Hudobivnik (2016).

Note that the last term in (2.59) has to be defined in a way that the strain energy is zero for the initial deformation state $\boldsymbol{u} = \mathbf{0}$.

2.2.2.2 Incompressible Materials

Nearly incompressible material behaviour can be analysed using the constitutive equations provided in the last section. However, when the incompressible behaviour is introduced via the constraint $J - 1 = 0$ then a special strain energy has to be employed since the first Lamé parameter Λ and the bulk modulus K approach infinity, see (2.51) and (2.44). By neglecting these terms and by using only the isochoric part of the right Cauchy-Green strain, see (2.11), the strain energies for Neo-Hooke, Mooney-Rivlin and Ogden materials reduce to

- Neo-Hookean material

$$\Psi^i(\boldsymbol{u}) = \frac{\mu}{2}\left(\mathrm{tr}\,[\widehat{\boldsymbol{C}}(\boldsymbol{u})] - 3\right), \tag{2.60}$$

- Mooney-Rivlin material

$$\Psi^i(\boldsymbol{u}) = c_1\left(\mathrm{tr}\,[\widehat{\boldsymbol{C}}(\mathbf{u}_e)] - 3\right) + c_2\left[\frac{1}{2}\left(\mathrm{tr}\,[\widehat{\boldsymbol{C}}(\boldsymbol{u})]^2 - \mathrm{tr}\,[\widehat{\boldsymbol{C}}(\boldsymbol{u}) \cdot \widehat{\boldsymbol{C}}(\boldsymbol{u})]\right) - 3\right], \tag{2.61}$$

- Ogden material

$$\Psi^i(\boldsymbol{u}) = \sum_{i=1}^{3} \frac{\mu_i}{\alpha_i}\left(\hat{\lambda}_1^{\alpha_i}(\boldsymbol{u}) + \hat{\lambda}_2^{\alpha_i}(\boldsymbol{u}) + \hat{\lambda}_3^{\alpha_i}(\boldsymbol{u}) - 3\right) \tag{2.62}$$

  with $\hat{\lambda}_i = \det \boldsymbol{F}^{-\frac{1}{3}} \lambda_i$.

When using these formulations the constraint $J - 1 = 0$ has to be added via a Lagrangian multiplier p leading to $p(J - 1)$.

2.2.2.3 Anisotropic Materials

A material in which a certain direction has a different constitutive behaviour than in other directions is called anisotropic. Generalizations of polyconvex strain energy functions to the anisotropic range can be found in Schröder and Neff (2003), Schröder (2009).

Here a transversely isotropic strain energy function is introduced that has to be added to an isotropic strain energy. The strain energy function is based on the introduction of a structural tensor $\boldsymbol{M}_a$, see (2.14), that defines a preferred direction $\boldsymbol{a}_a$, see Fig. 2.3.

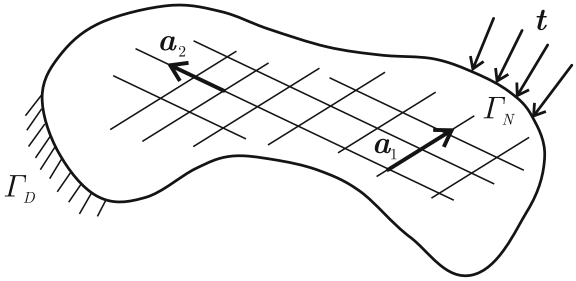


Fig. 2.3 Boundary and loading condition of body Ω

With that the transversely isotropic strain energy part Ψ^{ti} can be formulated for a general polyconvex case with one preferred direction as

$$\Psi^{ti}(\boldsymbol{C}, \operatorname{cof} \boldsymbol{C}, \det \boldsymbol{C}) = \\ C\left[\frac{1}{\alpha+1}[\operatorname{tr}(\boldsymbol{C}\,\boldsymbol{M})]^{\alpha+1} + \frac{1}{\beta+1}[\operatorname{tr}(\operatorname{cof} \boldsymbol{C}\,\boldsymbol{M})]^{\beta+1} + \frac{1}{\gamma}(\det \boldsymbol{C})^{-\gamma}\right]. \tag{2.63}$$

The constitutive parameters C , α , β and γ have to be selected such that $C > 0$, $\alpha \geq 0$, $\beta \geq 0$ and $\gamma \geq -1/2$, see Schröder and Neff (2003). In that case Ψ^{ti} is polyconvex and stress free in the initial configuration.

Note that the total strain energy will in general consist of an isotropic and an anisotropic part

$$\Psi^{total} = \Psi^{iso} + \Psi^{ti} \tag{2.64}$$

where the isotropic part Ψ^{iso} can be selected from Sect. 2.2.2.1 , for more details see e.g. Schröder and Neff (2003), Schröder (2009).

However, it is also possible to introduce the inextensibility constraint (2.15) for a preferred direction $\boldsymbol{a}$ directly. In that case the strain energy function can be written for two directional constraints, see Wriggers et al. (2016b),

$$\Psi^{total} = \Psi^{iso} + \sum_{k=1}^{2} s_k^{ti}\,(\operatorname{tr}[\boldsymbol{C}\,\boldsymbol{M}_k] - 1)\,. \tag{2.65}$$

where s_k^{ti} are Lagrangian parameters that can be interpreted as the stresses in the fiber due to the constraints. Relaxed formulations of these constraints are possible. This yields

$$\Psi^{total} = \Psi^{iso} + \sum_{k=1}^{2}\left[s_k^{ti}\,(\operatorname{tr}[\boldsymbol{C}\,\boldsymbol{M}_k] - 1) - \frac{1}{2C_k}(s_k^{ti})^2\right] \tag{2.66}$$

where C_k are constitutive parameters that represent the stiffness of the fibers.

2.2.3 *Elasto-Plasticity*

The von Mises plasticity models are summarized in this section for small and finite strains. More details regarding the modeling of plastic processes and the derivation of the underlying set of equations can be found in e.g. Simo and Hughes (1998), de Souza Neto et al. (2008).

2.2.3.1 Small Strain Elasto-Plasticity

The elasto-plastic behaviour of metals is in many cases governed by a von Mises model that describes ductile plasticity response. The following equations summarize this model for the case of linear isotropic hardening behaviour.

The strain tensor is additively split into an elastic $\boldsymbol{\varepsilon}^e$ and plastic part $\boldsymbol{\varepsilon}^p$

$$\boldsymbol{\varepsilon}^e = \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p\,. \tag{2.67}$$

The elastic constitutive relation (2.43) can be used to compute the strains, now written in terms of the elastic strains and the stresses are computed via differentiation with respect to $\boldsymbol{\varepsilon}^e$

$$\Psi = \frac{\Lambda}{2}\,[\mathrm{tr}(\boldsymbol{\varepsilon}^e)]^2 + \mu\,\mathrm{tr}(\boldsymbol{\varepsilon}^e\cdot\boldsymbol{\varepsilon}^e)\,,\qquad \boldsymbol{\sigma} = \frac{\partial\Psi}{\partial\boldsymbol{\varepsilon}^e}\,. \tag{2.68}$$

The plastic part of the strain is computed from the evolution equation

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\gamma}\,\frac{\partial f}{\partial \boldsymbol{s}} = \dot{\gamma}\,\boldsymbol{n} \tag{2.69}$$

in which f is the yield function that restricts the elastic domain of the material response. Here $\boldsymbol{s} = \boldsymbol{\sigma} - \frac{1}{3}\mathrm{tr}(\boldsymbol{\sigma})\boldsymbol{1}$ is the stress deviator, introduced in (2.49). For linear isotropic hardening it is given by the inequality

$$f(\boldsymbol{s}) = \|\boldsymbol{s}\| - \sqrt{\frac{2}{3}}(Y_0 + H\,\alpha) \leq 0 \quad\text{with}\quad \|\boldsymbol{s}\| = \sqrt{\boldsymbol{s}\cdot\boldsymbol{s}}\,. \tag{2.70}$$

The deviatoric stresses are used since the von Mises model assumes incompressible plastic flow. The value Y_0 describes the yield limit, H is the hardening modulus and α the hardening variable for which the evolution equation

$$\dot{\alpha} = \sqrt{\frac{2}{3}}\dot{\gamma} \tag{2.71}$$

holds.

Within this plasticity model the variables $\boldsymbol{\varepsilon}^p$ and α define the plastic state of the material. The latter variable is also known as equivalent plastic strain. These unknowns have to be computed from the above set of equations using a time integration scheme which will be discussed in more detail in Chap. 8.

2.2.3.2 Finite Strain Elasto-Plasticity

In forming and other technical operations metals can undergo finite strain states. These have to be considered in predictions using numerical methods. Thus the small strain von Mises plasticity model has to be extended to finite strains. Our description includes now isotropic nonlinear hardening behaviour.

The basic kinematical quantities were already defined in Sect. 2.1.1. These relations have to be extended for the modeling of finite strain plasticity. In finite deformation plasticity the deformation gradient (2.2) is split into an elastic—labeled as $\Box_e$—and plastic part—labeled as $\Box_p$—

$$\boldsymbol{F} = \boldsymbol{F}_e \, \boldsymbol{F}_p \, . \tag{2.72}$$

The right Cauchy-Green tensor is defined in (2.5) and the Jacobian of the deformation map given in (2.3). They are now rewritten using the multiplicative split (2.72). In case of von Mises plasticity the deformation is restricted by the plastic incompressibility condition $J_p = \det \boldsymbol{F}_p = 1$. With this constraint the determinant $J = J_e J_p$ from (2.72) can be written as $J = J_e = \det \boldsymbol{F}_e$. Inserting the split in (2.72) into (2.5) the elastic part of the right Cauchy-Green tensor yields

$$\boldsymbol{C} = \boldsymbol{F}_p^T \boldsymbol{F}_e^T \boldsymbol{F}_e \, \boldsymbol{F}_p = \boldsymbol{F}_p^T \boldsymbol{C}_e \, \boldsymbol{F}_p \longrightarrow \boldsymbol{C}_e = \boldsymbol{F}_p^{-T} \boldsymbol{C} \, \boldsymbol{F}_p^{-1} \, . \tag{2.73}$$

In the same way, the elastic part of the left Cauchy-Green tensor, see (2.6), $\boldsymbol{b}_e = \boldsymbol{F}_e \boldsymbol{F}_e^T$ can be expressed by the total deformation gradient and the inverse of the plastic part of the right Cauchy-Green tensor $\boldsymbol{C}_p = \boldsymbol{F}_p^T \, \boldsymbol{F}_p$. This leads to the expression

$$\boldsymbol{b}_e = \boldsymbol{F} \, \boldsymbol{C}_p^{-1} \, \boldsymbol{F}^T \, . \tag{2.74}$$

From (2.74) the elastic part of the Jacobian can be computed: $J_e = \sqrt{\det \boldsymbol{b}_e}$.

Using a strain energy function, like the one in (2.55), for the elastic part of the deformation the Kirchhoff stress follow by differentiation. For isotropic material response the Kirchhoff stresses can be computed using (2.54)

$$\boldsymbol{\tau} = 2 \boldsymbol{b}_e \frac{\partial \Psi_b}{\partial \boldsymbol{b}_e} \tag{2.75}$$

where Ψ_b is defined in (2.56).

The elasto-plastic model requires additionally the formulation of a yield function, a hardening law and an evolution equation for the plastic variables. The elastic region of the deformation is restricted by the yield function. For J_2-plasticity with isotropic linear harding and nonlinear saturation hardening the yield function can be written as

$$f(\boldsymbol{s},\alpha) = \sigma_{VM} - [\,Y_0 + (Y_\infty - Y_0)e^{-\delta\,\alpha} + H\,\alpha\,] \le 0, \tag{2.76}$$

where the variable Y_0 denotes the initial yield limit, H is the hardening modulus for linear hardening, Y_∞ and δ define nonlinear saturation hardening and α is the hardening variable. The von Mises stress σ_{VM} is defined as $\sigma_{VM} = \sqrt{\frac{3}{2}}\,\|\boldsymbol{s}\|$ with the deviatoric part

$$\boldsymbol{s} = \boldsymbol{\tau} - \frac{1}{3}\mathrm{tr}(\boldsymbol{\tau})\,\boldsymbol{1} \tag{2.77}$$

of the Kirchhoff stress, see e.g. (2.54).

The evolution equations for the plastic variables are, see e.g. Simo and Miehe (1992); Wriggers et al. (1992), Wriggers (2008),

$$-\frac{1}{2}\mathcal{L}_v\boldsymbol{b}_e = \dot\gamma\,\boldsymbol{n}\,\boldsymbol{b}_e \qquad \text{with } \boldsymbol{n} = \frac{\partial f}{\partial \boldsymbol{s}}\,, \tag{2.78}$$

$$\dot\alpha = \dot\gamma\,, \tag{2.79}$$

where $\mathcal{L}_v$ denotes the Lie derivative in time in the first equation and the second equation is the evolution equation for the hardening variable. The evolution equation (2.78) can be recast with (2.74) in an alternative form, see e.g. Korelc and Stupkiewicz (2014),

$$\dot{\boldsymbol{C}}_p^{-1} = -2\,\dot\gamma\,\boldsymbol{F}^{-1}\,\boldsymbol{n}\,\boldsymbol{F}\,\boldsymbol{C}_p^{-1} \tag{2.80}$$

which will be used later for the algorithmic treatment of plasticity within the numerical solution algorithm. The Karush-Kuhn-Tucker conditions (KKT) for the elasto-plastic model are

$$f \le 0\,, \qquad \dot\gamma \ge 0 \qquad \text{and } f\,\dot\gamma = 0\,. \tag{2.81}$$

2.3 Variational Formulation

The development of a discretization scheme, like the finite or virtual element method, generally relies on the weak form of equilibrium. It is also possible to use as a starting point the potential energy function. The latter is classically restricted to elasticity problems that are path independent. However, it is possible to construct for many problems a *pseudo*-potential which first variation yields the relevant weak form.

2.3.1 Potential and Weak Form

The potential energy of a solid can be written as

$$U(\boldsymbol{u}) = \int\limits_{\Omega} \left[\Psi(\boldsymbol{u}) - \bar{\boldsymbol{f}} \cdot \boldsymbol{u} \right] \, d\Omega - \int\limits_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \boldsymbol{u} \, d\Gamma \,. \tag{2.82}$$

Here $\Psi(\boldsymbol{u})$ is the elastic strain energy of the solid, $\bar{\boldsymbol{f}} \cdot \boldsymbol{u}$ is the work of the body forces and $\bar{\boldsymbol{t}} \cdot \boldsymbol{u}$ the work of the applied surface tractions. Here and in the following we use instead of $\varrho_0 \, \bar{\boldsymbol{b}}$ that appears in the momentum balance (2.19) the abbreviation $\bar{\boldsymbol{f}}$ for the body force.

The first variation of (2.82) yields the weak form of equilibrium

$$\delta U(\boldsymbol{u}) = \int\limits_{\Omega} \left[\frac{\partial \Psi(\boldsymbol{u})}{\partial \boldsymbol{F}} \cdot \delta \boldsymbol{F} - \bar{\boldsymbol{f}} \cdot \delta \boldsymbol{u} \right] d\Omega - \int\limits_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \delta \boldsymbol{u} \, d\Gamma \,. \tag{2.83}$$

Note that the space of test functions (variations) V which is given by $\boldsymbol{v} = \delta \boldsymbol{u}$ on Ω has to satisfy $\boldsymbol{v} = \boldsymbol{0}$ on Γ_D. From thermodynamic considerations one can deduce that by introducing a strain energy function $\Psi(\boldsymbol{u})$ for elastic problems the first Piola-Kirchhoff stresses follow from

$$\boldsymbol{P}(\boldsymbol{u}) = \frac{\partial \Psi(\boldsymbol{u})}{\partial \boldsymbol{F}} \,. \tag{2.84}$$

Using this relation now in (2.83), the weak form can be rewritten as[4]

$$a(\boldsymbol{u} \, , \boldsymbol{v}) = f(\boldsymbol{v}) \tag{2.85}$$

with

$$a(\boldsymbol{u} \, , \boldsymbol{v}) = \int\limits_{\Omega} \boldsymbol{P}(\boldsymbol{u}) \cdot \boldsymbol{F}(\boldsymbol{v}) \, d\Omega \,, \tag{2.86}$$

$$f(\boldsymbol{v}) = \int\limits_{\Omega} \bar{\boldsymbol{f}} \cdot \boldsymbol{v} \, d\Omega + \int\limits_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \boldsymbol{v} \, d\Gamma \,. \tag{2.87}$$

With the introduction of the second Piola-Kirchhoff stress, see (2.26), $\boldsymbol{S}(\boldsymbol{u}) = \boldsymbol{F}^{-1}(\boldsymbol{u}) \boldsymbol{P}(\boldsymbol{u})$, it follows $\boldsymbol{P}(\boldsymbol{u}) \cdot \boldsymbol{F}(\boldsymbol{v}) = \boldsymbol{F}(\boldsymbol{u}) \boldsymbol{S}(\boldsymbol{u}) \cdot \boldsymbol{F}(\boldsymbol{v}) = \boldsymbol{S}(\boldsymbol{u}) \cdot \boldsymbol{F}(\boldsymbol{u})^T \boldsymbol{F}(\boldsymbol{v})$. Thus the term $a(\boldsymbol{u} \, , \boldsymbol{v})$ can be equivalently formulated as

[4] This result can also be obtained by multiplying (2.19) with the test function $\boldsymbol{v}$, then integrating over the solid Ω and by using Gauss integral theorem. Note however that relation (2.85) is then obtained without using a constitutive equations and thus (2.85) can be applied to arbitrary materials.

$$a(\boldsymbol{u}\,,\boldsymbol{v}) = \int\limits_{\Omega} \frac{1}{2}\boldsymbol{S}(\boldsymbol{u}) \cdot \boldsymbol{C}(\boldsymbol{u}\,,\boldsymbol{v})\,\mathrm{d}\Omega \tag{2.88}$$

where the variation of the right Cauchy-Green tensor is computed from (2.5) as

$$\boldsymbol{C}(\boldsymbol{u}\,,\boldsymbol{v}) = \boldsymbol{F}(\boldsymbol{u})^T\boldsymbol{F}(\boldsymbol{v}) + \boldsymbol{F}(\boldsymbol{v})^T\boldsymbol{F}(\boldsymbol{u})\,.$$

2.3.2 Incompressibility

In case of an incompressible material the potential energy is given by

$$U^i(\boldsymbol{u}) = \int\limits_{\Omega}\left[\Psi^i(\boldsymbol{u}) - \bar{\boldsymbol{f}}\cdot\boldsymbol{u}\,\right]\mathrm{d}\Omega + \int\limits_{\Omega} p\,[\,J(\boldsymbol{u}) - 1\,]\,\mathrm{d}\Omega - \int\limits_{\Gamma_N}\bar{\boldsymbol{t}}\cdot\boldsymbol{u}\,\mathrm{d}\Gamma \tag{2.89}$$

where the incompressibility constraint $J(\boldsymbol{u}) - 1 = 0$ is added via a Lagrangian parameter p which can be interpreted as the pressure.

A different formulation is based on a Hu-Washizu variational formulation for the incompressibility constraint. It can be used for nearly incompressible materials

$$\begin{aligned} U^i_{HW}(\boldsymbol{u}\,,p\,,\Theta) &= \int\limits_{\Omega}\left[\Psi^i(\boldsymbol{u}) - \bar{\boldsymbol{f}}\cdot\boldsymbol{u}\,\right]\mathrm{d}\Omega \\ &+ \int\limits_{\Omega}\left\{p\,[\,J(\boldsymbol{u}) - \Theta\,] + \frac{K}{2}(\Theta - 1)^2\right\}\mathrm{d}\Omega - \int\limits_{\Gamma_N}\bar{\boldsymbol{t}}\cdot\boldsymbol{u}\,\mathrm{d}\Gamma \end{aligned} \tag{2.90}$$

where p again is the pressure and Θ associated with the volume dilatation. Once the bulk modulus K approaches infinity for a Poisson ratio $\nu \to 0.5$ incompressible behaviour is recovered. Formulation (2.89) was applied in Simo et al. (1985a) to construct special finite elements for finite deformation elasticity. It is now possible to use different ansatz orders for the variables $(\boldsymbol{u}\,,p\,,\Theta)$.

2.3.3 Plasticity

For finite strain elasto-plasticity classically the weak form of equilibrium is used as a starting point for the development of a discretization method. However, it is computationally more efficient to start the development of the elasto-plastic virtual element from a *pseudo* potential energy function directly instead of using the weak form, see Korelc and Stupkiewicz (2014). This is especially advantageous when the code is automatically generated using the software tool *AceGen*, see Korelc and Wriggers (2016). The *pseudo*-potential energy depends on elastic variables, see Sect. 2.2.3.2, and includes the plastic history variables ($\mathbf{h} = \{\boldsymbol{C}_p^{-1}, \gamma\}$)

$$U(\boldsymbol{u}, \mathbf{h}) = \int\limits_{\Omega} \Big[\Psi(\boldsymbol{u}, \mathbf{h}) - \overline{\boldsymbol{f}} \cdot \boldsymbol{u} \Big] \, \mathrm{d}\Omega - \int\limits_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \boldsymbol{u} \, \mathrm{d}\Gamma \,. \tag{2.91}$$

During the first variation of the potential with respect to the displacement field, the plastic part of the deformation is kept constant[5] which yields

$$\delta U(\boldsymbol{u}, \mathbf{h}) = \left. \frac{\partial U}{\partial \boldsymbol{u}} \right|_{\mathbf{h}=const.} \cdot \delta \boldsymbol{u} \tag{2.92}$$

where $\delta \boldsymbol{u}$ denotes the variation of $\boldsymbol{u}$.

2.3.4 Heat Conduction

The weak form related to pure heat conduction in the stationary case follows from (2.42) by multiplying this partial differential equation by a test function ϑ and integrating over the volume. Together with the application of the theorem of Gauss and Fourier's law one obtains

$$\int\limits_{\Omega} \big[\mathrm{Grad}\, \vartheta \cdot K_\theta \, \mathrm{Grad}\, \theta - \vartheta \, \varrho_0 R \, \big] \, \mathrm{d}\Omega + \int\limits_{\Gamma_N} \vartheta \, \bar{\boldsymbol{Q}} \cdot \boldsymbol{n} \, \mathrm{d}\Gamma = 0 \tag{2.93}$$

where K_θ is the thermal conductivity, $\varrho_0 R$ a volume source and $\boldsymbol{Q}$ the heat flux through the boundary with the outward normal $\boldsymbol{n}$.

In this special case one can formulate a potential function

$$Q(\theta) = \int\limits_{\Omega} \left[\frac{K_\theta}{2} (\mathrm{Grad}\, \theta \cdot \mathrm{Grad}\, \theta) - (\varrho_0 R) \, \theta \right] \mathrm{d}\Omega + \int\limits_{\Gamma_N} (\bar{\boldsymbol{Q}} \cdot \boldsymbol{n}) \, \theta \, \mathrm{d}\Gamma \tag{2.94}$$

from which the weak (2.93) follows by minimization.

Remark 2.1 The heat conduction equation (2.42) actually is also well known as Poisson equation and has many applications in applied engineering and physics. Among these applications are torsion problems, potential flow, electro statics, membranes and many more. The only thing that changes is the interpretation of the

[5] This formulation is equivalent to using the weak form directly since the variation of (2.91) with frozen history variables **h** yields exactly the classical weak form (2.85).

constants. E.g. for a membrane $\theta \rightarrow w$ is the deflection, $K_\theta \rightarrow \sigma_0$ becomes the prestress of the membrane, $\varrho_0 R \rightarrow f$ is then associated with a transverse load and $\bar{\boldsymbol{Q}} \cdot \boldsymbol{n} \rightarrow \nabla w \cdot \boldsymbol{n}$ is the slope of w at the boundary in normal direction which is zero for a support $w = 0$ at the entire boundary Γ. ■

References

Chadwick, P. 2012. *Continuum mechanics: Concise theory and problems*. Courier Corporation.

de Souza Neto, E.A., D. Peric, and D.R.J. Owen. 2008. *Computational methods for plasticity, theory and applications*. Chichester: Wiley.

Eringen, A. 1967. *Mechanics of continua*. New York, London, Sidney: Wiley.

Gurtin, M.E., E. Fried, and L. Anand. 2010. *The mechanics and thermodynamics of continua*. Cambridge University Press.

Holzapfel, G.A. 2000. *Nonlinear solid mechanics*. Chichester: Wiley.

Hudobivnik, B. 2016. *Automatic differentiation based solution of strongly coupled problems in engineering*. Ph.D. thesis, University of Ljubljana, Faculty of Civil and Geodetic Engineering.

Korelc, J., and S. Stupkiewicz. 2014. Closed-form matrix exponential and its application in finite-strain plasticity. *International Journal for Numerical Methods in Engineering* 98: 960–987.

Korelc, J., and P. Wriggers. 2016. *Automation of finite element methods*. Berlin: Springer.

Malvern, L.E. 1969. *Introduction to the mechanics of a continuous medium*. Englewood Cliffs, New Jersey: Prentice-Hall.

Marsden, J.E., and T.J.R. Hughes. 1983. *Mathematical foundations of elasticity*. Englewood Cliffs, New Jersey: Prentice-Hall.

Mooney, M. 1940. A theory of large elastic deformations. *Journal for Applied Physics* 11: 582–592.

Ogden, R.W. 1984. *Non-linear elastic deformations*. Chichester: Ellis Horwood and Wiley.

Schröder, J. 2009. Anisotropic polyconvex energies. In *Polyconvex analysis*, ed. J. Schröder, 1–53. CISM. Wien: Springer. 62.

Schröder, J., and P. Neff. 2003. Invariant formulation of hyperelastic transverse isotropy based on polyconvex free energy functions. *International Journal of Solids and Structures* 40 (2): 401–445.

Shabana, A.A. 2018. Computational continuum mechanics. Wiley.

Simo, J.C., and T.J.R. Hughes. 1998. *Computational inelasticity*. New York, Berlin: Springer.

Simo, J.C., and C. Miehe. 1992. Associative coupled thermoplasticity at finite strains: formulation, numerical analysis and implementation. *Computer Methods in Applied Mechanics and Engineering* 98: 41–104.

Simo, J.C., R.L. Taylor, and K.S. Pister. 1985. Variational and projection methods for the volume constraint in finite deformation elasto-plasticity. *Computer Methods in Applied Mechanics and Engineering* 51: 177–208.

Tadmor, E.B., R.E. Miller, and R.S. Elliott. 2012. Continuum mechanics and thermodynamics: From fundamental concepts to governing equations. Cambridge University Press.

Truesdell, C., and R. Toupin. 1960. The classical field theories. In Handbuch der Physik III/1. Berlin, Heidelberg, Wien: Springer.

Truesdell, C., and W. Noll. 1965. The nonlinear field theories of mechanics. In *Handbuch der Physik III/3*, ed. S. Flügge. Berlin, Heidelberg, Wien: Springer.

Wriggers, P. 2008. *Nonlinear finite elements*. Berlin, Heidelberg, New York: Springer.

Wriggers, P., C. Miehe, M. Kleiber, and J. Simo. 1992. A thermomechanical approach to the necking problem. *International Journal for Numerical Methods in Engineering* 33: 869–883.

Wriggers, P., J. Schröder, and F. Auricchio. 2016. Finite element formulations for large strain anisotropic materials. *International Journal on Advanced Modelling and Simulation in Engineering Sciences* 3 (25): 1–18.

Chapter 3
VEM Ansatz Functions and Projection for Solids

Virtual elements have so far been formulated for different problems in applied engineering and physics. One of the first papers was on the general mathematical formulation of the method, see Beirão da Veiga et al. (2013a). Then the method was subsequently applied to problems related to elliptic, hyperbolic and parabolic partial differential equations. Starting with the Laplace equation, see Beirão da Veiga et al. (2014), and virtual element formulations for elliptical equations, like elasticity, see e.g Beirão da Veiga et al. (2018). In the sequence problems related to diffusion, see Beirão da Veiga et al. (2014), or Cahn-Hillard equations, see Antonietti et al. (2016), were developed.

A large amount of work was related to solid mechanics. Here problems of elasticity for small and finite strain, see Beirão da Veiga et al. (2013b), Beirão da Veiga et al. (2015), Artioli et al. (2017a) and Wriggers et al. (2017) as well as inelastic responses, see e.g. Artioli et al. (2017b), Wriggers and Hudobivnik (2017) and Aldakheel et al. (2019b) were solved. But also contact problems Wriggers et al. (2016a), Kirchhoff and Reissner-Mindlin plate problems, see Brezzi and Marini (2013), Chinosi and Marini (2016), Beirão da Veiga et al. (2019c) and Wriggers et al. (2022), and problems regarding damage and fracture Aldakheel et al. (2018a), De Bellis et al. (2018) and Hussein et al. (2019) were considered. The beauty of the virtual element method is related to the possibility to use arbitrary element shapes which can be even non convex. In addition to that elements can be defined by an arbitrary number of nodes. Both properties ensure an extreme flexibility for applying virtual elements in engineering. In this chapter we will focus on ansatz spaces that need to be constructed for applications of virtual elements in the classical area of solid mechanics including small and large deformations.

The space of ansatz function for the virtual element method (VEM) is different from the finite element method (FEM). However it leads in the end to a virtual element that has the same nodal degrees of freedom as a finite element which actually enables a consistent and immediate coupling of virtual and finite elements within one mesh. Due to the possibility to formulate virtual elements for arbitrarily shaped

P. Wriggers et al., *Virtual Element Methods in Engineering Sciences*,
https://doi.org/10.1007/978-3-031-39255-9_3

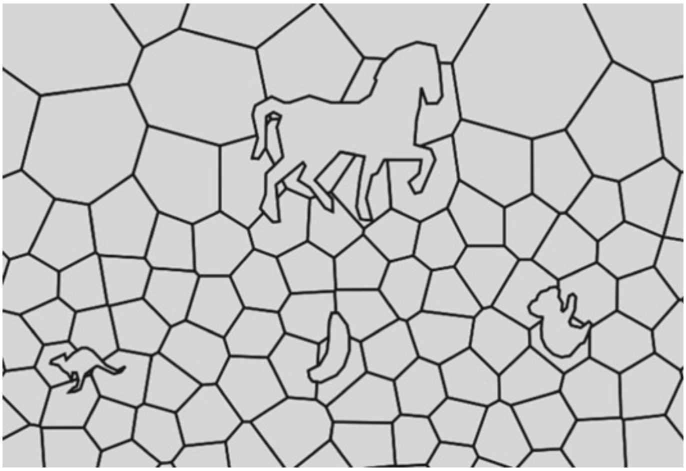


Fig. 3.1 Virtual element mesh with arbitrarily shaped elements

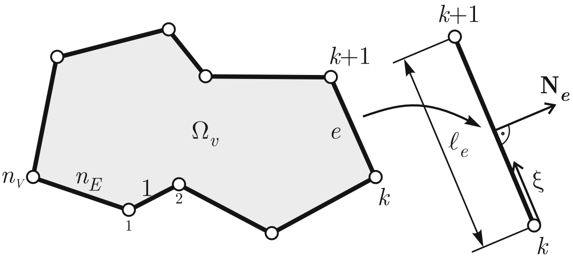
geometries, see e.g. the animals in Fig. 3.1, the space of ansatz function has to be selected in a special way. It contains polynomials and a other functions that need not to be computed explicitly. The main ingredient of the virtual element method is a projection operator which has to be constructed by a projection of the ansatz space onto a space of polynomials. In this chapter we will outline how this operator is computed for linear and quadratic polynomial ansatz spaces.

3.1 Two-Dimensional Case

The domain Ω is partitioned into non-overlapping polygonal elements Ω_v with a boundary Γ_v consisting of straight edges. The elements need not to be convex.

In Fig. 3.2 n_E, represents the number of edges of a polygon Ω_v. An edge, with length l_e, is denoted within the polygon by Γ_e where e is an index $1 \leq e \leq n_E$. A

Fig. 3.2 Virtual element and boundary discretization



normal $\mathbf{N}_e$ is related to each of the straight edges. A sample element is depicted in Fig. 3.2 together with its vertices k $(1 \le k \le n_V)$.

3.1.1 General Ansatz Space

As in the finite element method, the ansatz space $\mathbf{V}_h$ will be defined at the element level Ω_v. For that a local space $\mathbf{V}_{h|\Omega_v}$ is introduced which is associated with the degrees of freedom at the nodes and within the element, like in the finite element methodology. However contrary to the finite element method, the local space for virtual elements cannot be written explicitly. The finite dimensional space $V_{h|\Omega_v}$ is given for dimension $d = 2$ by

$$V_{h|\Omega_v} = \left\{ \boldsymbol{u}_h \in [H^1(\Omega_v)]^2 \, : \, \boldsymbol{u}_h|_{\Gamma_e} \in \mathsf{P}_n(\Gamma_e) \, \forall \, \Gamma_e \in \Gamma_v \, , \, \Delta \boldsymbol{u}_h \in \mathsf{P}_{n-2}(\Omega_v) \right\} \quad (3.1)$$

following Beirão da Veiga et al. (2013b).

The ansatz function for the displacement field $\boldsymbol{u}_h$ in two dimensions should have the properties

(a0): $\boldsymbol{u}_h$ is known at the vertices k of the polygon Ω_v,
(a1): $\boldsymbol{u}_h$ is a polynomial P_n of degree n at each edge $\Gamma_e \in \Gamma_v$ with values of $\boldsymbol{u}_h$ at $n-1$ equally spaced points,
(a2): $\boldsymbol{u}_h$ is continuous at all edges $\Gamma_e \in \Gamma_v$ of the polygon Ω_v,
(a3): Div(Grad $\boldsymbol{u}_h$) $= \Delta \boldsymbol{u}_h$ is a polynomial of degree P_{n-2} on the polygon Ω_v.

With these definitions the ansatz for the displacement field $\boldsymbol{u}_h$ is a harmonic function inside Ω_v which is only known at the edges Γ_e of Ω_v, but the ansatz is not known inside of the polygon Ω_v. This is why the resulting elements are called virtual. Additionally we have $[\mathsf{P}_n]^2 \subseteq \mathbf{V}_h$. All these properties guarantee the convergence of the virtual element method. For mathematical details see Beirão da Veiga et al. (2013b).

In the two dimensional case of elasticity the total number of unknowns for a virtual element is given by the above relations:

(a0): → $2\, n_V$ unknowns related to the n_V vertices of the polynomial,
(a1): → $2\, n_E (n-1)$ unknowns are related to nodes at each edge Γ_e,
(a3): → $2\, \frac{n\,(n-1)}{2}$ unknown variables result from the polynomial space P_{n-2}, these unknowns are called moments in virtual element technologies and represent a field in Ω_v in an integral sense.

The latter unknowns are not related to a specific location in Ω_v and the introduction is not obvious, but will become clear in the next steps of the construction of the ansatz functions for the virtual element method.

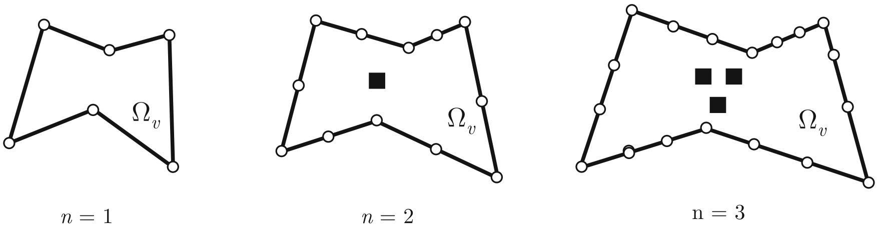


Fig. 3.3 VEM elements with different order of polynomial ansatz

Since $n_V = n_E$ the total number of unknowns is given by

$$2\, n_V\, n + n\, (n - 1) \tag{3.2}$$

where the last summand stands for the degree of freedoms associated with the moments. For the special cases of linear ($n = 1$), quadratic ($n = 2$) and cubic ($n = 3$) polynomials P_n the number of unknowns within a virtual element Ω_v are

$$\begin{aligned} n = 1: &\qquad 2\, n_V \\ n = 2: &\qquad 4\, n_V + 2 \\ n = 3: &\qquad 6\, n_V + 6 \end{aligned}$$

Moments do not appear in a virtual element space that is linear since with $n = 1$ it follows $\mathsf{P}_{-1}(\Omega_v) = \{0\}$. Figure 3.3 shows the nodes and moments for a linear ($n = 1$), quadratic ($n = 2$), and cubic ($n = 3$) ansatz. The nodes at the boundary are depicted as "○" and the moments in the interior as "■" for a virtual element Ω_v.

Ansatz functions In general one can define an ansatz function which is directly formulated in the coordinates that describe the real geometry of the domain. This is different from classical finite element methods where in most cases isoparametric ansatz spaces are introduced that are defined on a reference element and then mapped to the real geometry of the domain. Hence a possible ansatz for $V_{h|\Omega_v}$ is provided by[1]

$$\mathbf{N}_\pi^n = \left\langle 1\; X\; Y\; Z\; X^2\; XY\; Y^2\; YZ\; Z^2\; ZX\; \ldots\; Z^n \right\rangle \tag{3.3}$$

[1] In the mathematical literature, see e.g. Beirão da Veiga et al. (2013b), the polynomial (3.3) is often defined in terms of scaled polynomials

$$\bar{X} = \frac{X - X_c}{l_\Omega} \quad \bar{Y} = \frac{Y - Y_c}{l_\Omega} \quad \bar{Z} = \frac{Z - Z_c}{l_\Omega}$$

where X_c, Y_c and Z_c are the coordinates of the barycentre of the element and l_Ω is a length parameter related to the element size. This approach can have some advantages when evaluating the element matrices and eventually leads to tangent matrices with smaller condition numbers.

which is a complete polynomial of order n and has a length $l_n = \text{length}(\mathbf{N}_\pi^n)$. However the ansatz $\boldsymbol{u}_h$ is only known at the edges. Thus a specific approach has to be considered to find a connection between the nodal displacements $\boldsymbol{u}_k$ at the vertices and edges at boundary Γ_v and the strain field inside the virtual element Ω_v.

To introduce an explicit ansatz for the part (a1) we split the total boundary Γ_v of the virtual element into straight edges Γ_e. On each of these edges we can introduce the displacement field as

$$\boldsymbol{u}_{h|\Gamma_e} = \sum_{i=1}^{n+1} M_i(\xi)\,\mathbf{u}_i \qquad \text{on } \Gamma_e \in \Gamma_v \tag{3.4}$$

where M_i is the interpolation function for the displacements along the edge, ξ is a local coordinate defined at the edge Γ_e, see Fig. 3.2, and n is the polynomial degree of the ansatz. Explicit function for $M_i(\xi)$ can be found in many textbooks on finite elements, see e.g. Wriggers (2008) and Oñate (2009). Examples for $n = 1, 2$ are given below

$$n = 1: \qquad M_1 = 1 - \xi\,, \quad M_2 = \xi \tag{3.5}$$

$$n = 2: \qquad M_1 = 2\xi^2 - 3\xi + 1\,, \quad M_2 = (2\xi - 1)\xi\,, \quad M_3 = 4\xi(1 - \xi) \tag{3.6}$$

for $0 \leq \xi \leq 1$, see Fig. 3.2. These ansatz functions have the delta property: $M_i(\xi_k) = \delta_{ik}$, where ξ_k ist the nodal coordinate.

By using the same ansatz for an edge in all of the virtual elements used to discretize a engineering problem (a2) is fulfilled automatically.

Next we introduce moments $\mathbf{m}_{(\alpha,\beta)}$ in accordance with (a3). These moments appear only for $n > 1$ and are integrals of the single terms of a complete polynomial of degree $n - 2$. Thus we obtain the unknowns

$$n = 2: \qquad \mathbf{m}_{(0,0)} = \mathbf{m}_1 = \frac{1}{\Omega_v}\int_{\Omega_v} \boldsymbol{u}_h \, \mathrm{d}\Omega \tag{3.7}$$

$$n = 3: \quad \mathbf{m}_{(1,0)} = \mathbf{m}_2 = \frac{1}{\Omega_v}\int_{\Omega_v} X\,\boldsymbol{u}_h \, \mathrm{d}\Omega \qquad \mathbf{m}_{(0,1)} = \mathbf{m}_3 = \frac{1}{\Omega_v}\int_{\Omega_v} Y\,\boldsymbol{u}_h \, \mathrm{d}\Omega \tag{3.8}$$

and so forth where n is the order of the ansatz polynomial. Note that the complete ansatz for the moments for $n = 3$ contains the three terms $\mathbf{m}_{(0,0)}$, $\mathbf{m}_{(1,0)}$ and $\mathbf{m}_{(0,1)}$. The need to introduce the moments $\mathbf{m}_{(\alpha,\beta)}$, respectively $\mathbf{m}_\gamma$, will become clear in Sects. 3.1.2 and 3.1.6. Ansatz (3.4) together with the moments $\mathbf{m}_{(\alpha,\beta)}$ is now used in the next section to compute an approximation of the gradient needed for the formulation of the weak form.

Approximation of $\boldsymbol{u}_h$ Neither the function $\boldsymbol{u}_h$ nor its gradient $\nabla\boldsymbol{u}_h$ can be obtained explicitly from the definition of the space $V_{h|\Omega_v}$. Thus it is not possible to evaluate the weak form (2.85) of a solid in a standard way, like in a finite element environment. To connect the ansatz defined in (a0) to (a3) to the strains (depending on the gradient of the displacement field) in the virtual element Ω_v a special projection onto the polynomial ansatz space has to be designed, see Beirão da Veiga et al. (2013b). The projection

$$\begin{aligned} \Pi : V_{h|\Omega_v} &\longrightarrow \left[\mathsf{P}_n(\Omega_v)\right]^2 \\ \boldsymbol{u}_h &\mapsto \quad \Pi(\boldsymbol{u}_h) = \boldsymbol{u}_\pi \end{aligned} \tag{3.9}$$

approximates $\boldsymbol{u}_h$ by $\boldsymbol{u}_\pi$. It is based on the orthogonality relation of the gradients

$$\int\limits_{\Omega_v} \nabla\boldsymbol{u}_\pi \cdot \nabla\mathbf{N}_\pi^n \mathrm{d}\Omega = \int\limits_{\Omega_v} \nabla\boldsymbol{u}_h \cdot \nabla\mathbf{N}_\pi^n \mathrm{d}\Omega \tag{3.10}$$

using the polynomial (3.3) of ansatz order n. Since this projection excludes the constant terms the computation of the average values

$$\int\limits_{\Omega_v} \boldsymbol{u}_\pi \mathrm{d}\Gamma = \int\limits_{\Omega_v} \boldsymbol{u}_h \mathrm{d}\Gamma \tag{3.11}$$

completes the projection which has the property: $\Pi(\mathbf{N}_\pi^n) = \mathbf{N}_\pi^n \;\forall\, \mathbf{N}_\pi^n \in \left[\mathsf{P}_n(\Omega_v)\right]^2$, see also Beirão da Veiga et al. (2014).

For the explicit computation of the projection we will introduce different possibilities[2] throughout the next sections.

With the above the projection $\boldsymbol{u}_\pi$ can be approximated by

$$\boldsymbol{u}_\pi = \mathbf{A}\,\mathbf{N}_\pi^{n\,T} \tag{3.12}$$

where $\mathbf{N}_\pi^n$ is a polynomial of the ansatz order n and $\mathbf{A}$ are the unknown coefficients that will be determined from (3.10) and (3.11). The gradient $\nabla\boldsymbol{u}_\pi$ follows from

$$\nabla\boldsymbol{u}_\pi = \hat{\mathbf{A}}\,\nabla\mathbf{N}_\pi^{n\,T} \tag{3.13}$$

where $\hat{\mathbf{A}}$ contains now the non zero parameters of the ansatz (3.12). It is obvious that with this ansatz the left hand side of (3.10) is computable. However the right hand

[2] Such projection can also be performed directly for the strains (2.9), see e.g. Gain et al. (2014). However selecting the displacement gradient instead of a strain measure is more general, especially in view of finite deformations where the deformation gradient depends directly on the displacement gradient: $\boldsymbol{F}_\pi = \mathbf{1} + \nabla\boldsymbol{u}_\pi$. This allows to compute any strain measure needed in the continuum formulation, see Sect. 2.1.1.

side cannot be determined because $\nabla \boldsymbol{u}_h$ is not known within the element. Using the Gaussian theorem we obtain with $\mathrm{Div}[\nabla \mathbf{N}_\pi^n\,] = \Delta \mathbf{N}_\pi^n$ for the right hand side

$$\int\limits_{\Omega_v} \nabla \boldsymbol{u}_h \cdot \nabla \mathbf{N}_\pi^n \mathrm{d}\Omega = -\int\limits_{\Omega_v} \Delta \mathbf{N}_\pi^n \cdot \boldsymbol{u}_h \mathrm{d}\Omega + \int\limits_{\Gamma_v} (\boldsymbol{u}_h \otimes \boldsymbol{N}_v) \cdot \nabla \mathbf{N}_\pi^n \mathrm{d}\Gamma \,. \qquad (3.14)$$

This form is now computable. The first term on the right hand side yields with $\Delta \mathbf{N}_\pi^n \in \left[\mathsf{P}_{n-2}(\Omega_v)\right]^2$ the moments defined in (3.7). With (a0) and (a1) $\boldsymbol{u}_h$ is defined at the boundary Γ_v which allows the evaluation of the second term on the right hand side. Thus the right hand side of (3.14) is a function of the unknowns $\mathbf{u}_i$, see (3.4), and the moments $\mathbf{m}_j$, see (3.7).

The projection (3.10) yields an equation system for the unknown coefficients $\hat{\mathbf{A}}$ depending on the nodal unknowns $\mathbf{u}_i$ and the moments $\mathbf{m}_j$ of the virtual element Ω_v. The constant part of the ansatz (3.12) follows from inserting this ansatz into (3.11).

Now the ansatz for the virtual element is complete and can formally be written as $\boldsymbol{u}_\pi = \mathbb{M}(X, Y)\,(\mathbf{u}_v, \mathbf{m}_j)$ where $\mathbb{M}(X, Y)$ is an operator that emerges from the solution of the linear system resulting from (3.10) and (3.11) and $(\mathbf{u}_i, \mathbf{m}_j)$ contains all nodal unknowns and moments.

The next section provides details for the determination of the operator $\mathbb{M}(X, Y)$ which can be viewed like a shape function in the finite element methodology.

3.1.2 Computation of the Projection

There exist several methods to obtain the parameters $\hat{a}_i$ and with that the gradient $\nabla \boldsymbol{u}_\pi$ in terms of the unknown nodal values of a virtual element Ω_v. Here we want to discuss two of them. The basis is on one hand the equivalence of weak forms written in terms of the projection $\boldsymbol{u}_\pi$ and the ansatz $\boldsymbol{u}_h$ and on the other hand a Galerkin projection for $\nabla \boldsymbol{u}_\pi$ which yields in a least square sense the best approximation of $\nabla \boldsymbol{u}_h$.

- **Projection using the weak form**. In the first approach the equivalence of the left side of the weak form (2.85) when using the projection $\nabla \boldsymbol{u}_\pi$ and the gradient of the ansatz $\nabla \boldsymbol{u}_h$

$$a(\boldsymbol{u}_\pi\,, \boldsymbol{v}_h) = a(\boldsymbol{u}_h\,, \boldsymbol{v}_h) \qquad (3.15)$$

  is the basis to determine the parameters of the projection $\hat{\mathbf{A}}$ in (3.13). Only the gradients of the projection play a role in this equation, since as well the stresses as the strains depend on gradients. Thus this equivalence only allows to compute the parameters $\hat{a}_i \in \hat{\mathbf{A}}$ of (3.13) that are related to terms which appear in the gradients. The part (3.15) of the weak form can be stated more explicity as

$$\int_{\Omega} \boldsymbol{\varepsilon}(\boldsymbol{v}_h) \cdot \mathbb{C}\boldsymbol{\varepsilon}(\boldsymbol{u}_\pi)\, \mathrm{d}\Omega = \int_{\Omega} \boldsymbol{\varepsilon}(\boldsymbol{v}_h) \cdot \mathbb{C}\boldsymbol{\varepsilon}(\boldsymbol{u}_h)\, \mathrm{d}\Omega \tag{3.16}$$

where $\varepsilon(\boldsymbol{v}_h) = \frac{1}{2}(\nabla \boldsymbol{v}_h + \nabla^T \boldsymbol{v}_h)$ is the symmetric tensor for small strains, see (2.9), and $\mathbb{C}$ is the constitutive matrix for linear elasticity. The other strains — $\boldsymbol{\varepsilon}(\boldsymbol{u}_\pi)$ and $\boldsymbol{\varepsilon}(\boldsymbol{u}_h)$ — are computed in the same way.
In equation (3.16) the test function $\boldsymbol{v}_h$ is approximated by a polynomial of degree n leading to $\nabla \boldsymbol{v}_h = \mathbf{B}_v\, \mathbf{c}$ where $\mathbf{B}_v = \mathbf{B}_\pi$ is a matrix[3] that contains the derivatives of the ansatz (3.12). The vector $\mathbf{c}$ contains unknown coefficients of the test function. In the end the coefficients $\mathbf{c}$ cancel out when solving (3.16) like the parameters of the virtual displacements in a weak form. Thus the solution is independent of $\mathbf{c}$.
Let us remark:

– The computation of the parameter $\hat{\mathbf{a}}$ can be performed in a straight forward manner in case that the weak form (2.85) is related to small strains and a linear elastic material. In that case it is advantageous to directly approximate the symmetric linear strains by

$$\boldsymbol{\varepsilon}_\pi = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{Bmatrix} = \mathbf{B}^{\varepsilon}_{\pi}(X, Y)\, \bar{\mathbf{a}}\,. \tag{3.17}$$

For more details, see e.g. Wriggers et al. (2016a) and Artioli et al. (2017a).
– In case of a finite strain problem the 1^{st} Piola-Kirchhoff stress $\boldsymbol{P}$ in the weak form (2.85) depends in a nonlinear fashion on the $\nabla \boldsymbol{u}$. Using then (3.15) yields a nonlinear equation system for the unknown parameters $\hat{\mathbf{a}}$ which is computationally extremely inefficient. Hence the use of (3.15) is not advisable for finite strain problems.

- **Galerkin projection of the displacement gradients.**The second approach employs a Galerkin projection of the gradients, see also (3.10), to obtain the parameters $\hat{\mathbf{a}}$ for the computation of $\nabla \boldsymbol{u}_\pi$

$$\int_{\Omega_v} \nabla \mathbf{N}_\pi \cdot (\nabla \boldsymbol{u}_h - \nabla \boldsymbol{u}_\pi)\ \mathrm{d}\Omega = 0\,. \tag{3.18}$$

This formulation does not rely on the weak form $a(\boldsymbol{u}, \boldsymbol{v})$ and thus avoids nonlinear kinematics and nonlinear constitutive relations.[4] Hence this formulation provides a manageable way to compute the projection for $\nabla \boldsymbol{u}_\pi$ in the general case. Thus (3.18) will be used as the standard projection in the following chapters.

[3] Matrix $\mathbf{B}_v$ is introduced analogous to the B-matrix used in finite element environments to generate a matrix formulation of the projection. Details will be provided in Sects. 3.1.4 and 3.1.6.

[4] This projection as well as the projection based on the weak form (3.15) lead to identical parameters $\hat{\mathbf{a}}$ for the linear elasticity problems at small strains.

A polynomial that has the same degree as $\boldsymbol{u}_\pi$ is selected to compute the gradient $\nabla\mathbf{N}_\pi$ which is applied as weighting function. Equation (3.18) leads to

$$\int\limits_{\Omega_v} \nabla\mathbf{N}_\pi \cdot \nabla\boldsymbol{u}_\pi \, \mathrm{d}\Omega = \int\limits_{\Omega_v} \nabla\mathbf{N}_\pi \cdot \nabla\boldsymbol{u}_h \, \mathrm{d}\Omega \tag{3.19}$$

The left hand side of the above equation can be rewritten by using the Gauss theorem

$$\int\limits_{\Omega_v} \nabla\mathbf{N}_\pi \cdot \nabla\boldsymbol{u}_h \, \mathrm{d}\Omega = \underbrace{\int\limits_{\Gamma} \nabla\mathbf{N}_\pi \cdot (\boldsymbol{u}_h \otimes \boldsymbol{N}) \, \mathrm{d}\Gamma}_{b_\Gamma} - \underbrace{\int\limits_{\Omega_v} \div[\nabla\mathbf{N}_\pi] \cdot \boldsymbol{u}_h \, \mathrm{d}\Omega}_{b_{div}} \tag{3.20}$$

where $\boldsymbol{N}$ is the outward normal related to Γ, see Fig. 3.2.

When using the second approach, ansatz (3.4) for $\boldsymbol{u}_h$ and ansatz (3.13) for $\nabla\boldsymbol{u}_\pi$ have to be inserted into (3.19) and (3.20).

The integral on the left hand side of (3.19) depends only on the polynomials and can be evaluated directly. Since $\nabla\mathbf{N}_\pi$ has the same polynomial order as $\nabla\boldsymbol{u}_\pi$ we can write $\nabla\mathbf{N}_\pi = \mathbf{B}_\pi(X, Y)\,\mathbf{c}$ with unknown coefficients $\mathbf{c}$. This leads to

$$\int\limits_{\Omega_v} \nabla\mathbf{N}_\pi \cdot \nabla\boldsymbol{u}_\pi \, \mathrm{d}\Omega = \mathbf{c}^T \int\limits_{\Omega_v} [\mathbf{B}_\pi(X, Y)]^T \, \mathbf{B}_\pi(X, Y) \, \mathrm{d}\Omega \, \hat{\mathbf{a}} \,. \tag{3.21}$$

The integral in (3.21) depends only on X and Y and can be computed exactly for any polygon using the relations in Appendix A which follow from Green's theorem.

The integral of the right hand side of (3.19) was split in two terms, b_Γ and b_{div}. The integral b_Γ can be computed using (3.4) as a sum of integrals over each edge Γ_e, see Fig. 3.2,

$$\int\limits_{\Gamma} \nabla\mathbf{N}_\pi \cdot (\boldsymbol{u}_h \otimes \boldsymbol{N}) \, \mathrm{d}\Gamma = \mathbf{c}^T \sum_{e=1}^{n_E} \int\limits_{\Gamma_e} [\mathbf{B}_\pi(X, Y)]^T \sum_{i=1}^{n+1} M_i(\xi) \begin{Bmatrix} u_{xi} N_{xe} \\ u_{xi} N_{ye} \\ u_{yi} N_{xe} \\ u_{yi} N_{ye} \end{Bmatrix} \mathrm{d}\Gamma \tag{3.22}$$

where the column vector in (3.22) represents the tensor product $\boldsymbol{u}_h \otimes \boldsymbol{N}_e$ with (N_{ye}, N_{ye}) being the components of the outward normal $\boldsymbol{N}_e$ at the edge Γ_e, see Fig. 3.2. Furthermore we note that the variable ξ defining the local shape functions of order n at the boundary can be expressed in terms of the coordinates $\boldsymbol{X} = (X, Y)$ by a linear mapping

$$\boldsymbol{X} = (1 - \xi)\,\mathbf{X}_1 + \xi\,\mathbf{X}_2 \longrightarrow \xi = \frac{1}{l_e^2}(\boldsymbol{X} - \mathbf{X}_1) \cdot (\mathbf{X}_2 - \mathbf{X}_1) \tag{3.23}$$

since the edges are assumed to be straight. Here $\mathbf{X}_1$ and $\mathbf{X}_2$ are the nodal coordinates of the vertices and l_e is the length of the edge Γ_e.

We note that the polynomial degree to be integrated is a product of polynomials of order $n-1$ related to matrix $\mathbf{B}_\pi$ in (3.22), and of order n for the ansatz functions M_i at the edge. In total one has to integrate over polynomials of degree $n+(n-1)$. In the linear case, $n=1$, only linear terms appear in (3.22). For a quadratic ansatz, $n=2$, polynomials up to third order have to be integrated. Both of these cases can most simply, but accurately, integrated with the trapezoidal and the Gauss-Lobatto rule, respectively. Due to the delta property of M_i and location of the integration points at the nodes only the values (u_{xi}, u_{yi}) at the vertices have to be considered in the numerical integration for $n=1,2$ and the mapping in (3.23) is not needed.[5]

The second integral b_{div} in (3.20) needs special considerations. It is zero for linear ansatz functions since then $\nabla \mathbf{N}_\pi$ is constant and hence its divergence is zero. However for $n>1$ b_{div} has to be evaluated. This yields

$$\int\limits_{\Omega_v} \mathrm{Div}[\nabla \mathbf{N}_\pi] \cdot \boldsymbol{u}_h \, \mathrm{d}\Omega = \int\limits_{\Omega_v} \left\{ \begin{matrix} N_{\pi x,xx} + N_{\pi x,yy} \\ N_{\pi y,xx} + N_{\pi y,yy} \end{matrix} \right\} \cdot \boldsymbol{u}_h \, \mathrm{d}\Omega = \mathbf{c}^T \sum_{j=1}^{n_m} [\mathbf{B}^j_{\pi\, div}]^T \, \mathbf{m}_j \tag{3.24}$$

where $n_m = \sum_{i=1}^{n}(i-1)$ is the number of the moments $n_m = 0\,,1\,,3\,,6\,,\ldots$ for $n = 1\,,2\,,3\,,4\,,\ldots$ that have to be introduced depending on the order n of the polynomial. The matrix $\mathbf{B}^j_{\pi\, div}$ is constant and distributes the moments to the correct location in the matrix equation.

Now the left and right hand sides of (3.19) are explicitly expressed in terms of the unknowns of the virtual element formulation introduced in Sect. 3.1.1. With the positive definite symmetric matrix

$$\mathbf{G}_\pi = \int\limits_{\Omega_v} \mathbf{B}_\pi(X,Y)^T \, \mathbf{B}_\pi(X,Y) \, \mathrm{d}\Omega \tag{3.25}$$

and a vector that contains the unknown nodal values and moments of the virtual element

$$\mathbf{b}_\pi(\mathbf{u}_i\,,\mathbf{m}_j) = \sum_{e=1}^{n_E} \int\limits_{\Gamma_e} \mathbf{B}_\pi^T \sum_{i=1}^{n+1} M_i(\xi) \begin{Bmatrix} u_{xi} N_{xe} \\ u_{xi} N_{ye} \\ u_{yi} N_{xe} \\ u_{yi} N_{ye} \end{Bmatrix} \mathrm{d}\Gamma + \sum_{j=1}^{n_m} [\mathbf{B}^j_{\pi\, div}]^T \, \mathbf{m}_j \tag{3.26}$$

the projection (3.19) provides an equation system for the unknowns

$$\mathbf{G}_\pi \, \hat{\mathbf{a}} = \mathbf{b}_\pi(\mathbf{u}_i\,,\mathbf{m}_j) \longrightarrow \hat{\mathbf{a}} = [\mathbf{G}_\pi]^{-1} \, \mathbf{b}_\pi(\mathbf{u}_i\,,\mathbf{m}_j)\,. \tag{3.27}$$

[5] It is of course possible to use any other numerical integration rule, like the classical Gauss integration that is mainly applied in finite element analysis. However due to the special structure of the integral in (3.22) such rules are not optimal.

This equation has to be solved at element level and yields the unknown parameters $\hat{\mathbf{a}}$ in terms of the nodal displacements $\mathbf{u}_i$ and the moments $\mathbf{m}_j$. With this result the displacement gradient can be expressed for a virtual element Ω_v as

$$\nabla \boldsymbol{u}_\pi = \mathbf{B}_\pi(X, Y)\,\hat{\mathbf{a}} = \mathbf{B}_\pi(X, Y)\,[\mathbf{G}_\pi]^{-1}\,\mathbf{b}_\pi(\mathbf{u}_i\,, \mathbf{m}_j) \tag{3.28}$$

and can be used in the weak form to compute strains and stresses, see Chap. 5.

For a more detailed notation we introduce $\mathbf{B}_{u\,\pi}^{(d,n)}(X, Y)$ for $\mathbf{B}_\pi(X, Y)$. Here d is the dimension of the problem (in this section we have $d = 2$), n denotes the ansatz order and the subscript u of $\mathbf{B}$ refers to its relation to the displacement field. Now the last equation can be written in a more general form for the two-dimensional case

$$\nabla \boldsymbol{u}_\pi = \mathbf{B}_{u\,\pi}^{(2,n)}(X, Y)\,\hat{\mathbf{a}} = \mathbf{B}_{u\,\pi}^{(2,n)}(X, Y)\,[\mathbf{G}_{u\,\pi}^{(n)}]^{-1}\,\mathbf{b}_{u\,\pi}^{(n)}(\mathbf{u}_i\,, \mathbf{m}_j)\,. \tag{3.29}$$

Generally it is possible to construct a projection matrix $\mathbb{P}_\nabla$ that combines matrices $\mathbf{G}_\pi$ and $\mathbf{b}_\pi$ in a form that the unknowns of the virtual element

$$\mathbf{u}_v = \{\mathbf{u}_1\,, \mathbf{u}_2\,, \dots\,, \mathbf{u}_{n_V}\,, \mathbf{m}_1\,, \mathbf{m}_2\,, \dots\,, \mathbf{m}_{n_m}\}$$

are explicitly extracted. This leads to the general form

$$\nabla \boldsymbol{u}_\pi = \mathbf{B}_{u\,\pi}^{(2,n)}(X, Y)\,[\mathbf{G}_{u\,\pi}^{(n)}]^{-1}\,\mathbf{b}_{u\,\pi}^{(n)}(\mathbf{u}_i\,, \mathbf{m}_j) = \mathbf{B}_{u\,\pi}^{(2,n)}(X, Y)\,\mathbb{P}_\nabla^{(2,n)}\,\mathbf{u}_v \tag{3.30}$$

which will be specified in detail in the next sections.

So far the approximation for the projection $\boldsymbol{u}_\pi$ is not completely determined. The constant parts of the projected displacement field are missing since only the gradient $\nabla \boldsymbol{u}_\pi$ was considered. In order to determine the parameters that are related to the constant part of the ansatz $\mathbf{N}_\pi$, see (3.3), an additional equation is needed. The idea is to equate the average of the ansatz $\boldsymbol{u}_h$ and the projection $\boldsymbol{u}_\pi$ over the virtual element Ω_v. For $n = 1$ this yields

$$\sum_{I=1}^{n_V} \boldsymbol{u}_\pi(\mathbf{X}_I) = \sum_{I=1}^{n_V} \boldsymbol{u}_h(\mathbf{X}_I) = \sum_{I=1}^{n_V} \mathbf{u}_I \tag{3.31}$$

where $\mathbf{u}_I$ are the nodal degrees of freedom at node $\mathbf{X}_I$. For $n \geq 2$ the average is computed as

$$\int\limits_{\Omega_v} \boldsymbol{u}_\pi \mathrm{d}\Omega = \int\limits_{\Omega_v} \boldsymbol{u}_h \mathrm{d}\Omega \tag{3.32}$$

which leads with

$$\boldsymbol{u}_\pi = \mathbf{H}_u^{(2,n)}(X,Y)\,\mathbf{a} = \begin{bmatrix} 1\;0\;X\;0\;Y\;0\;X^2\;\;0\;\;XY\ldots Y^n\;\;0 \\ 0\;1\;0\;X\;0\;Y\;\;0\;\;X^2\;\;0\;\;\ldots\;0\;\;Y^n \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ \cdots \\ \cdots \\ a_{n_c} \end{Bmatrix} \tag{3.33}$$

to

$$\int\limits_{\Omega_v} \mathbf{H}_u^{(2,n)}(X,Y)\,\mathrm{d}\Omega\,\mathbf{a} = \mathbf{m}_1\,. \tag{3.34}$$

This equation can be used to determine the parameters a_1 and a_2 in terms of the element unknowns $\mathbf{u}_v$. We note that the parameters a_3 to a_{n_c} are already known since they are equivalent to $\hat{\mathbf{a}}$ in (3.29). To this end, we can introduce a projection matrix $\mathbb{P}_u$ that connects the ansatz $\boldsymbol{u}_\pi$ with the element unknowns $\mathbf{u}_v$

$$\boldsymbol{u}_\pi = \mathbf{H}_u^{(2,n)}(X,Y)\,\mathbb{P}_u^{(2,n)}\,\mathbf{u}_v\,. \tag{3.35}$$

The projection matrix $\mathbb{P}_u$will be explicitly computed in the next sections for specific ansatz spaces.

3.1.3 Equivalent Projector

In general the virtual element discretization relies on the definitions (a0) to (a3), see Sect. 3.1.1. As seen in this section the projected gradient can be constructed depending on the degrees of freedom of any polygonal element Ω_v. However, there are some limitation of the space $\mathbf{V}_h$ since we do not know the ansatz $\boldsymbol{u}_h$ inside the element. This is not necessary as long as we have to deal with gradients as in the first term in the weak form (2.85). Once the computation of the volume loading term $\int_{\Omega_v} \bar{\boldsymbol{f}} \cdot \boldsymbol{u}_h\,\mathrm{d}\Omega$ in (2.82) has to be carried out or a mass matrix has to be computed we see limitations since $\boldsymbol{u}_h$ is not defined in Ω_v. For this reason Ahmad et al. (2013) have introduced a new projector $\boldsymbol{u}_\pi^0$ based on an L^2-projection. It is called equivalent projector and defined by

$$\int\limits_{\Omega_v} \mathbf{N}_\pi^T(\boldsymbol{u}_\pi^0 - \boldsymbol{u}_h)\,\mathrm{d}\Omega = 0 \tag{3.36}$$

Here the first term can be evaluated exactly since $\boldsymbol{u}_\pi^0$ is assumed to be a polynomial in X, Y. But the second term is not computable within the space $\mathbf{V}_h$. This is why a new relaxed space $\mathbf{W}_h$ is introduced and instead of (a3) in Sect. 3.1.1 the following properties are assumed

- (a3)' Div(Grad $\boldsymbol{u}_h$) is a polynomial of degree n on the polygon Ω_v and
- (a3)" The following relation holds:

$$\int\limits_{\Omega_v} X_i^\alpha\, \boldsymbol{u}_\pi^0\ \mathrm{d}\Omega = \int\limits_{\Omega_v} X_i^\alpha\, \boldsymbol{u}_h\ \mathrm{d}\Omega \tag{3.37}$$

with $|\alpha| = n - 1\,, n,\ X_1 \equiv X$ and $X_2 \equiv Y$.

It was shown in Ahmad et al. (2013) that the projection defined in $\mathbf{W}_h$ is equivalent for $n = 1$ and $n = 2$ to the projection using the space $\mathbf{V}_h$, thus $\boldsymbol{u}_\pi^0 = \boldsymbol{u}_\pi$, but yields a different result for $n > 2$. Condition (a3)" allows now to compute the loading term using the new projector as

$$\int_{\Omega_v} \bar{\boldsymbol{f}} \cdot \boldsymbol{u}_h\ \mathrm{d}\Omega = \int_{\Omega_v} \bar{\boldsymbol{f}} \cdot \boldsymbol{u}_\pi^0\ \mathrm{d}\Omega = \int_{\Omega_v} \bar{\boldsymbol{f}} \cdot \boldsymbol{u}_\pi\ \mathrm{d}\Omega \quad \text{for } n = 1, 2 \tag{3.38}$$

which is a polynomial in X, Y when $\bar{\boldsymbol{f}}$ is a polynomial in Ω_v. This looks like an approximation, but it was shown in Ahmad et al. (2013) that using the space $\mathbf{W}_h$ provides the same approximation order than using $\mathbf{V}_h$. For higher order ansatz spaces $n > 2$ one has to compute $\boldsymbol{u}_\pi^0$ explicitly, for details see Ahmad et al. (2013) or Beirão da Veiga et al. (2014).

3.1.4 Projection for a Linear Ansatz

In the case of a linear ansatz element nodes are placed only at the vertices of the polygonal elements, see Fig. 3.4.

Let us define a general linear ansatz function for $\boldsymbol{u}_\pi$

$$\mathbf{H}_u^{(2,1)}(X, Y) = \begin{bmatrix} 1 & 0 & X & 0 & Y & 0 \\ 0 & 1 & 0 & X & 0 & Y \end{bmatrix} \tag{3.39}$$

This yields a constant displacement gradient grad $\boldsymbol{u}_\pi = \nabla \boldsymbol{u}_\pi = const$. In Voigt notation we can write for (3.13) explicitly

$$\nabla \mathbf{u}_\pi = \begin{Bmatrix} u_{x,x} \\ u_{x,y} \\ u_{y,x} \\ u_{y,y} \end{Bmatrix} = \mathbf{B}_{u\,\pi}^{(2,1)}\, \hat{\mathbf{a}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \hat{a}_4 \end{Bmatrix}. \tag{3.40}$$

The ansatz for $\boldsymbol{u}_h$ is defined with (3.4) at the edge Γ_e, see Fig. 3.4. This ansatz is of the same order as the ansatz for $\mathbf{H}_u$ and hence a linear ansatz for the deformation along the element edge can be defined for a boundary segment k of the virtual element, defined by the local nodes (1)–(2) by, see right side of Fig. 3.4, using (3.5)

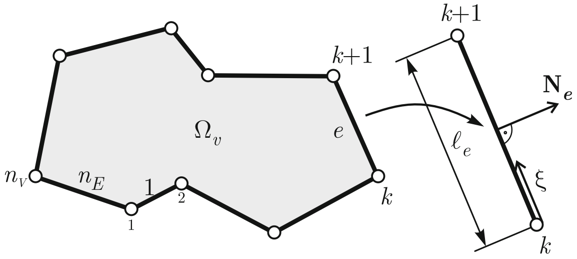


Fig. 3.4 Virtual element with n_V nodes and local boundary segment of the element

$$(\boldsymbol{u}_h)_e \; = (1-\xi_e)\,\mathbf{u}_1 + \xi_e\,\mathbf{u}_2 = M_{e\,1}\,\mathbf{u}_1 + M_{e\,2}\,\mathbf{u}_2 \quad \text{with } \xi_e = \frac{X_e}{l_e} \tag{3.41}$$

where, for example, $M_{k\,1}$ is the ansatz function along a segment k related to node (1), ξ_k is the local dimensionless coordinate and $\mathbf{u}_1$ is the nodal value at that node, see Fig. 3.4.

Based on (3.25) we compute with (3.40)

$$\mathbf{G}_\pi^{(1)} = \int\limits_{\Omega_v} [\mathbf{B}_{u\,\pi}^{(2,1)}]^T\,\mathbf{B}_{u\,\pi}^{(2,1)} \mathrm{d}\Omega = \Omega_v\,[\mathbf{B}_{u\,\pi}^{(2,1)}]^T\,\mathbf{B}_{u\,\pi}^{(2,1)} = \Omega_v\,\mathbf{I}\,. \tag{3.42}$$

The ansatz functions for $\nabla\boldsymbol{u}_\pi$ and $(\boldsymbol{u}_h)_k$ can now be inserted in (3.26). This yields together with the trapezoidal rule

$$\mathbf{b}_\pi = \sum_{e=1}^{n_E} [\mathbf{B}_{u\,\pi}^{(2,1)}]^T \int\limits_{\Gamma_e} \sum_{i=1}^{2} M(\xi) \begin{Bmatrix} u_{xi} N_x \\ u_{xi} N_y \\ u_{yi} N_x \\ u_{yi} N_y \end{Bmatrix} \mathrm{d}\Gamma = \sum_{e=1}^{n_E} \mathbf{B}_{u\,\pi}^{(2,1)} \begin{Bmatrix} (u_{xk} + u_{xk+1}) N_{xe} \\ (u_{xk} + u_{xk+1}) N_{ye} \\ (u_{yk} + u_{yk+1}) N_{xe} \\ (u_{yk} + u_{yk+1}) N_{ye} \end{Bmatrix} \frac{l_e}{2} \tag{3.43}$$

where index k denotes kth polygon vertex, for $1 \le k \le n_V$. Additionally, the nodal displacements $(u_{x\,n_V+1}, u_{y\,n_V+1})$ are assumed to be equal to the displacements of the first vertex, i.e., $u_{x\,n_V+1} = u_{x\,1}$. The parameters $\hat{\mathbf{a}}$ follow from (3.27) as

$$\hat{\mathbf{a}} = [\mathbf{G}_\pi^{(1)}]^{-1}\,\mathbf{b}_\pi = \frac{1}{2\Omega_v} \sum_{e=1}^{n_E} \begin{Bmatrix} (u_{xk} + u_{xk+1}) N_{xe} \\ (u_{yk} + u_{yk+1}) N_{xe} \\ (u_{xk} + u_{xk+1}) N_{ye} \\ (u_{yk} + u_{yk+1}) N_{ye} \end{Bmatrix} l_e \tag{3.44}$$

Finally, the gradient operator in (3.13) can be expressed in terms of the nodal unknowns u_{xk} and u_{yk}, leading with the normal vector $\boldsymbol{N}_e$ for a segment e

$$\boldsymbol{N}_e = \frac{1}{l_e} \begin{Bmatrix} N_x \\ N_y \end{Bmatrix}_e = \frac{1}{l_e} \begin{Bmatrix} -(Y_2 - Y_1) \\ X_2 - X_1 \end{Bmatrix}_e \tag{3.45}$$

to

$$\nabla \boldsymbol{u}_\pi = \begin{Bmatrix} u_{\pi x,x} \\ u_{\pi x,y} \\ u_{\pi y,x} \\ u_{\pi y,y} \end{Bmatrix} = \mathbf{B}_{u\,\pi}^{(2,1)}\,\hat{\mathbf{a}} = \frac{1}{2\Omega_v}\sum_{e=1}^{n_E} \begin{Bmatrix} (u_{xk}+u_{xk+1})(Y_1-Y_2)_e \\ (u_{xk}+u_{xk+1})(X_2-X_1)_e \\ (u_{yk}+u_{yk+1})(Y_1-Y_2)_e \\ (u_{yk}+u_{yk+1})(X_2-X_1)_e \end{Bmatrix} . \tag{3.46}$$

Thus, no equation system has to be solved in for a linear ansatz.

The projected gradient $\nabla \boldsymbol{u}_\pi$ in (3.46) can be directly expressed by the nodal unknowns of the virtual element. This is the reason why this discretization techniques are called *virtual*. In short notation we can write

$$\hat{\mathbf{a}} = \mathbb{P}_\nabla^{(2,1)}\,\mathbf{u}_v \tag{3.47}$$

where $\hat{\mathbf{a}}$ are the parameters of the virtual element ansatz, $\mathbb{P}_\nabla^{2,1}$ is a projection function which relates the vector $\mathbf{u}_v$, containing the nodal degrees of freedom of all vertices of the virtual element Ω_v, to the parameters $\hat{\mathbf{a}}$. The nodal displacements $\mathbf{u}_v$ of the virtual element can be cast in a vector

$$\mathbf{u}_v = \langle u_{x_1}\, u_{y_1}\, u_{x_2}\, u_{y_2}\, \ldots\, u_{x_{n_V}}\, u_{y_{n_V}} \rangle^T . \tag{3.48}$$

With some algebra the projection tensor can be specified in matrix form. By introducing the matrices

$$\mathbf{L}_x^T = \frac{1}{2\Omega_v}\begin{bmatrix} \mathbf{L}_{x1}^T \\ \mathbf{L}_{x2}^T \end{bmatrix} = \frac{1}{2\Omega_v}\begin{bmatrix} P_{x1} & 0 & P_{x2} & 0 & \ldots & P_{xn_V} & 0 \\ 0 & P_{x1} & 0 & P_{x2} & \ldots & 0 & P_{xn_V} \end{bmatrix} \tag{3.49}$$

and

$$\mathbf{L}_y^T = \frac{1}{2\Omega_v}\begin{bmatrix} \mathbf{L}_{y1}^T \\ \mathbf{L}_{y2}^T \end{bmatrix} = \frac{1}{2\Omega_v}\begin{bmatrix} P_{y1} & 0 & P_{y2} & 0 & \ldots & P_{yn_V} & 0 \\ 0 & P_{y1} & 0 & P_{y2} & \ldots & 0 & P_{yn_V} \end{bmatrix} \tag{3.50}$$

with the abbreviations $P_{xk} = (Y_{k-1} - Y_{k+1})$ and $P_{yk} = (X_{k+1} - X_{k-1})$[6] the parameters $\hat{a}_1$ to $\hat{a}_4$, see (3.46), can be expressed in terms of the nodal unknowns of the virtual element, see (3.48). Thus, the projection of the gradient can be simply written as

$$\nabla \boldsymbol{u}_\pi = \mathbf{B}_{u\,\pi}^{(2,1)}\,\hat{\mathbf{a}} = \mathbb{P}_\nabla^{(2,1)}\,\mathbf{u}_v \quad \text{with } \mathbb{P}_\nabla^{(2,1)} = \begin{bmatrix} \mathbf{L}_x^T \\ \mathbf{L}_y^T \end{bmatrix} . \tag{3.51}$$

where the projection tensor is constant and depends only on the coordinates of the vertices and the element area Ω_v which itself can be expressed by a sum over the vertices, see Appendix A.

[6] With the special cases $P_{x1} = (Y_{n_V} - Y_2)$, $P_{xn_V} = (Y_{n_V-1} - Y_1)$, $P_{y1} = (X_2 - X_{n_V})$ and $P_{yn_v} = (X_1 - X_{n_V-1})$.

One can use the result in (3.46) to construct an approximation of the displacement field in Ω_v. This can be achieved by using the linear ansatz (3.39) for displacement field

$$\boldsymbol{u}_\pi = \mathbf{H}_u^{(2,1)}(X,Y)\,\mathbf{a} = \begin{bmatrix} 1 & 0 & X & 0 & Y & 0 \\ 0 & 1 & 0 & X & 0 & Y \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ \dots \\ a_6 \end{Bmatrix} \tag{3.52}$$

which has the unknown parameters a_i. Computing the gradient of (3.52) yields the ansatz (3.40). Thus the parameters a_3 to a_6 are already known, see (3.46): $a_{i+2} = \hat{a}_i$ for $1 \le i \le 4$. Now the parameters a_1 and α_2, related to the constant parts in (3.52), have to be computed. This is performed in a different way by using the average displacement in the virtual element, see (3.31), which yields for every virtual element Ω_v

$$\sum_{k=1}^{n_V} \boldsymbol{u}_\pi(\mathbf{X}_k) = \sum_{k=1}^{n_V} \boldsymbol{u}_h(\mathbf{X}_k)\,, \tag{3.53}$$

where $\mathbf{X}_k$ are the coordinates of a vertex k. Equation (3.53) is valid for both components of the displacement field and hence yields two equations that determine the constants a_1 and a_2. With the already computed unknowns $\hat{a}_1$ to $\hat{a}_4$, see (3.46), we obtain after some algebra

$$\begin{aligned} a_1 &= \sum_{k=1}^{n_V} [\, u_{x\,k} - \hat{a}_1\, X_k - \hat{a}_3\, Y_k \,] \,/\, n_V \\ a_2 &= \sum_{k=1}^{n_V} [\, u_{y\,k} - \hat{a}_2\, X_k - \hat{a}_4\, Y_k \,] \,/\, n_V \end{aligned} \tag{3.54}$$

Now the approximate ansatz function $\boldsymbol{u}_\pi$ of the virtual element is completely defined in Ω_v.[7]

Remark 3.1 Since all parameters a_i $(i = 1 \dots 6)$ are determined by (3.51) and (3.54) in terms of the nodal values $\mathbf{u}_k$ (with $a_3 = \hat{a}_1$ to $a_6 = \hat{a}_4$) it is not necessary to use the parameters a_i within the discretization. A projection $\mathbb{P}_u^{(2,1)}$ provides, analogous to (3.47), a form of (3.52) that depends only on the nodal degrees of the vertices of the virtual element Ω_v. By defining

$$X^S = \sum_{k=1}^{n_V} X_k \quad \text{and} \quad Y^S = \sum_{k=1}^{n_V} Y_k \quad \text{and} \quad \mathbf{I}^T = \begin{bmatrix} 1 & 0 & 1 & 0 & \dots & 1 & 0 \\ 0 & 1 & 0 & 1 & \dots & 0 & 1 \end{bmatrix} \tag{3.55}$$

[7] Note again that the constant gradient $\nabla \boldsymbol{u}_\pi$ can be computed directly using (3.46) and (3.51). Thus for the computation of the strain energy it is not necessary to evaluate equation (3.54).

the parameters a_1 and a_2 follow according to (3.54)

$$\begin{Bmatrix} a_1 \\ a_2 \end{Bmatrix} = \frac{1}{n_V} \left[\mathbf{I}^T - X^S \, \mathbf{L}_x^T - Y^S \, \mathbf{L}_y^T \right] \mathbf{u}_v \tag{3.56}$$

which finally leads to the explicit form of the projection matrix

$$\mathbb{P}_u^{(2,1)} = \begin{bmatrix} (\mathbf{I}^T - X^S \, \mathbf{L}_x^T - Y^S \, \mathbf{L}_y^T)/n_V \\ \mathbf{L}_x^T \\ \mathbf{L}_y^T \end{bmatrix} . \tag{3.57}$$

With this short notation we have an explicit form that describes an approximation of the displacement field $\boldsymbol{u}_\pi$ within Ω_v in terms of the nodal values using (3.52)

$$\boldsymbol{u}_\pi = \mathbf{H}_u^{(2,1)}(X, Y) \, \mathbb{P}_u^{(2,1)} \, \mathbf{u}_v \tag{3.58}$$

which is a linear function in X and Y. ■

3.1.5 Computation of the Projection Using Symbolic Software

In case of a linear ansatz function the polynomial projection function for 1st order VEM is given by

$$\mathbf{N}_\pi = \langle 1 \; X \; Y \rangle \; .$$

In this case the gradients that have to evaluated using the ansatz functions are constant

$$\nabla \mathbf{N}_\pi = \text{const. and } \; \nabla \mathbf{u}_\pi = \text{const.}$$

which simplifies equation (3.20) since the integral b_{div} is zero and (3.19) can be written as

$$\int\limits_{\Omega_v} \nabla \boldsymbol{u}_h \; \mathrm{d}\Omega - \nabla \boldsymbol{u}_\pi \; \Omega_v = 0 \, . \tag{3.59}$$

Again, the Gauss theorem provides a simple way to compute the parameters $\hat{\mathbf{a}}$ which in this case is equivalent to compute the gradient $\nabla \mathbf{u}_\pi$

$$\hat{\mathbf{a}} = \nabla \mathbf{u}_\pi = \frac{1}{\Omega_v} \int\limits_{\Omega_v} \nabla \, \boldsymbol{u}_h \; \mathrm{d}\Omega = \frac{1}{\Omega_v} \int\limits_{\Gamma_v} \boldsymbol{u}_h \otimes \boldsymbol{N} \; \; \mathrm{d}\Gamma \tag{3.60}$$

where $\boldsymbol{N}$ is the outward normal at the boundary Γ_v of the domain Ω_v, see Fig. 3.2. Using now the Lobattto rule to integrate over the boundary Γ_v in (3.60), the gradient $\nabla \mathbf{u}_\pi$ can be written in tensor notation

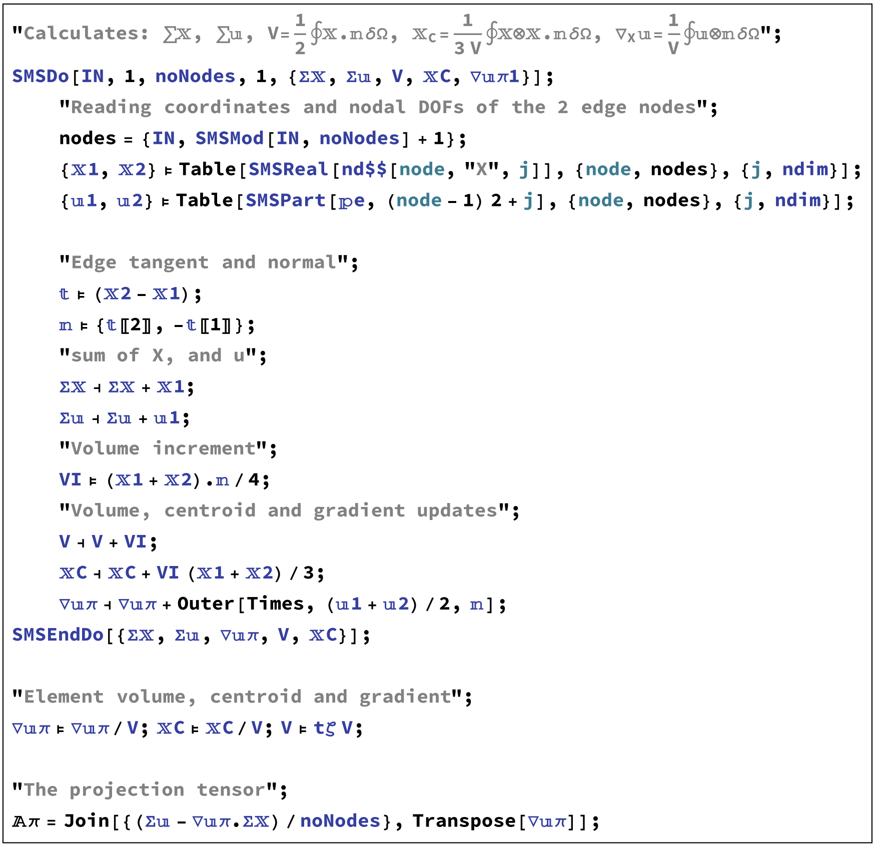
$$\nabla \mathbf{u}_\pi = \frac{1}{2\Omega_v} \sum_{e=1}^{n_E} \begin{Bmatrix} u_{xk} + u_{xk+1} \\ u_{yk} + u_{yk+1} \end{Bmatrix} \begin{Bmatrix} (Y_1 - Y_2)_e \\ (X_2 - X_1)_e \end{Bmatrix}^T \tag{3.61}$$

which is convenient when a symbolic algebra tool like *AceGen* is employed, see Korelc and Wriggers (2016).

Since $\hat{\mathbf{a}}$ is equivalent to the gradient $\nabla \mathbf{u}_\pi$ the constants a_1 and a_2 in (3.54) follow from

$$\begin{Bmatrix} a_1 \\ a_2 \end{Bmatrix} = \sum_{k=1}^{n_V} \left[\begin{Bmatrix} u_{xk} \\ u_{yk} \end{Bmatrix} - \nabla \mathbf{u}_\pi \begin{Bmatrix} X_k \\ Y_k \end{Bmatrix} \right] = \sum_{k=1}^{n_V} [\mathbf{u}_k - \nabla \mathbf{u}_\pi \, \mathbf{X}_k] \tag{3.62}$$

Within the software tool *AceGen* equations (3.61) and (3.62) are the basis for the generation of the code associated with the projection which links the nodal displacements to the parameters a_i. A sample code for the computation of the parameters a_i, $(i = 1, 6)$ is provided in Fig. 3.5. Here a sum over all nodes of a



```
"Calculates: ∑𝕏, ∑𝕦, V=1/2∮𝕏.𝕟δΩ, 𝕏C=1/(3 V)∮𝕏⊗𝕏.𝕟δΩ, ∇X𝕦=1/V∮𝕦⊗𝕟δΩ";
SMSDo[IN, 1, noNodes, 1, {Σ𝕏, Σ𝕦, V, 𝕏C, ∇𝕦π1}];
    "Reading coordinates and nodal DOFs of the 2 edge nodes";
    nodes = {IN, SMSMod[IN, noNodes] + 1};
    {𝕏1, 𝕏2} ⊨ Table[SMSReal[nd$$[node, "X", j]], {node, nodes}, {j, ndim}];
    {𝕦1, 𝕦2} ⊨ Table[SMSPart[𝕡e, (node - 1) 2 + j], {node, nodes}, {j, ndim}];

    "Edge tangent and normal";
    𝕥 ⊨ (𝕏2 - 𝕏1);
    𝕟 ⊨ {𝕥〚2〛, -𝕥〚1〛};
    "sum of X, and u";
    Σ𝕏 ⊣ Σ𝕏 + 𝕏1;
    Σ𝕦 ⊣ Σ𝕦 + 𝕦1;
    "Volume increment";
    VI ⊨ (𝕏1 + 𝕏2).𝕟 / 4;
    "Volume, centroid and gradient updates";
    V ⊣ V + VI;
    𝕏C ⊣ 𝕏C + VI (𝕏1 + 𝕏2) / 3;
    ∇𝕦π ⊣ ∇𝕦π + Outer[Times, (𝕦1 + 𝕦2) / 2, 𝕟];
SMSEndDo[{Σ𝕏, Σ𝕦, ∇𝕦π, V, 𝕏C}];

"Element volume, centroid and gradient";
∇𝕦π ⊨ ∇𝕦π / V; 𝕏C ⊨ 𝕏C / V; V ⊨ tξ V;

"The projection tensor";
𝔸π = Join[{(Σ𝕦 - ∇𝕦π.Σ𝕏) / noNodes}, Transpose[∇𝕦π]];
```

Fig. 3.5 Sample code for the determination of a_i as a function of $\mathbf{u}_v$

virtual element provides the gradient, as in (3.61) as $\nabla \mathbf{u}_\pi = (\mathbf{u}_1 + \mathbf{u}_2)/2 \otimes \mathbf{n} =$ `Outer[Times,` $(\mathbf{u}_1 + \mathbf{u}_2)/2$, $\mathbf{n}$] where $\mathbf{u}_1 \equiv \mathbf{u}_k$ and $\mathbf{u}_2 \equiv \mathbf{u}_{k+1}$ and $\mathbf{n}$ is the outward normal. Once the sum is performed over all nodes the projection tensor $\mathbf{A}_\pi$, equivalent to the matrix $\mathbf{a}$, follows from (3.61) and (3.62) by joining all parameters a_i, see last line in the code in Fig. 3.5.

3.1.6 Projection for a Quadratic Ansatz

A complete ansatz using the quadratic polynomial $\mathbf{N}_\pi^{(2)} = (1\,, X\,, Y\,, X^2\,, XY\,, Y^2)$ is basis for the second order virtual element. This yields an approximation of the gradient $\nabla \boldsymbol{u}_\pi$ containing linear functions in X and Y

$$\nabla \boldsymbol{u}_\pi = \begin{Bmatrix} u_{x,x} \\ u_{x,y} \\ u_{y,x} \\ u_{y,y} \end{Bmatrix} = \mathbf{B}_{u\,\pi}^{(2,2)}(X, Y)\,\hat{\mathbf{a}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 2X & 0 & Y & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & X & 0 & 2Y & 0 \\ 0 & 1 & 0 & 0 & 0 & 2X & 0 & Y & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & X & 0 & 2Y \end{bmatrix} \begin{Bmatrix} \hat{a}_1 \\ \hat{a}_1 \\ \dots \\ \hat{a}_{10} \end{Bmatrix} . \tag{3.63}$$

The ansatz $\boldsymbol{u}_h$ at the edge Γ_e, see Fig. 3.6 is defined by (3.4).

This ansatz is of the same order as the ansatz for $\mathbf{N}_\pi^{(2)}$ and thus we have to select the shape functions in (3.6). The element edge is assumed to be straight. Thus a boundary segment k of the virtual element is defined by the local nodes $(2k-1)$ – $(2k)$ – $(2k+1)$

$$\boldsymbol{u}_h = M_1(\xi)\,\mathbf{u}_{2k-1} + M_3(\xi)\,\mathbf{u}_{2k} + M_2(\xi)\,\mathbf{u}_{2k+1} \tag{3.64}$$

$$\text{with } \xi = \frac{1}{l_e^2}(\boldsymbol{X} - \mathbf{X}_{2k-1}) \cdot (\mathbf{X}_{2k+1} - \mathbf{X}_{2k-1})$$

where $M_i(\xi)$, $(i = 1, 2, 3)$, are the ansatz functions at segment k related to local nodes, ξ is the local dimensionless coordinate and $\mathbf{u}_i$ are the nodal value at the nodes, see Fig. 3.6. The outward normal vector related to the edge Γ_e is denoted by $\boldsymbol{N}_e$.

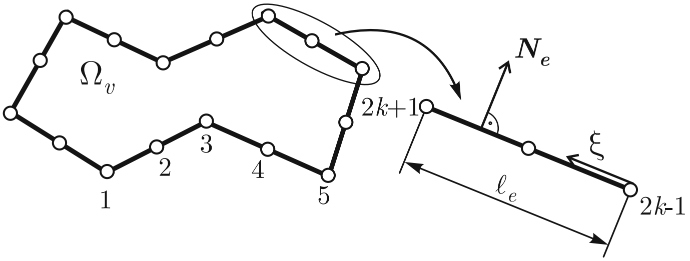


Fig. 3.6 Description of VEM elements for linear and quadratic ansatz spaces

The matrices $\mathbf{G}_\pi^{(2)}$ and $\mathbf{b}_\pi^{(2)}$ can be computed for the quadratic ansatz based on (3.63). The matrix $\mathbf{G}_\pi^{(2)}$ for quadratic ansatz functions follows by inserting (3.63) into (3.25)

$$\mathbf{G}_\pi^{(2)} = \int\limits_{\Omega_v} [\mathbf{B}_{u\,\pi}^{(2,2)}(X,Y)]^T \mathbf{B}_{u\,\pi}^{(2,2)}(X,Y)\mathrm{d}\,\Omega\,. \tag{3.65}$$

It has the size of (10×10). Since $\mathbf{B}_{u\,\pi}^{(2,2)}(X,Y)$ is a linear function of X and Y the integral in (3.65) contains the terms 1, X, Y, X^2, XY and Y^2. This integral can be evaluated exactly for arbitrary polygonal virtual elements by using the integration formulae in Appendix A.

The vector $\mathbf{b}_\pi^{(2)}$ follows with (3.64) and (3.7) from (3.26) as

$$\mathbf{b}_\pi^{(2)}(\mathbf{u}_i\,,\mathbf{m}_1) = \sum_{e=1}^{n_E} \int\limits_{\Gamma_e} [\mathbf{B}_{u\,\pi}^{(2,2)}(X,Y)]^T \sum_{i=1}^{3} M_i(\xi) \begin{Bmatrix} u_{xi}\,N_{xe} \\ u_{xi}\,N_{ye} \\ u_{yi}\,N_{xe} \\ u_{yi}\,N_{ye} \end{Bmatrix} \mathrm{d}\Gamma + [\mathbf{B}_{\pi\,div}^{(1)}]^T\,\mathbf{m}_1\,. \tag{3.66}$$

The first term in this integral contains a linear function of X and Y in $\mathbf{B}_{u\,\pi}^{(2,2)}(X,Y)$. Furthermore the ansatz functions M_i are quadratic functions of ξ and since the edge is straight, also of X and Y. Such integrals containing up to cubic polynomial orders can be evaluated exactly by using 2 point Gauss quadrature. However in this case it is more advantageous to use the Gauss-Lobatto rule since this needs only evaluations at the nodel points. By using additionally the property of the shape functions $M_i M_j = \delta_{ij}$ and setting locally $(2k-1) \to 1$, $(2k) \to 3$ and $(2k+1) \to 2$ the explicit form of the first integral is given by

$$\sum_{e=1}^{n_E} l_e \sum_{i=1}^{3} w_i [\mathbf{B}_{u\,\pi}^{(2,2)}(X_i,Y_i)]^T \begin{Bmatrix} u_{xi}\,N_{xe} \\ u_{xi}\,N_{ye} \\ u_{yi}\,N_{xe} \\ u_{yi}\,N_{ye} \end{Bmatrix} \tag{3.67}$$

with the weighting factors of the Gauss-Lobatto rule: $w_1 = w_2 = \frac{1}{6}$ and $w_3 = \frac{2}{3}$.

The second term in (3.66) can be obtained from (3.24). We note that the divergence of $\nabla\mathbf{N}_\pi$ is constant. This leads to

$$\mathbf{B}_{\pi\,div}^{(1)} = 2\begin{bmatrix} 0 & 0 & 0 & 0 & \Omega_v & 0 & 0 & 0 & \Omega_v & 0 \\ 0 & 0 & 0 & 0 & 0 & \Omega_v & 0 & 0 & 0 & \Omega_v \end{bmatrix} \tag{3.68}$$

and with $\mathbf{m}_1 = (m_{x1}, m_{y1})$ the evaluations of the integrals in equation (3.66) yield

$$\mathbf{b}_\pi^{(2)}(\mathbf{u}_i\,,\mathbf{m}_1) = \sum_{e=1}^{n_E} l_e \sum_{i=1}^{3} w_i [\mathbf{B}_{u\,\pi}^{(2,2)}(X_i,Y_i)]^T \begin{Bmatrix} u_{xi}\,N_{xe} \\ u_{xi}\,N_{ye} \\ u_{yi}\,N_{xe} \\ u_{yi}\,N_{ye} \end{Bmatrix} + [\mathbf{B}_{\pi\,div}^{(1)}]^T \begin{Bmatrix} m_{x1} \\ m_{y1} \end{Bmatrix}\,. \tag{3.69}$$

With the explicit forms of $\mathbf{G}_\pi^2$ in (3.65) and $\mathbf{b}_\pi^{(2)}$ in (3.69) the equations system (3.27) can be solved for the unknown parameters $\hat{\mathbf{a}}$ leading to

$$\nabla \boldsymbol{u}_\pi = \mathbf{B}_{u\,\pi}^{(2,2)}(X, Y)\,\hat{\mathbf{a}} = \mathbf{B}_{u\,\pi}^{(2,2)}(X, Y)\,[\mathbf{G}_\pi^{(2)}]^{-1}\,\mathbf{b}_\pi^{(2)}(\mathbf{u}_i\,,\mathbf{m}_1) \tag{3.70}$$

which yields a displacement gradient that is a linear function in X and Y depending on the nodal values $\mathbf{u}_i$ and the moment $\mathbf{m}_1$.

As for the linear ansatz it is possible to express (3.70) by an explicit function of the degrees of freedom of the virtual element Ω_v. The vector contains the displacements at the vertices and the moments

$$\mathbf{u}_v = \langle u_{x_1}\; u_{y_1}\; u_{x_2}\; u_{y_2}\; \ldots u_{x_{2n_V}}\; u_{y_{2n_V}}\; m_{x1}\, m_{y1}\rangle^T = \langle \mathbf{u}_1\; \mathbf{u}_2\; \ldots\; \mathbf{u}_{2n_V}\; \mathbf{m}_1\rangle^T. \tag{3.71}$$

With the abbreviation $\mathbf{B}_{u\,\pi}^{(2,2)}(X_i, Y_i) = \mathbf{B}_i$ and the introduction

$$\mathbf{N}_e = \begin{bmatrix} N_{xe} & 0 \\ N_{ye} & 0 \\ 0 & N_{xe} \\ 0 & N_{ye} \end{bmatrix} = \frac{1}{l_e}\begin{bmatrix} (Y_1 - Y_2)_e & 0 \\ (X_2 - X_1)_e & 0 \\ 0 & (Y_1 - Y_2)_e \\ 0 & (X_2 - X_1)_e \end{bmatrix} = \frac{1}{l_e}\bar{\mathbf{N}}_e \tag{3.72}$$

Equation (3.69) can be written as

$$\begin{aligned}\mathbf{b}_\pi^{(2)}(\mathbf{u}_i\,,\mathbf{m}_1) = &\sum_{e=1}^{n_E} l_e\left[\frac{1}{6}\mathbf{B}_{2k-1}^T\mathbf{N}_e\mathbf{u}_{2k-1} + \frac{2}{3}\mathbf{B}_{2k}^T\mathbf{N}_e\mathbf{u}_{2k} + \frac{1}{6}\mathbf{B}_{2k+1}^T\mathbf{N}_e\mathbf{u}_{2k+1}\right] \\ &+ [\mathbf{B}_{\pi\,div}^{(1)}]^T\mathbf{m}_1\end{aligned} \tag{3.73}$$

where the dependence on the degrees of freedom is more obvious than in (3.70). In order to derive a form like in (3.51) one has to assemble all contributions related to a node within the virtual element Ω_v. This yields a matrix of size $(10 \times 4n_V + 2)$

$$\mathbf{B}_v^{(2)} = \left[\mathbf{P}_1^T\;\mathbf{P}_2^T\;\ldots\;\mathbf{P}_{2k-1}^T\;\mathbf{P}_{2k}^T\;\mathbf{P}_{2k+1}^T\;\ldots\mathbf{P}_{2n_V}^T\;\mathbf{B}_{\pi\,div}^{1\,T}\right] \tag{3.74}$$

with submatrices $\mathbf{P}_i^T$ of size (10×2)

$$\begin{aligned}\mathbf{P}_{2k-1}^T &= \frac{1}{6}\mathbf{B}_{2k-1}^T(l_e\mathbf{N}_e + l_{e-1}\mathbf{N}_{e-1}) = \frac{1}{6}\mathbf{B}_{2k-1}^T(\bar{\mathbf{N}}_e + \bar{\mathbf{N}}_{e-1}) \\ \mathbf{P}_{2k}^T &= \frac{2}{3}\mathbf{B}_{2k}^T l_e\mathbf{N}_e = \frac{2}{3}\mathbf{B}_{2k}^T\bar{\mathbf{N}}_e \\ \mathbf{P}_{2k+1}^T &= \frac{1}{6}\mathbf{B}_{2k+1}^T(l_e\mathbf{N}_e + l_{e+1}\mathbf{N}_{e+1}) = \frac{1}{6}\mathbf{B}_{2k+1}^T(\bar{\mathbf{N}}_e + \bar{\mathbf{N}}_{e+1})\end{aligned}$$

and the special case $\mathbf{P}_1^T = \frac{1}{6}\mathbf{B}_1^T (l_1\mathbf{N}_1 + l_{n_E}\mathbf{N}_{n_E}) = \frac{1}{6}\mathbf{B}_1^T (\bar{\mathbf{N}}_1 + \bar{\mathbf{N}}_{n_E})$. In analogy to the abbreviations used in (3.49) and (3.50) the sums of the relations above can be further simplified.

With (3.74) and (3.71) it is now possible to write $\mathbf{b}_\pi^{(2)}(\mathbf{u}_i\,,\mathbf{m}_1) = \mathbf{B}_v^{(2)}\,\mathbf{u}_v$ and equation (3.70) as

$$\nabla \boldsymbol{u}_\pi = \mathbf{B}_{u\,\pi}^{(2,2)}(X,Y)\,[\mathbf{G}_\pi^{(2)}]^{-1}\,\mathbf{B}_v^{(2)}\,\mathbf{u}_v = \mathbf{B}_{u\,\pi}^{(2,2)}(X,Y)\,\mathbb{P}_\nabla^{(2,2)}\,\mathbf{u}_v \tag{3.75}$$

where $\mathbb{P}_\nabla^{(2,2)} = [\mathbf{G}_\pi^{(2)}]^{-1}\,\mathbf{B}_v^{(2)}$. Now the projected gradient $\nabla\boldsymbol{u}_\pi$ depends directly on the unknown $\mathbf{u}_v$ related to the virtual element Ω_v.

Remark 3.2 In case that a virtual element is formulated for linear elasticity and small strains it is sufficient to directly approximate the symmetric strain tensor $\hat{\boldsymbol{\varepsilon}}$ in (2.46) with (3.17). Here we can use the result from (3.75) since the first two terms $\hat{\boldsymbol{\varepsilon}}$ are equivalent with $\nabla\boldsymbol{u}_\pi$. The shear strain part is a sum from the third and fourth term in $\nabla\boldsymbol{u}_\pi$. Thus only a new matrix $\mathbf{B}_\pi^\varepsilon$ has to be defined. This yields the ansatz for the strains

$$\boldsymbol{\varepsilon}_\pi = \begin{Bmatrix} u_{x,x} \\ u_{y,y} \\ u_{x,y} + u_{y,x} \end{Bmatrix} = \mathbf{B}_\pi^\varepsilon(X,Y)\,\hat{\mathbf{a}} = \begin{bmatrix} 1\,0\,0\,0\,2X & 0 & Y & 0 & 0 & 0 \\ 0\,0\,0\,1 \;\; 0 & 0 & 0 & X & 0 & 2Y \\ 0\,1\,1\,0 \;\; 0 & 2X & X & Y & 2Y & 0 \end{bmatrix} \begin{Bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \cdots \\ \hat{a}_{10} \end{Bmatrix}. \tag{3.76}$$

which again is linear in X and Y. Thus all equations – from (3.64) to (3.75) – can be utilized, by exchanging $\mathbf{B}_{u\,\pi}^{(2,2)}(X,Y)$ with $\mathbf{B}_\pi^\varepsilon(X,Y)$, to compute the relation

$$\boldsymbol{\varepsilon}_\pi = \mathbf{B}_\pi^\varepsilon(X,Y)\,\hat{\mathbf{a}} = \mathbf{B}_\pi^\varepsilon(X,Y)\,[\mathbf{G}_\pi^{(2)}]^{-1}\,\mathbf{B}_v^{(2)}\,\mathbf{u}_v = \mathbf{B}_\pi^\varepsilon(X,Y)\,\mathbb{P}_u^{(2,2)}\,\mathbf{u}_v \tag{3.77}$$

where the matrix $\mathbf{G}_\pi^{(2)}$ has again size (10 × 10) as in (3.65). ■

Remark 3.3 Following the arguments leading to (3.62) an approximate function for the displacement field in Ω_v can be computed for the quadratic ansatz. We introduce a complete polynomial of second order as

$$\boldsymbol{u}_\pi = \mathbf{H}_u^{(2,2)}(X,Y)\,\mathbf{a} = \begin{bmatrix} 1 & 0 & X & 0 & Y & 0 & X^2 & 0 & XY & 0 & Y^2 & 0 \\ 0 & 1 & 0 & X & 0 & Y & 0 & X^2 & 0 & XY & 0 & Y^2 \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ \dots \\ \dots \\ a_{12} \end{Bmatrix}. \tag{3.78}$$

The following equivalence holds for ($n \geq 2$), see Beirão da Veiga et al. (2013b),

$$\int\limits_{\Omega_v} \boldsymbol{u}_\pi \;\mathrm{d}\Omega = \int\limits_{\Omega_v} \boldsymbol{u}_h \;\mathrm{d}\Omega\,. \tag{3.79}$$

It can be reformulated with (3.7) and (3.78) as

$$\frac{1}{\Omega_v}\int\limits_{\Omega_v}\mathbf{H}_u^{(2,2)}(X,Y)\,\mathbf{a}\,\mathrm{d}\Omega=\mathbf{m}_1\,. \tag{3.80}$$

Here the right hand side is the moment which is a degree of freedom in the vector $\mathbf{u}_v$, see (3.71). From the 12 unknowns a_i in (3.80) 10 unknowns were already computed in (3.70). An obvious split of (3.78) into the known parameters $\hat{\mathbf{a}}$, see (3.75), and the two unknown parameters $\bar{\mathbf{a}}$ is

$$\boldsymbol{u}_\pi=\mathbf{H}_u^{(2,2)}(X,Y)\,\mathbf{a}=\mathbf{1}\,\bar{\mathbf{a}}+\hat{\mathbf{H}}_u^{(2,2)}(X,Y)\,\hat{\mathbf{a}}\quad\text{with }\bar{\mathbf{a}}=\begin{Bmatrix}a_1\\a_2\end{Bmatrix} \tag{3.81}$$

where $\mathbf{1}$ is a (2×2) unit matrix. The parameters a_1 and a_2 follow with (3.75) directly from

$$\bar{\mathbf{a}}=\mathbf{m}_1-\frac{1}{\Omega_v}\int\limits_{\Omega_v}\hat{\mathbf{H}}_u^{(2,2)}(X,Y)\,\mathrm{d}\Omega\,\hat{\mathbf{a}}=\mathbf{m}_1-\frac{1}{\Omega_v}\int\limits_{\Omega_v}\hat{\mathbf{H}}_u^{(2,2)}(X,Y)\,\mathrm{d}\Omega\,[\mathbf{G}_\pi^{(2)}]^{-1}\,\mathbf{B}_v^{(2)}\,\mathbf{u}_v\,. \tag{3.82}$$

Since $\hat{\mathbf{H}}_u^{(2,2)}(X,Y)$ only contains polynomials of order 2 the integral in (3.82) is exactly computable for any shape of Ω_v, see Appendix A. By introducing a matrix $\mathbf{I}_m=[\mathbf{0}\,,\mathbf{0}\,,\ldots\,\mathbf{0}\,,\mathbf{1}]$ with the length $(2\times n_v+1)$, the moment $\mathbf{m}_1$ can be expressed by $\mathbf{m}_1=\mathbf{I}_m\,\mathbf{u}_v$. Thus (3.82) can be rewritten with (3.75) as

$$\bar{\mathbf{a}}=\left[\mathbf{I}_m-\frac{1}{\Omega_v}\int\limits_{\Omega_v}\hat{\mathbf{H}}_u^{(2,2)}(X,Y)\,\mathrm{d}\Omega\,\mathbb{P}_\nabla^{(2,2)}\right]\mathbf{u}_v=\bar{\mathbb{Q}}_u^{(2)}\,\mathbf{u}_v\,. \tag{3.83}$$

Now the projected displacement field in Ω_v follows from (3.81)

$$\boldsymbol{u}_\pi=\left[\bar{\mathbb{Q}}_u^{(2)}+\hat{\mathbf{H}}_u^{(2,2)}(X,Y)\,\mathbb{P}_\nabla^{(2,2)}\right]\mathbf{u}_v=\mathbb{P}_u^{(2,2)}(X,Y)\,\mathbf{u}_v \tag{3.84}$$

which is a quadratic polynomial in X and Y. ■

3.1.7 Serendipity Virtual Element for a Quadratic Ansatz

Serendipity elements are well known in finite element methods. They have the advantage that no interior nodes appear for quadratic ansatz functions. Within virtual element ansatz function using a quadratic ansatz, like in the last section, not only nodal displacements but also moments $\mathbf{m}_1$ appear, see (3.71). Analogous to serendipity

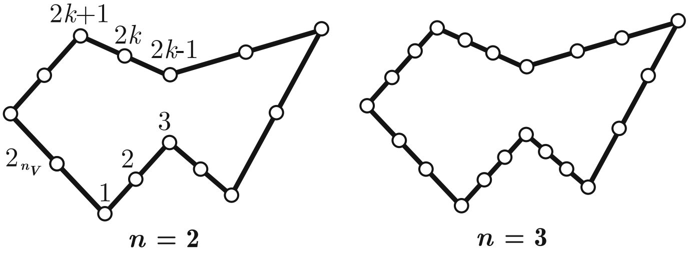


Fig. 3.7 Quadratic and cubic serendipity virtual elements

formulations within finite element analysis, it would be advantageous to condense these moments out such that the virtual element ansatz is based purely on the nodal displacement. Following ideas in Beirão da Veiga et al. (2017) and De Bellis et al. (2019) this can be achieved, see Fig. 3.7 for quadratic and cubic virtual elements.

Necessary amendments to the ansatz for a quadratic virtual element with arbitrary number of nodes, see (3.75), are discussed which yield a virtual element without internal degrees of freedom.

The independent vector variable, introduced as the moment $\mathbf{m}_1$, see (3.71), is replaced by an explicitly defined function, only depending on the nodal displacement vector $\bar{\mathbf{u}}_v$

$$\bar{\mathbf{u}}_v = \langle u_{x_1}\, u_{y_1}\, u_{x_2}\, u_{y_2}\, \ldots\, u_{x_{2n_V}}\, u_{y_{2n_V}} \rangle^T = \langle \mathbf{u}_1\, \mathbf{u}_2\, \ldots\, \mathbf{u}_{2n_V} \rangle^T . \tag{3.85}$$

This leads to

$$\mathbf{m}_1(\bar{\mathbf{u}}_v) = \int_{\Omega^e} \mathcal{F}(\bar{\mathbf{u}}_v)\, \mathrm{d}\Omega , \tag{3.86}$$

where $\mathcal{F}(\bar{\mathbf{u}}_v)$ is assumed to be a second order polynomial, i.e. $\mathcal{F}(\bar{\mathbf{u}}_v)$ can be expressed with (3.78) as

$$\mathcal{F}(\bar{\mathbf{u}}_v) = \mathbf{H}_u^{(2,2)}(X, Y)\, \mathbf{c} \tag{3.87}$$

with the unknowns $\mathbf{c} = \{c_1\,, c_2\,, \ldots, c_{12}\}^T$. In doing so, the moment $\mathbf{m}_1$ depends on the remaining nodal degrees of freedom $\bar{\mathbf{u}}_v$ and the resulting virtual element is characterized by a reduced number of degrees of freedom.

Now the operator $\mathcal{F}(\bar{\mathbf{u}}_v)$ has to be defined explicitly which is equivalent to determine $\mathbf{c}$. Following the same procedure as was used for the computation of the projection for the gradient we introduce a projection including all nodes of the element

$$\mathbf{u}_p^T \left[\mathcal{F}(\bar{\mathbf{u}}_v) - \bar{\mathbf{u}}_v \right] = 0 \tag{3.88}$$

which is a projection in the least square sense and yields an equation system for the unknown $\mathbf{c}$. Here $\mathbf{u}_p$ is a quadratic function with $\mathbf{u}_p = \mathbf{H}_u^{(2,2)}(X, Y)\boldsymbol{\alpha}$. By summing up all nodal contributions and with the abbreviation $\mathbf{X}_k = (X_k, Y_k)$ Eq. (3.88) yields

$$\sum_{k=1}^{n_V} [\mathbf{H}_u^{(2,2)}(\mathbf{X}_k)]^T \, \left[\mathbf{H}_u^{(2,2)}(\mathbf{X}_k)\, \mathbf{c} - \bar{\mathbf{u}}_v\right] = 0\,. \tag{3.89}$$

This equation system can be written in short notation as

$$\mathbf{G}^{(2)}\, \mathbf{c} = \mathbf{b}^{(2)}\,(\bar{\mathbf{u}}_v), \tag{3.90}$$

where $\mathbf{G}^{(2)}$ is a (12×12) matrix that includes the sum over all outer products appearing in the sum of equation system (3.89). The matrix $\mathbf{G}^{(2)}$ is invertible and $\mathbf{b}^{(2)}\,(\bar{\mathbf{u}}_v)$ is a vector which depends on the unknowns $\bar{\mathbf{u}}_v$. In particular, the matrix $\mathbf{G}^{(2)}$ and the vector $\mathbf{b}^{(2)}$ have the block structure

$$\mathbf{G}^{(2)} = \begin{bmatrix} \mathbf{H}_u^{(2,2)}(\mathbf{X}_1) \\ \mathbf{H}_u^{(2,2)}(\mathbf{X}_2) \\ \cdots \\ \mathbf{H}_u^{(2,2)}(\mathbf{X}_{n_V}) \end{bmatrix}^T \begin{bmatrix} \mathbf{H}_u^{(2,2)}(\mathbf{X}_1) \\ \mathbf{H}_u^{(2,2)}(\mathbf{X}_2) \\ \cdots \\ \mathbf{H}_u^{(2,2)}(\mathbf{X}_{n_V}) \end{bmatrix}, \tag{3.91}$$

$$\mathbf{b}^{(2)}\,(\bar{\mathbf{u}}_v) = \begin{bmatrix} \mathbf{H}_u^{(2,2)}(\mathbf{X}_1) \\ \mathbf{H}_u^{(2,2)}(\mathbf{X}_2) \\ \cdots \\ \mathbf{H}_u^{(2,2)}(\mathbf{X}_{n_V}) \end{bmatrix}^T \bar{\mathbf{u}}_v = [\bar{\mathbf{H}}^{(2)}]^T \bar{\mathbf{u}}_v\,. \tag{3.92}$$

The matrices $\mathbf{H}^{(2,2)}$ and $\bar{\mathbf{H}}^{(2)}$ are evaluated at the coordinates of the n_V boundary nodes of the virtual element. The vector $\mathbf{b}^{(2)}\,(\bar{\mathbf{u}}_v)$ has length (12×1). Once the unknowns $\mathbf{c}$ are found from solving (3.90) the moment $\mathbf{m}_1$ can be replaced using (3.86)

$$\mathbf{m}_1(\bar{\mathbf{u}}_v) = \int_{\Omega_v} \mathbf{H}_u^{(2,2)}(X, Y)\, \mathrm{d}\Omega \; [\mathbf{G}^{(2)}]^{-1}\, [\bar{\mathbf{H}}^{(2)}]^T \bar{\mathbf{u}}_v \tag{3.93}$$

The integral on the right hand side of (3.93) contains only polynomials of second order and thus can be exactly evaluated using the formulae in Appendix A. With (3.93) the moment $\mathbf{m}_1$ is expressed in terms of the displacement unknowns and by plugging this result into (3.70) the ansatz for a quadratic serendipity element is provided analogous to (3.84) and (3.75)

$$\boldsymbol{u}_\pi = \mathbb{P}_{uS}^{(2,2)}(X, Y)\, \bar{\mathbf{u}}_v \quad \text{and} \quad \nabla \boldsymbol{u}_\pi = \mathbf{B}_{u\pi}^{(2,2)}(X, Y)\, \mathbb{P}_{\nabla S}^{(2,2)}\, \bar{\mathbf{u}}_v \tag{3.94}$$

where the index S indicates the ansatz for the serendipity element.

Remark 3.4 A fundamental issue distinguishes serendipity isoparametric finite elements from serendipity virtual elements in the case of distorted meshes. It is related to the polynomial completeness of the ansatz functions considered. More precisely, it is well-established for 8-noded serendipity finite elements that the second order polynomial is complete in the reference space (ξ, η) of the isoparametric mapping. It loses its completeness during the transformation into the Cartesian coordinates

(X, Y) once a distorted quadrilateral element is mapped. The order of accuracy of the serendipity finite element is thus reduced to first order. On the other hand, the serendipity virtual elements are directly defined on the actual element geometry, without resorting to any parametric mapping, and consequently the second order polynomial completeness is not altered even in the case of distorted elements, hence the approximation order is not changed. Within the finite element methodology the polynomial completeness can be kept by using a 9-node isoparametric element which is complete in the reference space (ξ, η) and the Cartesian space (X, Y) for straight edges. By condensing the unknowns of the 9th-middle node out, also a serendipity type of element can be created for FEM. ■

3.1.8 Computation of the Second Order Projection Using Automatic Differentiation

A general way to compute the Galerkin projection in (3.18) introduces the gradients together with the L^2-projection for the displacements (3.34). This yields all unknown parameters $\mathbf{A} = a_{ij}$ in one step in terms of the unknown nodal values $\mathbf{u}_v$. The L^2-projection of the vector $\boldsymbol{u}_\pi$ and the Galerkin projection and its gradient $\nabla \boldsymbol{u}_\pi$ is carried out in the virtual element Ω_v

$$\int\limits_{\Omega_v} \left[\boldsymbol{u}_0^p \cdot (\boldsymbol{u}_h - \boldsymbol{u}_\pi) + \nabla \boldsymbol{u}^p \cdot (\nabla \boldsymbol{u}_h - \nabla \boldsymbol{u}_\pi)\right] \mathrm{d}\Omega = 0. \tag{3.95}$$

The quadratic test function $\boldsymbol{u}^p = \mathbf{A}^p [\mathbf{N}_\pi^{(2)}]^T$, see (3.12), is defined in the same space as $\boldsymbol{u}_\pi = \mathbf{A}[\mathbf{N}_\pi^{(2)}]^T$ i.e. $\boldsymbol{u}^p \in \mathbb{P}_2$ with

$$\mathbf{N}_\pi^{(2)} = \langle\, 1\; X\; Y\; X^2\; XY\; Y^2 \,\rangle \tag{3.96}$$

With these definitions the gradient $\nabla \boldsymbol{u}^p$ and the constant part[8] $\boldsymbol{u}_0^p = \left[a_{11}^p \; a_{21}^p\right]^T$ can be introduced. In the two-dimensional case $\mathbf{A}_p$ is a matrix of coefficients a_{ij}^p, $i \in \{1, 2\} \wedge j \in \{1, 2, ..., 6\}$, as well as $\mathbf{A}$, for the quadratic ansatz and

$$\begin{aligned} \mathbf{A}_p &= \bigcup_{i=1}^{2} \bigcup_{j=1}^{6} a_{ij}^p = \left[\ldots a_{ij}^p \ldots\right] \\ \mathbf{A} &= \bigcup_{i=1}^{2} \bigcup_{j=1}^{6} a_{ij} = \left[\ldots a_{ij} \ldots\right] \end{aligned} \tag{3.97}$$

[8] As defined, the test function $\boldsymbol{u}^p$ is a quadratic polynomial, but since all the constants a_{ij}^p related to the polynomial terms with X and Y are contained in $\nabla \boldsymbol{u}^p$ it is sufficient to use in the first term of (3.95) only the constant part.

are vector of all coefficient a_{ij}^p and a_{ij}, respectively. By automatic differentiation with respect to the unknowns $\mathbf{A}^p$ and $\mathbf{A}$ the vector $\mathbf{b}_\pi^{(2)}$ and the matrix $\mathbf{G}_\pi^{(2)}$ can be obtained without going through the matrix calculus of the previous section.

Since it is not possible to integrate over the element volume Ω_v Eq. (3.95) has to be reformulated on the left- and right-hand side, where the part g corresponds to the projection $\boldsymbol{u}_\pi$ and the part b to the approximation $\boldsymbol{u}_h$

$$\underbrace{\int\limits_{\Omega_v} \left(\boldsymbol{u}_0^p \cdot \boldsymbol{u}_\pi + \nabla \boldsymbol{u}^p \cdot \nabla \boldsymbol{u}_\pi\right) \mathrm{d}\Omega}_{g} = \underbrace{\int\limits_{\Omega_v} \left(\boldsymbol{u}_0^p \cdot \boldsymbol{u}_h + \nabla \boldsymbol{u}^p \cdot \nabla \boldsymbol{u}_h\right) \mathrm{d}\Omega}_{b} . \tag{3.98}$$

As in (3.20) the integrals of the gradients (3.98) are transformed to the boundary by utilizing the divergence or Gauss theorem

$$\int\limits_{\Omega_v} \nabla \boldsymbol{u}^p \cdot \nabla \boldsymbol{u}_\pi \mathrm{d}\Omega = \int\limits_{\Gamma_v} \nabla \boldsymbol{u}^p \boldsymbol{N} \cdot \boldsymbol{u}_\pi \mathrm{d}\Gamma - \int\limits_{\Omega_v} \Delta \boldsymbol{u}^p \cdot \boldsymbol{u}_\pi \mathrm{d}\Omega \tag{3.99}$$

$$\int\limits_{\Omega_v} \nabla \boldsymbol{u}^p \cdot \nabla \boldsymbol{u}_h \mathrm{d}\Omega = \int\limits_{\Gamma_v} \nabla \boldsymbol{u}^p \boldsymbol{N} \cdot \boldsymbol{u}_h \mathrm{d}\Gamma - \int\limits_{\Omega_v} \Delta \boldsymbol{u}^p \cdot \boldsymbol{u}_h \mathrm{d}\Omega . \tag{3.100}$$

Gradient $\nabla \boldsymbol{u}^p$, given in tensor notation, and Laplacian $\Delta \boldsymbol{u}^p$ have the explicit form

$$\nabla \boldsymbol{u}^p = \begin{bmatrix} (a_{12}^p + a_{14}^p Y + 2a_{15}^p X) & (a_{13}^p + a_{14}^p X + 2a_{16}^p Y) \\ (a_{22}^p + a_{24}^p Y + 2a_{25}^p X) & (a_{23}^p + a_{24}^p X + 2a_{26}^p Y) \end{bmatrix} \tag{3.101}$$

$$\text{and } \Delta \boldsymbol{u}^p = \nabla \cdot \nabla \boldsymbol{u}^p = 2\left[(a_{15}^p + a_{16}^p) \;\; (a_{25}^p + a_{26}^p)\right]^T \tag{3.102}$$

with $N = \left[\, N_X \;\; N_Y \,\right]^T$ being the outward unit normal at Γ_v. The Laplacian $\Delta \mathsf{u}^p$, is equivalent to the divergence of the gradient of $\boldsymbol{u}_p$ and has a constant value within the element domain Ω_v. Since the terms $\boldsymbol{u}_0^p$ and $\Delta \boldsymbol{u}^p$ in (3.95), (3.99) and (3.100) are constant, g and b can be written as

$$g = \int\limits_{\Gamma_v} \nabla \boldsymbol{u}^p \mathsf{N} \cdot \boldsymbol{u}_\pi \mathrm{d}\Gamma - (\Delta \boldsymbol{u}^p - \boldsymbol{u}_0^p) \cdot \int\limits_{\Omega_v} \boldsymbol{u}_\pi \mathrm{d}\Omega , \tag{3.103}$$

$$b = \int\limits_{\Gamma_v} \nabla \boldsymbol{u}^p \mathsf{N} \cdot \boldsymbol{u}_h \mathrm{d}\Gamma - (\Delta \boldsymbol{u}^p - \boldsymbol{u}_0^p) \cdot \int\limits_{\Omega_v} \boldsymbol{u}_h \mathrm{d}\Omega . \tag{3.104}$$

Here only the integrals $\int_{\Omega_v} \boldsymbol{u}_\pi$ and $\int_{\Omega_v} \boldsymbol{u}_h$ require special treatment. The area integral of the polynomial projection $\boldsymbol{u}_\pi = \{u_{\pi 1} , u_{\pi 2}\}^T$ can be evaluated over the edge Γ_v by using the divergence theorem component-wise ($i = 1, 2$)

$$\int\limits_{\Omega_v} u_{\pi i} \mathrm{d}\Omega = \frac{1}{2} \int\limits_{\Gamma_v} \begin{bmatrix} \int u_{\pi i} dX \\ \int u_{\pi i} dY \end{bmatrix} \cdot \boldsymbol{N} \mathrm{d}\Gamma = \frac{1}{2} \int\limits_{\Gamma_v} \textstyle\int \boldsymbol{u}_{\pi i} \cdot \boldsymbol{N} \, d\Gamma \,. \tag{3.105}$$

The analytical integration of $\int u_{\pi i}$ can be performed directly by integrating each term of the polynomial projection function $\mathbf{N}_\pi$, see (3.12),

$$\textstyle\int \boldsymbol{u}_{\pi i} = \boldsymbol{a}_i \, \mathcal{I}_\pi = \begin{bmatrix} a_{i1} X + a_{i2} \frac{X^2}{2} + a_{i3} XY + a_{i4} \frac{X^2 Y}{2} + a_{i5} \frac{X^3}{3} + a_{i6} X Y^2 \\ a_{i1} Y + a_{i2} XY + a_{i3} \frac{Y^2}{2} + a_{i4} \frac{XY^2}{2} + a_{i5} X^2 Y + a_{i6} \frac{X^3}{3} \end{bmatrix}^T , \tag{3.106}$$

$$\mathcal{I}_\pi = \begin{bmatrix} \mathcal{I}_{\pi X} \\ \mathcal{I}_{\pi Y} \end{bmatrix}^T = \begin{bmatrix} \int \mathbf{N}_\pi \, dX \\ \int \mathbf{N}_\pi \, dY \end{bmatrix}^T = \begin{bmatrix} X & \frac{X^2}{2} & XY & \frac{X^2 Y}{2} & \frac{X^3}{3} & XY^2 \\ Y & XY & \frac{Y^2}{2} & \frac{XY^2}{2} & X^2 Y & \frac{Y^3}{3} \end{bmatrix}^T . \tag{3.107}$$

The integral of $\boldsymbol{u}_h$ cannot be evaluated, thus it will be treated as an internal unknown and replaced by the internal moments $\mathbf{m}_1 = \frac{1}{\Omega_v} \int_\Omega \boldsymbol{u}_h \, d\Omega$, see also (3.70). However at this point the serendipity concept of Sect. 3.1.7 is utilized, thus $\mathbf{m}_1$ can be eliminated at element level. This leads to a substitution of the internal moment $\mathbf{m}_1 = \{m_{1x} \, , m_{1y}\}^T$ by a function $\boldsymbol{u}^c = \{u^c_x \, , u^c_y\}^T$ which depends only on the nodal degrees of freedom

$$m_{1i} = \int\limits_{\Omega_v} u^c_i \, d\Omega = \frac{1}{2} \int\limits_{\Gamma_v} \begin{bmatrix} \int u^c_i dX \\ \int u^c_i dY \end{bmatrix} \cdot \boldsymbol{N} \, d\Gamma = \frac{1}{2} \int\limits_{\Gamma_v} \textstyle\int \boldsymbol{u}^c_i \cdot \boldsymbol{N} \, . \, d\Gamma \tag{3.108}$$

The integral $\int u^c_i$ can be evaluated in the same manner as $\int u_{\pi i}$ in (3.106). Note that only the scalars g and b have to be computed. The matrices that are needed to compute the projection follow by differentiation with respect to the coefficients $\mathbf{A}$ and $\mathbf{A}^p$. This avoids the extensive matrix algebra considered in the previous sections.

Based on this approach the unknowns $\mathbf{A}$ in (3.97), related to the projection, can be obtained by solving two linear systems of equations since differentiation of g leads to a matrix that can be efficiently formulated by a block structure. By defining the coefficient matrices

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} \quad \text{and} \quad \mathbf{A}^p = \begin{bmatrix} \mathbf{a}^p_1 \\ \mathbf{a}^p_2 \end{bmatrix} \tag{3.109}$$

the equation system

$$g = b \longrightarrow \mathbf{G}_j \mathbf{a}_j = \mathbf{b}_j \quad \rightarrow \quad \mathbf{a}_j = \mathbf{G}_j^{-1} \mathbf{b}_j , \quad \text{for } j = 1, 2 \tag{3.110}$$

results from differentiation. The left-hand side is defined as

$$\mathbf{G}_j = \frac{\partial^2 g}{\partial \mathbf{a}_j^p \, \partial \mathbf{a}_j} \tag{3.111}$$

$$\text{with}\, \mathbf{G}_j = \int\limits_\Gamma \left[(\nabla \mathbf{N}_\pi^{(2)} \boldsymbol{N}) \, [\mathbf{N}_\pi^{(2)}]^T + \frac{1}{2} \mathbf{C}_j^p \, (\mathcal{I}_\pi \boldsymbol{N})^T \right] d\Gamma, \tag{3.112}$$

where $\mathbf{G}_1$ and $\mathbf{G}_2$ are (6×6) matrices. Explicit expressions of the terms in (3.112) are provided by

$$\begin{aligned} \nabla \mathbf{N}_\pi &= \begin{bmatrix} 0 & 1 & 0 & Y & 2X & 0 \\ 0 & 0 & 1 & X & 0 & 2Y \end{bmatrix}^T , \\ \nabla \mathbf{N}_\pi \boldsymbol{N}_0 &= \begin{bmatrix} 0 & N_X & N_Y & N_X Y + N_Y X & 2 N_X X & 2 N_Y Y \end{bmatrix}^T , \\ \mathcal{I}_\pi \boldsymbol{N} &= N_X \mathcal{I}_{\pi X} + N_Y \mathcal{I}_{\pi Y} \, , \\ \text{and} \qquad \mathbf{C}_j^p &= \frac{\partial \left(\boldsymbol{u}_0^p - \Delta \boldsymbol{u}^p \right)_j}{\partial \mathbf{a}_j^p} = \begin{bmatrix} 1 & 0 & 0 & 0 & -2 & -2 \end{bmatrix}^T . \end{aligned} \tag{3.113}$$

The right-hand is given by

$$\mathbf{b}_j = \frac{\partial b}{\partial \mathbf{a}_i^p} \tag{3.114}$$

with

$$\begin{aligned} \mathbf{b}_j &= \int\limits_\Gamma \left[(\nabla \mathbf{N}_\pi^{(2)} \boldsymbol{N}) \mathbf{u}_j + \frac{1}{2} \mathbf{C}_j^p \textstyle\int \boldsymbol{u}_j^c \cdot \boldsymbol{N} \right] d\Gamma \\ \textstyle\int \boldsymbol{u}_j^c \cdot \boldsymbol{N} &= \mathbf{a}_1^c N_X \mathcal{I}_{\pi X} + \mathbf{a}_2^c N_Y \mathcal{I}_{\pi Y} \end{aligned} \tag{3.115}$$

where in $\mathbf{u}_j$ the components of the nodal displacement vector in X and Y direction are contained in a suitable manner. The parameters $\mathbf{a}_1^c$ and $\mathbf{a}_2^c$ are computed below. These parameters depend directly on the nodal displacements of the virtual element.

Elimination of internal moment $\mathbf{m}_1$. In a quadratic serendipity virtual element the internal degrees of freedom, moments $\mathbf{m}_1$ are eliminated. This results from expressing $\mathbf{m}_1$ explicitly by a function $\boldsymbol{u}^c$ in terms of the element nodal degrees of freedom $\hat{\mathbf{u}}_v$

$$\mathbf{m}_1 = \int_{\Omega_v} \boldsymbol{u}^c(\hat{\mathbf{u}}_v) \, d\Omega, \tag{3.116}$$

According to Beirão da Veiga et al. (2016), De Bellis et al. (2019) this function, see also (3.86), can be written as

$$\boldsymbol{u}^c(\hat{\mathbf{u}}_v) = \mathbf{A}^c \, \mathbf{N}_\pi^{(2)} \tag{3.117}$$

where $\boldsymbol{u}^c(\hat{\mathbf{u}}_v)$ is assumed to be a second order polynomial, with $\mathbf{N}_\pi^{(2)}$ being defined in (3.96). The matrix $\mathbf{A}^c$ contains the unknown coefficients $c_{ij}, i \in \{1, 2\} \wedge j \in$

$\{1, 2, .., 6\}$ and $\mathbf{a}^c = \big[\dots c_{ij} \dots \big]$ is a vector of length 12 with all unknown coefficients, see (3.97).

To obtain the coefficients $\mathbf{A}^c = \{\mathbf{a}_1^c\,, \mathbf{a}_2^c\}^T$ the following Galerkin type projection is performed with $\boldsymbol{u}_\pi$ as a test function and $\mathbf{u}_i$ as nodal displacements

$$\sum_{i=1}^{n_{en}} \boldsymbol{u}_\pi(\mathbf{X}_i) \cdot \big(\boldsymbol{u}^c(\mathbf{X}_i) - \mathbf{u}_i\big) = 0 \longrightarrow g^c = b^c \tag{3.118}$$

with

$$g^c = \sum_{i=1}^{n_{en}} \boldsymbol{u}^c(\mathbf{X}_i) \cdot \boldsymbol{u}_\pi(\mathbf{X}_i) \quad \text{and} \quad b^c = \sum_{i=1}^{n_{en}} \boldsymbol{u}_\pi(\mathbf{X}_i) \cdot \mathbf{u}_i \tag{3.119}$$

The coefficients $\mathbf{a}_j^c$ can be obtained by solving the linear system of equation

$$\mathbf{G}_j^c \mathbf{a}_j^c = \mathbf{b}_j^c \quad \rightarrow \quad \mathbf{a}_j^c = (\mathbf{G}_j^c)^{-1} \mathbf{b}_j^c \tag{3.120}$$

where the matrix on the left-hand side is given by

$$\mathbf{G}_j^c = \frac{\partial^2 g^c}{\partial \mathbf{a}_j^c \partial \mathbf{a}_j} \quad \text{with } \mathbf{G}_j^c = \sum_{i=1}^{n_{en}} [\mathbf{N}_\pi^{(2)}(\mathbf{X}_i)]^T \mathbf{N}_\pi^{(2)}(\mathbf{X}_i) \tag{3.121}$$

and the right-hand by

$$\mathbf{b}_j^c = \frac{\partial b^c}{\partial \mathbf{a}_j} = \sum_{i=1}^{2n_v} \mathbf{N}_\pi^{(2)}(\mathbf{X}_i)\, u_{ij}\,. \tag{3.122}$$

where u_{ij} are the components of the nodal degrees of freedom with $1 \leq i \leq n_v$ and $1 \leq j \leq 2$. Note again that only the scalars g^c and b^c have to be formed and the matrices follow from differentiation.

3.1.9 Higher Order Ansatz for Virtual Elements

One can now utilize the formulations provided so far to design virtual elements with cubic or even higher ansatz order ($n > 2$). Here the derivation for the quadratic ansatz, see Sect. 3.1.6, can be used by just inserting the higher order ansatz functions. The general formulation does not change and was proposed in many papers for different applications. For elasticity we refer to e.g. Beirão da Veiga et al. (2013b), Artioli et al. (2017a) and for Poisson equations to Dassi and Mascotto (2018). Some of these include nonlinear applications, see e.g. Beirão da Veiga et al. (2015), and others include error estimation and adaptivity, see e.g. Beirão da Veiga et al. (2019b). Most of the papers that address virtual element methods are from the mathematical community. Hence the application range is often limited to test cases which illustrate the performance of the method so far VEM was not used in large real life problems.

Generally, in engineering the scope is large. Many applications, like metal forming, frictional contact and crack propagation result in non smooth solutions that have either to be tackled by special approaches or robust low order methods. This is the reason why we restrict ourselves in this book to ansatz spaces that are of polynomial order ($n \leq 2$). There are, of course, several applications in engineering, like rubber elasticity, where solutions are very smooth and thus higher order ansatz functions can be applied successfully, see e.g. Düster et al. (2003, 2001). Furthermore, in most engineering applications a methodology based quadratic ansatz functions yields results that have an accuracy which is sufficient with respect to other approximations and assumptions made during the modeling of a problem, e.g. material parameters and loading conditions.

A good overview with respect to the formulation of higher order virtual element methods can be found for the two-dimensional problems in elasticity in Artioli et al. (2017a). For three-dimensional cases of the Poisson equation general formulations are reported in Ahmad et al. (2013), Dassi and Mascotto (2018), for elasticity problems in Gain et al. (2014) and for magnetostatic problems in Beirão da Veiga et al. (2018).

3.1.10 Virtual Elements Ansatz Functions for Curved Surfaces

In the last section it was assumed that the virtual elements have straight edges which allows a simple integration. For general applications it might be preferable to introduce virtual elements with curved faces which are e.g. advantageous when complex geometries with curved boundaries are present.

First developments can be found in Beirão da Veiga et al. (2019a, d) for elliptic problems. Higher order ansatz spaces were discussed in Bertoluzza et al. (2019) where a Nitsche method was employed to incorporate curved boundaries. Some enhancements of the formulation for virtual elements with curved edges were provided by Beirão da Veiga et al. (2020) who introduced a polynomial preserving formulation.

Applications to solid mechanics can be found in Artioli et al. (2020a) for small strains and in Wriggers et al. (2020) for finite strains where in the latter a specific formulation was developed which is based on three configurations. The idea is to define virtual elements with straight edges in a reference configuration and then map from this reference configuration to an initial configuration with curved edges. Loading is then applied in the initial configuration which allows to compute the deformed configuration. This method was further developed in Wriggers et al. (2021b) to include NURBS based mappings from the reference to the initial configuration.

A mixed formulation for virtual elements using curved elements can be found in Dassi et al. (2021). Homogenization with circular inclusions that are modeled by curved edges is discussed in Artioli et al. (2020b). The ansatz functions developed in Beirão da Veiga et al. (2019d, 2020a) have also been applied in Aldakheel et al. (2020) to solve contact problems.

Most of these formulations need special quadrature rules for the integration of ansatz functions over the curved edges and elements. Associated integration methods can be found in e.g. Chin and Sukumar (2020).

3.2 Three-Dimensional Case

Virtual elements for three-dimensional problems in engineering are generally constructed in the same way as virtual elements in two dimensions. However their geometry is more rich since the elements have polyhedral shape with faces and edges as depicted in the discretization of a mesh with virtual elements in Fig. 3.8.

As polygons in two dimensions polyhedra can have an arbitrary number of vertices and their shape can be convex and non-convex. Thus, these elements are ideally suited for a direct meshing of solids with a polycrystalline microstructures.

In a three-dimensional setting the domain Ω is partitioned into non-overlapping polyhedral elements Ω_v. In the following it will be assumed that the polyhedral elements Ω_v consist of plane faces and straight edges. Besides that, the element can assume an arbitrary shape. An example of a polyhedral virtual element is depicted in Fig. 3.9.

A face in the polyhedral element is denoted by Γ_f where f is an index $1 \leq f \leq n_F$. The entire surface Γ of a polyhedral element is covered by n_F faces. The element is assumed to have n_V vertices denoted by the index k, related to the nodes with $1 \leq k \leq n_V$. Each of the faces has edges γ_e which are shared by two faces. The virtual element consists of n_E edges with $1 \leq e \leq n_E$.

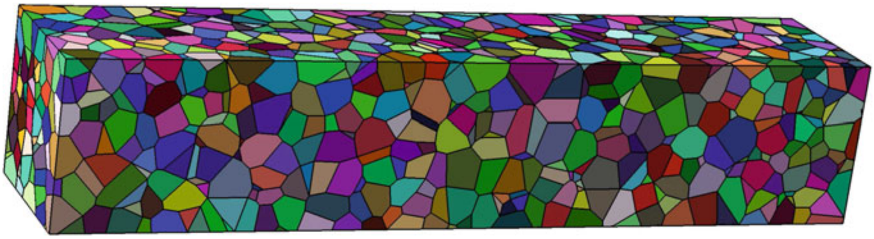


Fig. 3.8 Mesh with three-dimensional virtual elements

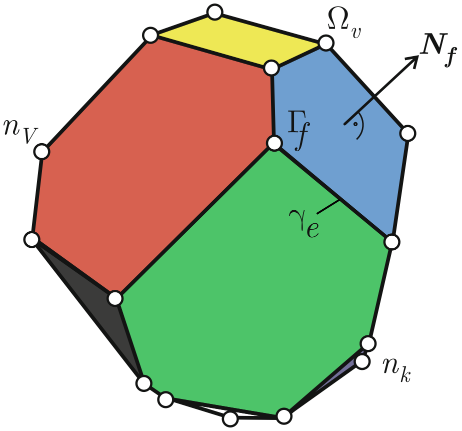


Fig. 3.9 Polyhedral VEM element

3.2.1 General Ansatz Space in Three Dimensions

As in the two-dimensional case, the ansatz function for the displacement field $\boldsymbol{u}_h$ within a virtual element Ω_v has specific properties

- (a0): $\boldsymbol{u}_h$ is known at the vertices k of the polyhedron Ω_v,
- (a1): $\boldsymbol{u}_h$ is a polynomial P_n of degree n at each edge γ_e of the face Γ_f,
- (a2): $\boldsymbol{u}_h$ is continuous at all edges $\gamma_e \in \Gamma_f$ of the polyhedron Ω_v,
- (a3): $\text{Div}(\text{Grad}\,\boldsymbol{u}_h) = \Delta\boldsymbol{u}_h$ is a polynomial of degree $< n - 2$ on the face Γ_f,
- (a4): $\text{Div}(\text{Grad}\,\boldsymbol{u}_h) = \Delta\boldsymbol{u}_h$ is a polynomial of degree $< n - 2$ in the element Ω_v.

Following this definition the ansatz for the displacement field $\boldsymbol{u}_h$ is known at the edges γ_e of Ω_v, but the ansatz is not known inside the faces Γ_f nor inside the polyhedron Ω_v. Additionally we have $[\mathsf{P}_n]^3 \subseteq \mathbf{V}_h$. However, as shown in Beirão da Veiga et al. (2014), the restriction of the ansatz $\boldsymbol{u}_h$ on all faces Γ_f belongs to the space W_k defined in Sect. 3.1.3. This assumption is needed for the evaluation of the integrals. All these properties guarantee the convergence of the virtual element method and the computability of the projector $\boldsymbol{u}_\pi$. For mathematical details see Beirão da Veiga et al. (2013b). We note, that with the result in (3.69) one can exchange (a1) by the statement that $\boldsymbol{u}_h$ is known at the Gauss-Lobatto points on the edge, related to the ansatz order n, see also Dassi and Mascotto (2018).

In the three dimensional case we have in solid mechanics 3 unknowns per node. Thus 3 n_V unknowns are related to the n_V vertices of the polynomial, $3\,n_E(n-1)$ unknowns to the additional nodes (Gauss-Lobatto points) at each edge γ_e and $3\,n_F\,\frac{n\,(n-1)}{2}$ unknown moments at the faces Γ_f and $3\frac{n\,(n-1)}{2}$ moments in Ω_v. For the special cases of linear ($n = 1$) and quadratic ($n = 2$) polynomials P_n the number of unknowns within a virtual element Ω_v are

$$
\begin{aligned}
n = 1: &\qquad 3\,n_V \\
n = 2: &\qquad 3\,n_V + 3\,n_E + 3\,n_F + 3
\end{aligned}
$$

It is obvious that moments do not appear in a virtual element space that is linear, $(n = 1)$.

The ansatz function is directly formulated in the coordinates (X, Y, Z) related to the initial geometry of the solid. The displacement in Ω_v is approximated by an ansatz using a complete polynomial of order n with length $l_n = \text{length}(\mathbf{N}_\pi)$. Since the ansatz $\boldsymbol{u}_h$ is only known at the edges the displacement gradient will be approximated by a projection as in the two-dimensional case.

An explicit ansatz for (a1) can be formulated at each edge γ_e of a polyhedron by the interpolation defined in (3.4) with the special shape functions provided for $(n = 1)$ in (3.5) and for $(n = 2)$ in (3.6).

By using the same ansatz for an edge in all of the virtual elements used to discretize an engineering problem, (a2) is fulfilled automatically. As in the two-dimensional case this ansatz has to be connected to the displacement gradient in the virtual element Ω_v. For that we introduce a gradient projector in Voigt notation of size (9×1)

$$
\nabla \boldsymbol{u}_\pi = \begin{Bmatrix} u_{\pi x,x} \\ u_{\pi x,y} \\ u_{\pi x,z} \\ u_{\pi y,x} \\ \dots \\ u_{\pi z,z} \end{Bmatrix} = \mathbf{B}_{u\,\pi}^{(3,n)}(X, Y, Z)\,\hat{\mathbf{a}} \tag{3.123}
$$

where the matrix $\mathbf{B}_{u\,\pi}^{(3,n)}(X, Y, Z)$ includes complete polynomials in all coordinate directions of order $(n - 1)$ and $\hat{\mathbf{a}}$ are unknown parameters.

Next we introduce moments $\mathbf{m}_{(\alpha,\beta)}$ in accordance with (a3) and (a4). These moments appear only for $(n > 1)$ and are integrals of the single terms of a complete polynomial of degree $(n - 2)$, see e.g (3.7). This leads to additional unknowns (moments), which are only reported here for $(n = 2)$

$$
\text{face}: \qquad \mathbf{m}_1^\Gamma = \frac{1}{\Gamma_f} \int\limits_{\Gamma_f} \boldsymbol{u}_h \,\mathrm{d}\Omega \tag{3.124}
$$

$$
\text{volume}: \qquad \mathbf{m}_1^\Omega = \frac{1}{\Omega_v} \int\limits_{\Omega_v} \boldsymbol{u}_h \,\mathrm{d}\Omega \tag{3.125}
$$

where $\mathbf{m}_1^\Gamma$ is the moment related to the faces and $\mathbf{m}_1^\Omega$ is the moment related to the volumetric part of the polyhedron. As already shown in Sects. 3.1.2 and 3.1.6, these moments are needed to compute the projection $\nabla \boldsymbol{u}_\pi$.

3.2.2 Computation of the Projection in Three Dimensions

Again a Galerkin projection of the gradients is employed to obtain the parameters $\hat{\mathbf{a}}$ for the computation of $\nabla \boldsymbol{u}_\pi$ as in in (3.18)

$$\int\limits_{\Omega_v} \nabla \mathbf{N}_\pi \cdot (\nabla \boldsymbol{u}_h - \nabla \boldsymbol{u}_\pi)\ \mathrm{d}\Omega = 0 \tag{3.126}$$

The polynomial function $\mathbf{N}_\pi$ has the same order as $\boldsymbol{u}_\pi$. Its gradient $\nabla \mathbf{N}_\pi$ is applied as weighting function. Equation (3.126) leads with (3.19) to

$$\int\limits_{\Omega_v} \nabla \mathbf{N}_\pi \cdot \nabla \boldsymbol{u}_\pi\ \mathrm{d}\Omega = \underbrace{\int\limits_{\Gamma_v} \nabla \mathbf{N}_\pi \cdot (\boldsymbol{u}_h \otimes \boldsymbol{N})\ \mathrm{d}\Gamma}_{b_\Gamma} - \underbrace{\int\limits_{\Omega_v} \div [\nabla \mathbf{N}_\pi] \cdot \boldsymbol{u}_h\ \mathrm{d}\Omega}_{b_{div}} \tag{3.127}$$

where $\boldsymbol{N}$ is the outward normal related to the faces Γ, see Fig. 3.9. The difference to the two-dimensional case is that the surface integral has now to be evaluated with respect to the faces.

Using the same approach as in the two-dimensional case, we introduce for the gradient of the polynomial $\nabla \mathbf{N}_\pi = \mathbf{B}^{(3,n)}_{u\pi}(X, Y, Z)\,\mathbf{c}$. With this choice the term on the left side of (3.127) can be written as

$$\int\limits_{\Omega_v} \nabla \mathbf{N}_\pi \cdot \nabla \boldsymbol{u}_\pi\ \mathrm{d}\Omega = \mathbf{c}^T \int\limits_{\Omega_v} [\mathbf{B}^{(3,n)}_{u\pi}(X, Y, Z)]^T \mathbf{B}^{(3,n)}_{u\pi}(X, Y, Z)\ \mathrm{d}\Omega\,\hat{\mathbf{a}} = \mathbf{c}^T\, \mathbf{G}^{(n)}_\pi\, \hat{\mathbf{a}} \tag{3.128}$$

Matrix $\mathbf{G}^{(n)}_\pi$ is a unit matrix for ($n = 1$) and includes polynomials in X, Y and Z up to second order for ($n = 2$). The integral can be evaluated exactly, see e.g. Bernardini (1999) and De Loera et al. (2013).

The last term on the right hand side of (3.127) cannot simply be computed as in the two-dimensional case. This is due to the fact that $\boldsymbol{u}_h$ only is known along the edges, but not on the faces defining Γ_v. The first term on the right hand side can be discretized as

$$b_\Gamma = \int\limits_{\Gamma_v} \nabla \mathbf{N}_\pi \cdot (\boldsymbol{u}_h \otimes \boldsymbol{N})\ \mathrm{d}\Gamma = \sum_{f=1}^{n_F} \mathbf{c}^T \int\limits_{\Gamma_f} [\mathbf{B}^{(3,n)}_{u\pi}(X, Y, Z)]^T \widehat{\mathbf{N}}_f\, \mathbf{u}_{hf}\ \mathrm{d}\Gamma \tag{3.129}$$

where the dyadic product $\boldsymbol{u}_h \otimes \boldsymbol{N}$ is written in matrix form as $\widehat{\mathbf{N}}_f\, \mathbf{u}_{hf}$ with the (9×3) matrix $\widehat{\mathbf{N}}_f$ and the (3×1) matrix $\mathbf{u}_{hf}$

$$\widehat{\mathbf{N}}_f = \begin{bmatrix} \boldsymbol{N}_f & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{N}_f & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{N}_f \end{bmatrix} \text{ with } \boldsymbol{N}_f = \begin{Bmatrix} N_x \\ N_y \\ N_z \end{Bmatrix}^f , \; \mathbf{0} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} , \text{ and } \mathbf{u}_{hf} = \begin{Bmatrix} u_x \\ u_y \\ u_z \end{Bmatrix}_h^f \tag{3.130}$$

leading to a (9 × 1) vector that correspond to the matrix $\mathbf{B}_{u\,\pi}^{(3,n)}$. The normal $\boldsymbol{N}_f$ to the face Γ_f is defined in equation (3.130).

In Eq. (3.129) $\mathbf{u}_{hf}$ is not known in Γ_f. It has to be computed for each face Γ_f in terms of the nodal vertices at the edges of the face Γ_f which are the unknowns, see (a0). This computation can be performed in different ways. By using the space $\mathbf{W}_h$ with assumption (a3)", see Sect. 3.1.2 we can replace $\mathbf{u}_{hf}$ by $\mathbf{u}_\pi^0$ which is equivalent to $\mathbf{u}_\pi$ for ansatz order $n = 1$ and $n = 2$.

The last term on the right hand side

$$b_{div} = \mathbf{c}^T [\mathbf{B}_{\pi\,div}^{(3)}]^T \begin{Bmatrix} m_{x1} \\ m_{y1} \\ m_{z1} \end{Bmatrix}^\Omega = \mathbf{c}^T [\mathbf{B}_{\pi\,div}^{(3)}]^T \, \mathbf{m}_1^\Omega \tag{3.131}$$

appears only for $n = 2$ where $\mathbf{B}_{\pi\,div}^{(3)}$ is constructed in the same way as in (3.69).

Once both terms in (3.127) are computed and the dependency on the nodal degrees of freedom and the moments is established, like in the two-dimensional case in (3.27), the unknown parameters $\hat{\mathbf{a}}$ follow from an equation system with the coefficient matrix (3.128) and the right hand sides (3.131) and (3.129). These depend on the unknowns $\mathbf{u}_v$, $\mathbf{m}^\Omega$ and $\mathbf{m}^\Gamma$. The procedure to compute the projection using $\boldsymbol{u}_\pi^0$ in the three-dimensional case will be worked out for a linear ansatz order in the next section. For higher order virtual element formulations, see e.g. Dassi and Mascotto (2018) and Beirão da Veiga et al. (2018).

3.2.3 Projection for Linear Ansatz in Three Dimensions

The computation of the projection $\boldsymbol{u}_\pi$ for three-dimensional virtual elements is described in many papers, see e.g. Beirão da Veiga et al. (2014); Dassi and Mascotto (2018) for linear problems. For finite strain applications in solid mechanics we refer to Hudobivnik et al. (2018). First we employ the anatz discussed in the previous section and describe two evaluations of the surface integral in (3.129) including the one used in Hudobivnik et al. (2018).

The projection $\boldsymbol{u}_\pi$ is defined at element level by the linear ansatz in three dimensions

$$\boldsymbol{u}_\pi = \mathbf{H}_u^{(3,1)}(X, Y, Z)\, \mathbf{a} = \begin{bmatrix} 1 & 0 & 0 & X & 0 & 0 & Y & 0 & 0 & Z & 0 & 0 \\ 0 & 1 & 0 & 0 & X & 0 & 0 & Y & 0 & 0 & Z & 0 \\ 0 & 0 & 1 & 0 & 0 & X & 0 & 0 & Y & 0 & 0 & Z \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ \dots \\ a_{12} \end{Bmatrix} \tag{3.132}$$

The direct computation of the projected gradient yields with (3.132) a constant matrix $\mathbf{B}_{u\,\pi}^{(3,1)}$ of size (9×9) when the gradient is formulated in Voigt notation. This leads to the displacement gradient

$$\nabla \mathbf{u}_\pi = \begin{Bmatrix} u_{x,x} \\ u_{x,y} \\ u_{x,z} \\ u_{y,x} \\ \dots \\ u_{z,z} \end{Bmatrix} = \mathbf{B}_{u\,\pi}^{(3,1)}\,\hat{\mathbf{a}} = \begin{bmatrix} 1\;0\;0\;0\;0\;0\;0\;0\;0 \\ 0\;0\;0\;1\;0\;0\;0\;0\;0 \\ 0\;0\;0\;0\;0\;0\;1\;0\;0 \\ 0\;1\;0\;0\;0\;0\;0\;0\;0 \\ \cdots\cdots\cdots\cdots\cdots \\ 0\;0\;0\;0\;0\;0\;0\;0\;1 \end{bmatrix} \begin{Bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \dots \\ \hat{a}_9 \end{Bmatrix} \tag{3.133}$$

The displacement field at the edge γ_e of a face Γ_f is interpolated according to (a1). Here we assume a linear ansatz, see (3.4), which yields

$$\boldsymbol{u}_{h\,|\gamma_e} = \sum_{i=1}^{2} M_i(\xi)\,\mathbf{u}_i \tag{3.134}$$

where the ansatz functions $M_i(\xi)$ are defined in (3.5).

Next the integral in (3.128) has to be evaluated. It follows for the constant gradient immediately that matrix $\mathbf{G}$ is a (9×9) unit matrix multiplied with the volume of the virtual element

$$\mathbf{G}^{(1)} = \Omega_v\,\mathbf{I}\,. \tag{3.135}$$

The right hand side (3.131) is zero for a constant gradient $\nabla \mathbf{u}_\pi$. Thus it remains to compute the integral (3.129)

$$\sum_{f=1}^{n_F} \int\limits_{\Gamma_f} [\mathbf{B}_{u\,\pi}^{(3,1)}(X,Y,Z)]^T \widehat{\mathbf{N}}_f\,\boldsymbol{u}_h\,\mathrm{d}\Gamma = [\mathbf{B}_{u\,\pi}^{(3,1)}]^T \sum_{f=1}^{n_F} \widehat{\mathbf{N}}_f \int\limits_{\Gamma_f} \boldsymbol{u}_h\,\mathrm{d}\Gamma \tag{3.136}$$

We can now think of replacing $\boldsymbol{u}_h$ by $\boldsymbol{u}_\pi$ by using the fact that $\mathbf{u}_h$ on $\Gamma_f \in W_k$, see also (3.37).

Remark 3.5 Often it is simpler to use the tensor notation as a starting point, especially when applying automatic differentiation procedures, like provided in *AceGen*, see Korelc and Wriggers (2016). Then the projected gradient is provided for a linear ansatz by, see (3.60),

$$\nabla \boldsymbol{u}_\pi = \frac{1}{\Omega_v} \int\limits_{\Gamma_v} \boldsymbol{u}_h \otimes \boldsymbol{N}\,\mathrm{d}\Gamma \tag{3.137}$$

with the explicit form of the projected gradient in tensor notation

$$\nabla \boldsymbol{u}_\pi = \begin{bmatrix} u_{\pi x,x} & u_{\pi x,y} & u_{\pi x,z} \\ u_{\pi y,x} & u_{\pi y,y} & u_{\pi y,z} \\ u_{\pi z,x} & u_{\pi z,y} & u_{\pi z,z} \end{bmatrix} = \begin{bmatrix} \hat{a}_1 & \hat{a}_2 & \hat{a}_3 \\ \hat{a}_4 & \hat{a}_5 & \hat{a}_6 \\ \hat{a}_6 & \hat{a}_8 & \hat{a}_9 \end{bmatrix} = \hat{\mathbf{A}}\,. \tag{3.138}$$

With the assumption of flat faces the outward normal $\boldsymbol{N}$ is constant at each face and thus we arrive with the normal $\boldsymbol{N}_f$ related to face Γ_f at

$$\nabla \boldsymbol{u}_\pi = \frac{1}{\Omega_v} \sum_{f=1}^{n_F} \int\limits_{\Gamma_f} \boldsymbol{u}_h \, \mathrm{d}\Gamma \otimes \boldsymbol{N}_f \tag{3.139}$$

which has to be evaluated using (3.134). ■

Equation (3.53) can be applied to determine the remaining parameters (a_1, a_2, a_3) related to the constant part of the ansatz function, see (3.132). This leads with (3.138) and (3.152) after some manipulations to

$$\begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \frac{1}{n_V} \sum_{k=1}^{n_v} (\mathbf{u}_k - \nabla \boldsymbol{u}_\pi \, \mathbf{X}_k) = \frac{1}{n_V} \sum_{k=1}^{n_v} \Big(\mathbf{u}_k - [\mathbb{P}_\nabla^{(3,1)} \mathbf{u}_v] \, \mathbf{X}_k \Big) \tag{3.140}$$

where $\mathbf{X}_k$ are the initial coordinates of the nodal point k and the sum includes all n_V vertices of the polygon.

In the literature exist exact evaluations and several approximations for the computation of the surface integral in (3.136) and (3.139). Appropriate strategies that are straight forward can be employed for the integration on the faces. These are:

- **Integration over the edges of a face.** An exact integration of the right hand side of (3.138) can be performed directly on the edges. The general idea is pointed out in Fig. 3.10 where the surface Γ_v of the virtual element Ω_v can be split in n_F faces Γ_f with $\Gamma_v = \bigcup_{f=1}^{n_F} \Gamma_f$. The volume integral is then transformed via the divergence theorem to a surface integral. For each of the surfaces a local coordinate system $\{\mathsf{e}_X^f, \mathsf{e}_Y^f\}$ can be defined and then by employing Greens theorem the surface integral

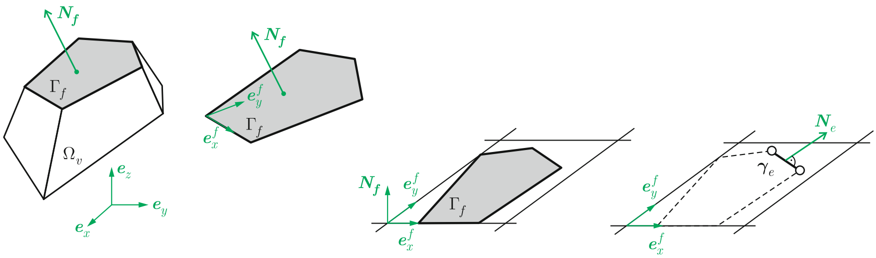


Fig. 3.10 From volume via face to edge integral

can be evaluated at the edge $\gamma_f = \bigcup_{e=1}^{n_e} \gamma_e$ of the surface Γ_f, by computing each contribution of the straight parts γ_e of γ_f to the integral.
For such formulation first the divergence theorem is applied on an arbitrary element face $\Gamma_f \in \mathbb{R}^3$

$$\begin{aligned}\int\limits_{\Gamma_f} \phi \, d\Gamma &= \frac{1}{2} \int\limits_{\gamma_f} \begin{bmatrix} \int \phi dX^f \\ \int \phi dY^f \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{e}_X^{f\,T} \\ \boldsymbol{e}_Y^{f\,T} \end{bmatrix} \boldsymbol{N}_e \, d\gamma \\ &= \frac{1}{2} \int\limits_{\gamma_f} \left(\int \phi dX^f \, \boldsymbol{e}_X^{f\,T} \boldsymbol{N}_e + \int \phi dY^f \, \boldsymbol{e}_Y^{f\,T} \boldsymbol{N}_e \right) d\gamma \end{aligned} \tag{3.141}$$

The integrand $\phi(X, Y, Z) \leftarrow \phi^f(X^f, Y^f)$ has to be expressed by the local coordinate system of a face $\boldsymbol{X}^f = (X^f, Y^f) = \mathbf{R}^f(\boldsymbol{X} - \mathbf{X}_m)$ with $\mathbf{R}^f = \{\boldsymbol{e}_X^f, \boldsymbol{e}_Y^f\}^T$ being a (2×3) transformation matrix. The local coordinate system is defined by the base vectors, see Fig. 3.11

$$\boldsymbol{e}_X^f = \frac{\mathbf{X}_2 - \mathbf{X}_1}{\|\mathbf{X}_2 - \mathbf{X}_1\|}, \quad \boldsymbol{e}_Y^f = \boldsymbol{N}_f \times \boldsymbol{e}_X^f \tag{3.142}$$

where $\mathbf{N}_f$ is the unit normal of the face that follows from

$$\boldsymbol{n}_f = (\mathbf{X}_1 - \mathbf{X}_m) \times (\mathbf{X}_2 - \mathbf{X}_m), \Longrightarrow \boldsymbol{N}_f = \frac{\boldsymbol{n}_f}{\|\boldsymbol{n}_f\|} \tag{3.143}$$

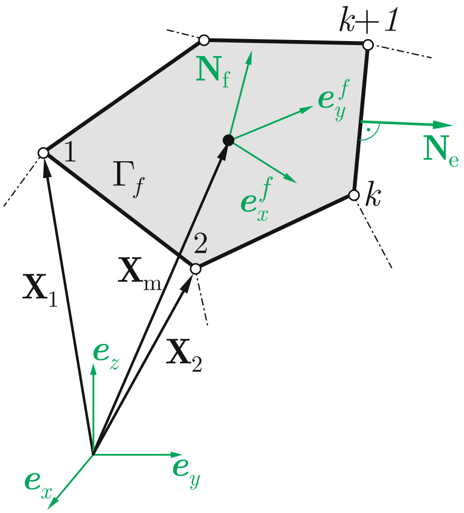


Fig. 3.11 Virtual element face Γ_f and local coordinate system

where $\mathbf{X}_m$ is the center of the face given by the sum over all n_V^f vertices attached to the face: $\mathbf{X}_m = \frac{1}{n_V^f} \sum_{I=1}^{n_V^f} \mathbf{X}_I$. Furthermore the normal to the edge γ_e is given by

$$\boldsymbol{N}_e = \frac{\mathbf{X}_{k+1} - \mathbf{X}_k}{\|\mathbf{X}_{k+1} - \mathbf{X}_k\|} \times \boldsymbol{N}_f \tag{3.144}$$

Due to the definition of the virtual element method the faces have to be planar. Hence the normal $\boldsymbol{N}_f$ is constant within the face leading to the projection in (3.139). The integral over all n_F faces can be evaluated over the edges. For one face Γ_f in the sum we obtain using (3.141)

$$\int\limits_{\Gamma_f} \boldsymbol{u}_h \, d\Gamma = \frac{1}{2} \sum_{e=1}^{n_E} \int\limits_{\gamma_e} \begin{bmatrix} \int \boldsymbol{u}_h \, dX^f \\ \int \boldsymbol{u}_h \, dY^f \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{e}_X^f \\ \boldsymbol{e}_Y^f \end{bmatrix} \boldsymbol{N}_e \, d\gamma \tag{3.145}$$

The problem is now that we do not know $\boldsymbol{u}_h$. A way out is to approximate $\boldsymbol{u}_h$ by $\boldsymbol{u}_\pi$ in $\int_{\Gamma_f} \boldsymbol{u}_h \, d\Gamma = \int_{\Gamma_f} \boldsymbol{u}_\pi \, d\Gamma$, see (3.37), which follows from the equivalence of the functions for ($n = 1$), see Ahmad et al. (2013)

$$\int\limits_{\Gamma_f} \boldsymbol{u}_h \, d\Gamma = \frac{1}{2} \sum_{e=1}^{n_E} \int\limits_{\gamma_e} \left(\int \boldsymbol{u}_\pi^f \, dX^f \, (\boldsymbol{e}_X^f)^T \boldsymbol{N}_e + \int \boldsymbol{u}_\pi^f \, dY^f \, (\boldsymbol{e}_Y^f)^T \boldsymbol{N}_e \right) d\gamma \, . \tag{3.146}$$

The displacements $\boldsymbol{u}_\pi^f$ are related to the local coordinate system (X^f, Y^f) attached to the face Γ_f. The projection follows now which follows from

$$\begin{aligned} \begin{bmatrix} a_{12}^f & a_{13}^f \\ a_{22}^f & a_{23}^f \\ a_{32}^f & a_{33}^f \end{bmatrix} &= \nabla \boldsymbol{u}_\pi^f = \frac{1}{\Gamma_f} \int\limits_{\Gamma_f} \mathbf{R}^f \, \nabla \boldsymbol{u}_h^f \, d\Gamma = \frac{1}{\Gamma_f} \int\limits_{\gamma_f} \boldsymbol{u}_h^f \otimes \mathbf{R}^f \, \mathsf{N}_\gamma \, d\gamma \\ &= \frac{1}{\Gamma_f} \sum_{e=1}^{n_E} \int\limits_{\gamma_e} \boldsymbol{u}_h^f \, d\gamma \, (\mathbf{R}^f \boldsymbol{N}_e)^T = \frac{1}{\Gamma_f} \sum_{e=1}^{n_E} \frac{l_e}{2} (\mathbf{u}_k + \mathbf{u}_{k+1}) \, (\mathbf{R}^f \boldsymbol{N}_e)^T \end{aligned} \tag{3.147}$$

where $\mathbf{u}_k$ and $\mathbf{u}_{k+1}$ are the nodal values at the edge e with length l_e, see Fig. 3.11. Note that the trapezoidal rule yields an exact integration for the linear ansatz of $\boldsymbol{u}_h^f$ at the edge.[9]

Additionally equation (3.140) provides the missing constants a_{1i}^f

[9] The transformation of the volume to line integrals can be performed in a different way, see Mirtich (1996). It depends on the specific geometry of the virtual element which way is more efficient.

$$\begin{Bmatrix} a^f_{1\,1} \\ a^f_{1\,2} \\ a^f_{1\,3} \end{Bmatrix} = \frac{1}{n^f_V} \sum_{I=1}^{n^f_V} \left(\mathbf{u}_I - \nabla \mathsf{u}^f_\pi \, \mathbf{X}^f_I \right) = \frac{1}{n^f_V} \sum_{I=1}^{n^f_V} \mathbf{u}_I \,. \tag{3.148}$$

Due to the choice of the origin $\mathbf{X}_m$ of the local coordinate system the second term in (3.148) vanishes: $\sum_{I=1}^{n^f_V} \mathbf{X}^f_I = \mathbf{0}$. The projected quantity u^f_π is then given at any point in the face Γ_f by the ansatz

$$\begin{aligned} \boldsymbol{u}^f_\pi &= \left(\mathbf{N}^f_\pi \, \mathbf{a}^f_i \right) \boldsymbol{e}_i \quad \text{with} \\ \mathbf{N}^f_\pi &= \langle 1 \; X^f \; Y^f \, \rangle \quad \text{and} \quad \mathbf{a}^f_i = \langle a^f_{i\,1} \, a^f_{i\,2} \, a^f_{i\,3} \rangle^T \end{aligned} \tag{3.149}$$

in terms of the nodal unknowns $\mathbf{u}_v$, see (3.147) and (3.148). The ansatz in (3.149) can now be applied in (3.146) to compute the integral which finally leads to the projection (3.139).

- **Split into quadrilaterals.** A simple integration scheme is presented in Gain et al. (2014). In this work the faces are split into quadrilaterals where one corner lies at the face node, two corners lie in center of adjacent edges and the fourth corner lies in face centroid. Integration is then performed over the quadrilaterals.
- **Gauss integration using triangles.** A further alternative is to split the polygonal faces Γ_f, see Fig. 3.12, into triangles τ_i and to integrate over each triangle by using standard linear ansatz function and Gauss integration. The ansatz for each triangle is given by

$$\mathbf{N}_\tau = \{N_1, N_2, N_3\} = \{\xi, \eta, 1 - \xi - \eta\} \tag{3.150}$$

$$\boldsymbol{u}^\tau_h = \sum_{I=1}^{3} N_I(\xi, \eta) \, \mathbf{u}_I . \tag{3.151}$$

Here $\boldsymbol{u}^\tau_h$ is the linear approximation of the displacements at each triangle $\mathcal{T}$ within a polygonal face f. In (3.150) ξ and η are the local dimensionless coordinates and $(\mathbf{u}_I)_{\forall I \in \mathcal{T}}$ is a list containing the three nodal displacement vectors, one at each node of the triangle $\mathcal{T}$. The local nodes of $\mathcal{T}$ are vertices of the face under consideration as depicted in Fig. 3.12.

The right hand side of (3.60) yields for n_τ triangles describing a polygonal face $\Gamma_f = \cup_{\tau=1}^{n_\tau} \Gamma_\tau$ of the virtual element

$$\begin{aligned} \nabla \boldsymbol{u}_\pi &= \frac{1}{\Omega_v} \int\limits_{\Gamma_v} \boldsymbol{u}^\tau_h \otimes \mathbf{N} \, \mathrm{d}\Gamma = \frac{1}{\Omega_v} \sum_{f=1}^{n_F} \sum_{\tau=1}^{n_\tau} \int\limits_{\Gamma_\tau} \boldsymbol{u}^\tau_h \otimes \mathbf{N}_f \, \mathrm{d}\Gamma \\ &= \frac{1}{\Omega_v} \sum_{f=1}^{n_F} \sum_{\tau=1}^{n_\tau} \sum_{g=1}^{n_g} w_g \, \boldsymbol{u}^\tau_h(\boldsymbol{\xi}_g) \, \mathbf{N}^T_f = \mathbb{P}^{(3,1)}_\nabla \mathbf{u}_v \,. \end{aligned} \tag{3.152}$$

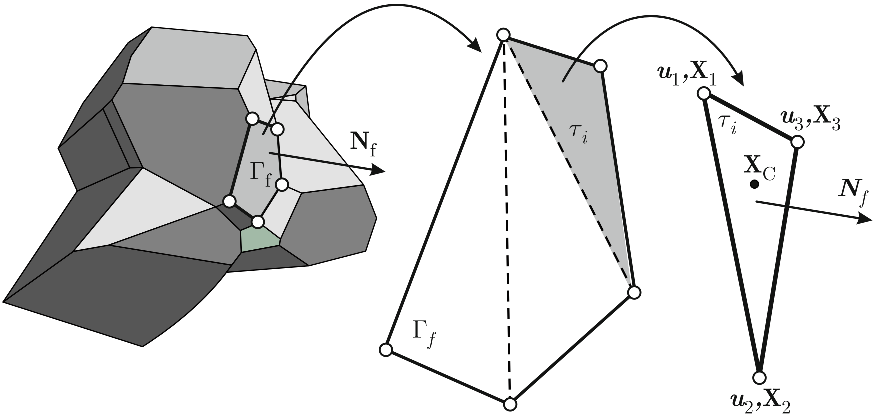


Fig. 3.12 Virtual element faces split into multiple triangles

For the chosen linear shape function (3.150) a one point quadrature rule $n_g = 1$ is sufficient with Gauss point weight $w_g = 1/2$. All quantities are evaluated at the Gauss point $\boldsymbol{\xi}_g = (\xi_g, \eta_g)$ with $\xi_g = 1/3$ and $\eta_g = 1/3$. The normal vector $\boldsymbol{N}_f$ is computed using the convective base vectors $\boldsymbol{G}_\xi$ and $\boldsymbol{G}_\eta$ related to the surface of the triangle

$$\boldsymbol{X}^\tau = \sum_{I=1}^{3} N_I(\xi, \eta)\, \mathbf{X}_I , \tag{3.153}$$

$$G_\xi = \boldsymbol{X}^\tau_{,\xi}, \;\; \boldsymbol{G}_\eta = \boldsymbol{X}^\tau_{,\eta}, \; \mathbf{N}_k = \boldsymbol{G}_\xi \times \boldsymbol{G}_\eta \,. \tag{3.154}$$

The vector $\boldsymbol{N}_k$ needs not to be normalized since the term $\|\boldsymbol{G}_\xi \times \boldsymbol{G}_\eta\|$ cancels out when the second integral in (3.152) is transformed to the reference coordinates ξ and η within a triangle $\mathcal{T}$. This yields a more efficient formulation. All quantities are related to the initial configuration. By comparing (3.133) and the right hand side of (3.152) the unknown parameters $\hat{a}_i$ $(i = 1, 2, \ldots, 9)$ follow by inspection as in the two-dimensional case.

With equations (3.152) and (3.132) the ansatz function $\boldsymbol{u}_\pi$ of the virtual element can be completely defined in terms of the nodal displacements $\mathbf{u}_v$ that are given by

$$\mathbf{u}_v = \left\langle u_{x_1}\, u_{y_1}\, u_{z_1}\, u_{x_2}\, u_{y_2} \,\ldots\, u_{y_{n_V}}\, u_{z_{n_V}} \right\rangle^T = \left\langle \mathbf{u}_1\, \mathbf{u}_2 \,\ldots\, \mathbf{u}_{n_V} \right\rangle^T . \tag{3.155}$$

Again we can use the short notation to write

$$\mathbf{a} = \mathbb{P}_u^{(3,1)}\, \mathbf{u}_v \tag{3.156}$$

where $\mathbf{a}$ are the parameters related to the virtual element ansatz, $\mathbb{P}_u^{(3,1)}$ is a projection function which relates the vector $\mathbf{u}_v$, containing the nodal degrees of freedom of all vertices of the virtual element Ω_v, to the parameters $\mathbf{a}$. This projection matrix can be constructed using (3.140) and (3.133). With that the projected displacement $\boldsymbol{u}_\pi$ in the virtual element Ω_v can be recast, using (3.132), as

$$\boldsymbol{u}_\pi = \mathbf{H}_u^{(3,1)}(X, Y, Z)\, \mathbb{P}_u^{(3,1)}\, \mathbf{u}_v\,. \tag{3.157}$$

References

Ahmad, B., A. Alsaedi, F. Brezzi, L. Marini, and A. Russo. 2013. Equivalent projectors for virtual element methods. *Computers and Mathematics with Applications* 66: 376–391.

Aldakheel, F., B. Hudobivnik, A. Hussein, and P. Wriggers. 2018. Phase-field modeling of brittle fracture using an efficient virtual element scheme. *Computer Methods in Applied Mechanics and Engineering* 341: 443–466.

Aldakheel, F., B. Hudobivnik, and P. Wriggers. 2019. Virtual elements for finite thermo-plasticity problems. *Computational Mechanics* 64: 1347–1360.

Aldakheel, F., B. Hudobivnik, L. Beirão da Veiga, and P. Wriggers. 2020. Curvilinear virtual elements for contact mechanics. *Computer Methods in Applied Mechanics and Engineering* 372: 113394.

Antonietti, P.F., L.B. da Veiga, S. Scacchi, and M. Verani. 2016. A C^1 virtual element method for the Cahn-Hilliard equation with polygonal meshes. *SIAM Journal on Numerical Analysis* 54 (1): 34–56.

Artioli, E., L. Beirão da Veiga, and E. Sacco. 2017a. Arbitrary order 2d virtual elements for polygonal meshes: Part I, elastic problem. *Computational Mechanics* 60: 355–377.

Artioli, E., L. Beirão da Veiga, and E. Sacco. 2017b. Arbitrary order 2d virtual elements for polygonal meshes: Part II, inelastic problem. *Computational Mechanics* 60: 643–657.

Artioli, E., L. Beirão da Veiga, and F. Dassi. 2020a. Curvilinear virtual elements for 2d solid mechanics applications. *Computer Methods in Applied Mechanics and Engineering* 359: 112667.

Artioli, E., L. Beirão da Veiga, and M. Verani. 2020b. An adaptive curved virtual element method for the statistical homogenization of random fibre-reinforced composites. *Finite Elements in Analysis and Design* 177: 103418.

Beirão da Veiga, L., F. Brezzi, A. Cangiani, G. Manzini, L. Marini, and A. Russo. 2013a. Basic principles of virtual element methods. *Mathematical Models and Methods in Applied Sciences* 23 (01): 199–214.

Beirão da Veiga, L., F. Brezzi, and L. Marini. 2013b. Virtual elements for linear elasticity problems. *SIAM, Journal of Numerical Analysis* 51: 794–812.

Beirão da Veiga, L., F. Brezzi, L.D. Marini, and A. Russo. 2014. The hitchhiker's guide to the virtual element method. *Mathematical Models and Methods in Applied Sciences* 24 (8): 1541–1573.

Beirão da Veiga, L., C. Lovadina, and D. Mora. 2015. A virtual element method for elastic and inelastic problems on polytope meshes. *Computer Methods in Applied Mechanics and Engineering* 295: 327–346.

Beirão da Veiga, L., F. Brezzi, L. Marini, and A. Russo. 2016. Serendipity nodal VEM spaces. *Computers and Fluids* 141: 2–12.

Beirão da Veiga, L., F. Brezzi, M.L. D, and A. R. 2017. Serendipity face and edge VEM spaces. *Atti Accad. Naz. Lincei Cl. Sci. Fis. Mat. Natur* 28: 143–180.

Beirão da Veiga, L., F. Brezzi, and F. Dassi. 2018. Serendipity virtual elements for general elliptic equations in three dimensions. *Chinese Annals of Mathematics, Series B* 39 (2): 315–334.

Beirão da Veiga, L., F. Brezzi, F. Dassi, L.D. Marini, and A. Russo. 2018. A family of three-dimensional virtual elements with applications to magnetostatics. *SIAM Journal on Numerical Analysis* 56 (5): 2940–2962.

Beirão da Veiga L., F. Brezzi, L.D. Marini, A. Russo. 2019a. Virtual elements and curved edges. arXiv:1910.10184.

Beirão da Veiga, L., G. Manzini, and L. Mascotto. 2019b. A posteriori error estimation and adaptivity in hp virtual elements. *Numerische Mathematik* 143 (1): 139–175.

Beirão da Veiga, L., D. Mora, and G. Rivera. 2019c. Virtual elements for a shear-deflection formulation of Reissner–Mindlin plates. *Mathematics of Computation* 88 (315): 149–178.

Beirão da Veiga, L., A. Russo, and G. Vacca. 2019d. The virtual element method with curved edges. *ESAIM: Mathematical Modelling and Numerical Analysis* 53 (2): 375–404.

Beirão da Veiga, L., F. Brezzi, L. Marini, and A. Russo. 2020. Polynomial preserving virtual elements with curved edges. *Mathematical Models and Methods in Applied Sciences* 30 (08): 1555–1590.

Bernardini, F. 1999. Integration of polynomials over polyhedra. *Computer-Aided Design* 23: 51–58.

Bertoluzza, S., M. Pennacchio, and D. Prada. 2019. High order vem on curved domains. *Rendiconti Lincei* 30 (2): 391–412.

Brezzi, F., and L.D. Marini. 2013. Virtual element methods for plate bending problems. *Computer Methods in Applied Mechanics and Engineering* 253: 455–462.

Chin, E.B., and N. Sukumar. 2020. An efficient method to integrate polynomials over polytopes and curved solids. *Computer Aided Geometric Design* 82: 101914.

Chinosi, C., and L.D. Marini. 2016. Virtual element method for fourth order problems: L2-estimates. *Computers & Mathematics with Applications* 72 (8): 1959–1967.

Dassi, F., and L. Mascotto. 2018. Exploring high-order three dimensional virtual elements: bases and stabilizations. *Computers and Mathematics with Applications* 75 (9): 3379–3401.

Dassi, F., A. Fumagalli, D. Losapio, S. Scialò, A. Scotti, and G. Vacca. 2021. The mixed virtual element method on curved edges in two dimensions. *Computer Methods in Applied Mechanics and Engineering* 386: 114098.

De Bellis, M., P. Wriggers, B. Hudobivnik, and G. Zavarise. 2018. Virtual element formulation for isotropic damage. *Finite Element Analysis and Design* 144: 38–48.

De Bellis, M., P. Wriggers, and B. Hudobivnik. 2019. Serendipity virtual element formulation for nonlinear elasticity. *Computers & Structures* 223: 106094.

De Loera, J.A., B. Dutra, M. Köppe, S. Moreini, G. Pinto, and J. Wu. 2013. Software for exact integration of polynomials over polyhedra. *Computational Geometry* 46: 232–252.

Düster, A., H. Bröker, and E. Rank. 2001. The p-version of the finite element method for three-dimensional curved thin walled structures. *International Journal for Numerical Methods in Engineering* 52: 673–703.

Düster, A., S. Hartmann, and E. Rank. 2003. p-FEM applied to finite isotropic hyperelastic bodies. *Computer Methods in Applied Mechanics and Engineering* 192: 5147–5166.

Gain, A.L., C. Talischi, and G.H. Paulino. 2014. On the virtual element method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes. *Computer Methods in Applied Mechanics and Engineering* 282: 132–160.

Hudobivnik, B., F. Aldakheel, and P. Wriggers. 2018. Low order 3d virtual element formulation for finite elasto-plastic deformations. *Computational Mechanics* 63: 253–269.

Hussein, A., F. Aldakheel, B. Hudobivnik, P. Wriggers, P.A. Guidault, and O. Allix. 2019. A computational framework for brittle crack propagation based on an efficient virtual element method. *Finite Elements in Analysis and Design* 159: 15–32.

Korelc, J., and P. Wriggers. 2016. *Automation of finite element methods*. Berlin: Springer.

Mirtich, B. 1996. Fast and accurate computation of polyhedral mass properties. *Journal of Graphics Tools* 1 (2): 31–50.

Oñate, E. 2009. *Structural analysis with the finite element method, Vol. 1: basis and solids*. Springer.

Wriggers, P. 2008. *Nonlinear finite elements*. Berlin, Heidelberg, New York: Springer.

Wriggers, P., and B. Hudobivnik. 2017. A low order virtual element formulation for finite elasto-plastic deformations. *Computer Methods in Applied Mechanics and Engineering* 327: 459–477.

Wriggers, P., W. Rust, and B. Reddy. 2016. A virtual element method for contact. *Computational Mechanics* 58: 1039–1050.

Wriggers, P., B. Reddy, W. Rust, and B. Hudobivnik. 2017. Efficient virtual element formulations for compressible and incompressible finite deformations. *Computational Mechanics* 60: 253–268.

Wriggers, P., B. Hudobivnik, and F. Aldakheel. 2020. A virtual element formulation for general element shapes. *Computational Mechanics* 66: 963–977.

Wriggers, P., B. Hudobivnik, and F. Aldakheel. 2021. NURBS-based geometries: A mapping approach for virtual serendipity elements. *Computer Methods in Applied Mechanics and Engineering* 378: 113732.

Wriggers, P., B. Hudobivnik, and O. Allix. 2022. On two simple virtual Kirchhoff-Love plate elements for isotropic and anisotropic materials. *Computational Mechanics* 69: 615–637.

Chapter 4
VEM Ansatz Functions and Projection for the Poisson Equation

The formulation of virtual elements for heat conduction problems, see (2.42), or more general for the Poisson equation ($-\Delta\,\theta = f$) was one of the starting points of this method, see Beirão da Veiga et al. (2013), Beirão da Veiga et al. (2014). Since Laplace, Poisson and the elasticity equations are of elliptical nature, the ansatz functions defined in the previous chapter can be employed for these scalar valued problems as well.

4.1 Two-Dimensional Case

The definition of the ansatz space for the virtual element method in Sect. 3.1.1 holds directly for the case of a scalar field, like the temperature θ in the heat conduction Eq. (2.42) and the associated weak form (2.93). Thus the projection in dimension $d = 2$ onto a polynomial space of order n can be based on the ansatz

$$\theta_\pi = \left\langle 1\; X\; Y\; X^2\; XY\; Y^2\; \ldots\; XY^{n-1}\; Y^n \right\rangle \mathbf{a}_\theta = \mathbf{H}_\theta^{(2,n)}(X,Y)\,\mathbf{a}_\theta \tag{4.1}$$

with the constants $\mathbf{a}_\theta^T = \langle a_1\, a_2\, a_3\, a_4\, \ldots\, a_{n_V}\rangle$, where n_V is the number of vertices of the virtual element Ω_v. This ansatz has to be completed for ansatz functions with order $n > 1$ by the moments which are scalar values

$$n = 2: \qquad m_{(0,0)} = m_1 = \frac{1}{\Omega_v}\int\limits_{\Omega_v} \theta_h \, \mathrm{d}\Omega \tag{4.2}$$

$$n = 3: \quad m_{(1,0)} = m_2 = \frac{1}{\Omega_v}\int\limits_{\Omega_v} X\,\theta_h \, \mathrm{d}\Omega \qquad m_{(0,1)} = m_3 = \frac{1}{\Omega_v}\int\limits_{\Omega_v} Y\,\theta_h \, \mathrm{d}\Omega \tag{4.3}$$

and so forth for higher polynomial ansatz orders $n > 3$.

P. Wriggers et al., *Virtual Element Methods in Engineering Sciences*,
https://doi.org/10.1007/978-3-031-39255-9_4

The gradient projector for the temperature is generally given by

$$\nabla\theta_\pi = \begin{Bmatrix} \theta_{\pi,x} \\ \theta_{\pi,y} \end{Bmatrix} = \mathbf{B}^{(2,n)}_{\theta\,\pi}(X,Y)\,\hat{\mathbf{a}}_\theta\,. \tag{4.4}$$

where $\hat{\mathbf{a}}_\theta$ relates to the unknown parameters $a_2\,,a_3\,,a_4\,,\ldots,a_{n_V}$.

4.1.1 Computation of the Projection

The Galerkin projection defined in (3.19) will now be computed to link the unknown parameters $\mathbf{a}_\theta$ to the nodal degrees of freedom θ_i at the vertices of the virtual element with $i = 1\,,\ldots,n_V$.

For the projection based on a scalar field θ all equations from Sect. 3.1.2 hold in a similar fashion. Thus from (3.19) follows

$$\int\limits_{\Omega_v} \nabla N_\pi \cdot \nabla\theta_\pi \;\mathrm{d}\Omega = \int\limits_{\Omega_v} \nabla N_\pi \cdot \nabla\theta_h \;\mathrm{d}\Omega \tag{4.5}$$

and the right hand side is

$$\int\limits_{\Omega_v} \nabla N_\pi \cdot \nabla\theta_h \;\mathrm{d}\Omega = \underbrace{\int\limits_{\Gamma_v} \nabla N_\pi \cdot (\theta_h\,\boldsymbol{N})\;\mathrm{d}\Gamma}_{b_\Gamma} - \underbrace{\int\limits_{\Omega_v} \mathrm{div}[\nabla N_\pi]\,\theta_h\;\mathrm{d}\Omega}_{b_{div}}\,. \tag{4.6}$$

This leads with the ansatz (4.4) to the matrix

$$\mathbf{G}^{(n)}_{\theta\,\pi} = \int\limits_{\Omega_v} \left[\mathbf{B}^{(2,n)}_{\theta\,\pi}(X,Y)\right]^T\,\mathbf{B}^{(2,n)}_{\theta\,\pi}(X,Y)\;\mathrm{d}\Omega \tag{4.7}$$

and with the ansatz along the edges $M_i(\xi)$, see (3.5) and (3.6), to

$$\mathbf{b}^{(n)}_\pi(\theta_i\,,m_j) = \sum_{e=1}^{n_E}\int\limits_{\Gamma_e}\left[\mathbf{B}^{(2,n)}_{\theta\,\pi}(X,Y)\right]^T\sum_{i=1}^{n+1} M_i(\xi)\,\theta_i \begin{Bmatrix} N_{xe} \\ N_{ye} \end{Bmatrix}\mathrm{d}\Gamma + \sum_{j=1}^{n_m}[\mathbf{B}^{j}_{\theta\,\pi\,div}]^T\,m_j\,. \tag{4.8}$$

The last two relations enter the equation system already defined in (3.27) and yield

$$\nabla\theta_\pi = \mathbf{B}^{(2,n)}_{\theta\,\pi}(X,Y)\,\hat{\mathbf{a}} = \mathbf{B}^{(2,n)}_{\theta\,\pi}(X,Y)\,[\mathbf{G}^{(n)}_{\theta\,\pi}]^{-1}\,\mathbf{b}^{(n)}_\pi(\theta_i\,,m_j) \tag{4.9}$$

4.1.2 Projection for a Linear Ansatz

In case of a linear ansatz ($n = 1$) element nodes are placed only at the vertices of the polygonal elements, see Fig. 3.4. One can use the result in (4.9) to construct an approximation of the temperature field gradient in Ω_v by introducing an approximate linear displacement field.

For the ansatz based on a scalar field, see (4.4), and following the steps in Sect. 3.1.4 Eq. (3.46) simplifies with (4.9) to

$$\nabla\theta_\pi = \begin{Bmatrix} \theta_{\pi,x} \\ \theta_{\pi,y} \end{Bmatrix} = \mathbf{B}^{(2,1)}_{\theta\,\pi}(X, Y)\,\hat{\mathbf{a}} = \frac{1}{2\Omega_v}\sum_{e=1}^{n_E}(\theta_k + \theta_{k+1}) \begin{Bmatrix} (Y_1 - Y_2)_e \\ (X_2 - X_1)_e \end{Bmatrix} \tag{4.10}$$

where the last term relates to the normal vector at the edge and θ_k are the nodal degrees of freedom. Again, no equation system has to be solved in for the linear ansatz. Furthermore, by introducing the vector of nodal unknowns of the virtual element Ω_v

$$\boldsymbol{\theta}_v = \langle \theta_1\, \theta_2\, \dots\, \theta_{n_V} \rangle^T \tag{4.11}$$

the projection tensor, provided in (3.49), can be used adequately with the same definitions as in (3.49) for the components P_{xk} and P_{yk}. This leads to

$$\mathbb{P}^{(2,1)}_{\nabla\theta} = \frac{1}{2\Omega_v} \begin{bmatrix} P_{x1} & P_{x2} & \dots & P_{xn_V} \\ P_{y1} & P_{y2} & \dots & P_{yn_V} \end{bmatrix} = \begin{bmatrix} \mathbf{L}^{h\,T}_x \\ \mathbf{L}^{h\,T}_y \end{bmatrix} \tag{4.12}$$

Now a direct connection between the projected gradient and the nodel degrees of freedom $\boldsymbol{\theta}_v$ of the virtual element can be formulated

$$\nabla\theta_\pi = \mathbb{P}^{(2,1)}_{\nabla\theta}\,\boldsymbol{\theta}_v\,. \tag{4.13}$$

With the ansatz for the temperature

$$\theta_\pi = \mathbf{H}^{(2,1)}_\theta(X, Y)\,\mathbf{a} = \langle 1 \;\; X \;\; Y \rangle \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} \tag{4.14}$$

it is possible to compute, as in Sect. 3.1.4, the first constant a_1. The constants $a_2 = \hat{a}_1$ and $a_3 = \hat{a}_2$ are known from (4.12) and thus a_1 follows according to (3.54) as

$$a_1 = \sum_{k=1}^{n_V}[\,\theta_k - \hat{a}_1\, X_k - \hat{a}_2\, Y_k\,]\,/\,n_V \tag{4.15}$$

which can be written in vector form using the definition of $\boldsymbol{\theta}_v$ in (3.48) and (3.55) as

$$a_1 = \frac{1}{n_V}\left[\mathbf{1}^T - X_S\,\mathbf{L}_x^{h\,T} - Y_S\,\mathbf{L}_y^{h\,T}\right]\boldsymbol{\theta}_v \tag{4.16}$$

with the vector $\mathbf{1}^T = \{1, 1, \ldots, 1\}$ of length n_V. The approximation θ_π can now be obtained by combining the last equation with (4.12) leading to

$$\theta_\pi = \mathbf{H}_\theta^{(2,1)}(X, Y)\,\mathbb{P}_\theta^{(2,1)}\boldsymbol{\theta}_v \quad \text{with } \mathbb{P}_\theta^{(2,1)} = \begin{bmatrix} (\mathbf{1}^T - X_S\,\mathbf{L}_x^{h\,T} - Y_S\,\mathbf{L}_y^{h\,T})/n_V \\ \mathbf{L}_x^{h\,T} \\ \mathbf{L}_y^{h\,T} \end{bmatrix}. \tag{4.17}$$

with $\mathbb{P}_\theta^{(2,1)}$ being a $(3 \times n_V)$ matrix.

4.1.3 Projection for a Quadratic Ansatz

The Poisson equation has a scalar valued function as unknown. Here we refer to the temperature θ that appears in the heat conduction equation. Also for the quadratic ansatz we can explore the analogy to elasticity and write the projected gradient as

$$\nabla\theta_\pi = \begin{Bmatrix} \theta_{,x} \\ \theta_{,y} \end{Bmatrix} = \mathbf{B}^{(2,2)}{}_{\theta\,\pi}(X, Y)\,\hat{\mathbf{a}} = \begin{bmatrix} 1 & 0 & 2X & Y & 0 \\ 0 & 1 & 0 & X & 2Y \end{bmatrix} \begin{Bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \ldots \\ \hat{a}_5 \end{Bmatrix}. \tag{4.18}$$

Again the ansatz (3.64) at the straight edge Γ_e is introduced and the outward normal vector related to the edge is denoted by $\mathbf{N}_e$.

With this ansatz the matrices $\mathbf{G}_{\theta\,\pi}^{(2)}$ and $\mathbf{b}_\pi^{(2)}(\theta_i\,, m_1)$ can be computed. The matrix $\mathbf{G}_{\theta\,\pi}^{(2)}$ follows directly from (3.65). The only difference is that $\mathbf{G}_{\theta\,\pi}^{(2)}$ has only the size of (5×5). Again, the integral defining $\mathbf{G}_{\theta\,\pi}^{(2)}$ contains only polynomials up to second order and can be evaluated exactly.

The vector $\mathbf{b}_\pi^h$ follows from (3.66) and can be expressed with (4.2) as

$$\mathbf{b}_\pi^{(2)}(\theta_i\,, m_1) = \sum_{e=1}^{n_E} l_e \sum_{i=1}^{3} w_i [\mathbf{B}_{\theta\,\pi}^{(2,2)}(X, Y)]^T \theta_i \begin{Bmatrix} N_{xe} \\ N_{ye} \end{Bmatrix} + [\mathbf{B}_{\theta\,\pi\,div}^1]^T\, m_1 \tag{4.19}$$

with the weighting factors of the Gauss-Lobatto rule: $w_1 = w_2 = \frac{1}{6}$ and $w_3 = \frac{2}{3}$.

The second term in (4.19) can be obtained from (4.6) as $N_{\pi,xx} + N_{\pi,yy}$ which is constant in the quadratic case and leads to

$$\mathbf{B}_{\theta\,\pi\,div}^1 = 2\left\langle 0\;0\;\Omega_v\;0\;\Omega_v \right\rangle. \tag{4.20}$$

With the explicit forms of $\mathbf{G}_{\theta\,\pi}^{(2)}$ and $\mathbf{b}_\pi^h$ in (4.19) the equations system (3.27) can be solved for the unknown parameters $\hat{\mathbf{a}}$

$$\nabla \theta_\pi = \mathbf{B}^{(2,2)}_{\theta\,\pi}(X, Y)\,\hat{\mathbf{a}} = \mathbf{B}^{(2,2)}_{\theta\,\pi}(X, Y)\,[\mathbf{G}^{(2)}_{\theta\,\pi}]^{-1}\,\mathbf{b}^{(2)}_{\pi}(\theta_i\,, m_1) \tag{4.21}$$

An explicit dependency of the projected gradient on the unknown vector of the virtual element Ω_v

$$\boldsymbol{\theta}_v = \langle \theta_1\,\theta_2\,\ldots\,\theta_{nV}\,m_1\rangle^T$$

can be constructed easily in the same manner as in (3.75). With normal vector at the edge Γ_e

$$\bar{\mathbf{N}}^\theta_e = \begin{Bmatrix} (Y_1 - Y_2)_e \\ (X_2 - X_1)_e \end{Bmatrix} \tag{4.22}$$

and by applying the Gauss-Lobatto rule to integrate (4.19)

$$\begin{aligned} \mathbf{P}^{\theta\,T}_{2k-1} &= \frac{1}{6}\mathbf{B}^{\theta\,T}_{2k-1}(\bar{\mathbf{N}}^\theta_e + \bar{\mathbf{N}}^\theta_{e-1}) \\ \mathbf{P}^{\theta\,T}_{2k} &= \frac{2}{3}\mathbf{B}^{\theta\,T}_{2k}\bar{\mathbf{N}}^\theta_e \\ \mathbf{P}^{\theta\,T}_{2k+1} &= \frac{1}{6}\mathbf{B}^{\theta\,T}_{2k+1}(\bar{\mathbf{N}}^\theta_e + \bar{\mathbf{N}}^\theta_{e+1})\,, \end{aligned}$$

with the special case $\mathbf{P}^{\theta\,T}_1 = \frac{1}{6}\mathbf{B}^{\theta\,T}_1(\bar{\mathbf{N}}^\theta_1 + \bar{\mathbf{N}}^\theta_{n_E})$, this yields the matrix of size ($5 \times 2n_V + 1$)

$$\mathbf{B}^{(2)}_\theta = \left[\mathbf{P}^{h\,T}_1\ \mathbf{P}^{\theta\,T}_2\ \ldots\ \mathbf{P}^{\theta\,T}_{2k-1}\ \mathbf{P}^{\theta\,T}_{2k}\ \mathbf{P}^{\theta\,T}_{2k+1}\ \ldots\mathbf{P}^{\theta\,T}_{2n_V}\ \mathbf{B}^{1\,T}_{\theta\,\pi\,div}\right]. \tag{4.23}$$

Again the projected gradient $\nabla\theta_\pi$ depends on the unknowns $\boldsymbol{\theta}_v$ and is given by

$$\nabla \theta_\pi = \mathbf{B}^{(2,2)}_{\theta\,\pi}(X, Y)\,[\mathbf{G}^{(2)}_{\theta\,\pi}]^{-1}\,\mathbf{B}^{(2)}_\theta\,\boldsymbol{\theta}_v = \mathbf{B}^{(2,2)}_{\theta\,\pi}(X, Y)\,\mathbb{P}^{(2,2)}_{\nabla\theta}\,\boldsymbol{\theta}_v\,. \tag{4.24}$$

where $\mathbb{P}^{(2,2)}_{\nabla\theta} = [\mathbf{G}^{(2)}_{\theta\,\pi}]^{-1}\,\mathbf{B}^{(2)}_\theta$ is a constant matrix of size $5 \times 2n_V + 1$.

As in the solid mechanics case it is possible to introduce the ansatz

$$\theta_\pi = \left\langle 1\,X\,Y\,X^2\,XY\,Y^2\right\rangle \begin{Bmatrix} a_1 \\ a_2 \\ \ldots \\ a_6 \end{Bmatrix} = \mathbf{H}^{(2,2)}_\theta(X, Y)\,\mathbf{a} \tag{4.25}$$

where the constants a_2 to a_6 are already known from the gradient projection. For the determination of a_1 the condition

$$\frac{1}{\Omega_v}\int\limits_{\Omega_v} \theta_\pi \mathrm{d}\Omega = \frac{1}{\Omega_v}\int\limits_{\Omega_v} \theta_h \mathrm{d}\Omega \tag{4.26}$$

is used, see also (3.34). Observe that the right hand side is equivalent to the moment m_1 and by introducing the ansatz (4.25) one obtains

$$\frac{1}{\Omega_v} \int\limits_{\Omega_v} \mathbf{H}_\theta^{(2,2)}(X,Y) \mathrm{d}\Omega \, \mathbf{a} = m_1 \, . \tag{4.27}$$

Splitting now the ansatz (4.25) by taking out the constant term

$$\theta_\pi = a_1 + \hat{\mathbf{H}}_\theta^{(2,2)}(X,Y) \, \hat{\mathbf{a}}$$

it follows

$$a_1 = m_1 - \frac{1}{\Omega_v} \left(\int\limits_{\Omega_v} \hat{\mathbf{H}}_\theta^{(2,2)}(X,Y) \mathrm{d}\Omega \right) \mathbb{P}_{\nabla\theta}^{(2,2)} \, \boldsymbol{\theta}_v \tag{4.28}$$

With the abbreviation $\hat{\mathbf{H}}^{int} = \frac{1}{\Omega_v} \int_{\Omega_v} \hat{\mathbf{H}}_\theta^{(2,2)}(X,Y) \mathrm{d}\Omega$ the projected variable follows as

$$\theta_\pi = \mathbf{H}_\theta^{(2,2)}(X,Y) \, \mathbb{P}_\theta^{(2,2)} \, \boldsymbol{\theta}_v \quad \text{with } \mathbb{P}_\theta^{(2,2)} = \begin{Bmatrix} \mathbf{I}_m - \hat{\mathbf{H}}^{int} \mathbb{P}_{\nabla\theta}^{(2,2)} \\ \mathbb{P}_{\nabla\theta}^{(2,2)} \end{Bmatrix} \tag{4.29}$$

where $\mathbf{I}_m = \langle 0 \, 0 \, \ldots \, 1 \rangle^T$ takes care of m_1 having the size $(1 \times 2n_V + 1)$ and $\mathbb{P}_\theta^{(2,2)}$ is a constant matrix within Ω_v.

4.2 Three-Dimensional Case

The derivation of the ansatz functions in three-dimensions follows along the lines of the two-dimensional case. Compared to three-dimensional solids it is simpler, especially for a linear ansatz which will be discussed next.

The linear ansatz for the temperature

$$\theta_\pi = \mathbf{H}_\theta^{(3,1)}(X,Y,Z) \, \mathbf{a} = \langle 1 \;\; X \;\; Y \;\; Z \rangle \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} \tag{4.30}$$

yields a constant gradient

$$\nabla\theta_\pi = \begin{Bmatrix} \theta_{,x} \\ \theta_{,y} \\ \theta_{,z} \end{Bmatrix} = \begin{Bmatrix} a_2 \\ a_3 \\ a_4 \end{Bmatrix} = \mathbf{B}^{(3,1)}_{\theta\,\pi}\,\hat{\mathbf{a}} \tag{4.31}$$

where in this special case $\mathbf{B}^{(3,1)}_{\theta\,\pi} = \mathbf{1}$. Since the gradient is constant we can use (3.60) in *Remark 3.1* which yields for scalar functions

$$\nabla\theta_\pi = \frac{1}{\Omega_v}\int\limits_{\Omega_v} \nabla\theta_h \ \mathrm{d}\Omega = \frac{1}{\Omega_v}\int\limits_{\Gamma_v} \theta_h\, \boldsymbol{N} \ \mathrm{d}\Gamma \tag{4.32}$$

The integral over the surface Γ_v of the virtual element has to be evaluated over all its faces γ_f

$$\int\limits_{\Gamma_v} \theta_h\, \boldsymbol{N} \ \mathrm{d}\Gamma = \sum_{f=1}^{n_F}\int\limits_{\Gamma_f} \theta_h\, \boldsymbol{N}_f \ \mathrm{d}\Gamma = \sum_{f=1}^{n_F}\int\limits_{\Gamma_f} \theta_\pi \ \mathrm{d}\Gamma\, \boldsymbol{N}_f \tag{4.33}$$

where the normal vector $\boldsymbol{N}_f$ is constant at each face Γ_f. As observed in the case of solids, θ_h is not known within the face, but defined along the edges γ_e of the face. Hence the right hand side is not directly computable. As for solids, the way out is to use the equivalent projector θ^0_π, defined in Sect. 3.1.3, which for the ansatz order $n = 1, 2$ is equivalent to θ_π. This leads to the last term in (4.33).

Alternatively, one can approximate the variable θ_h by a linear function using a triangularization on each face with a linear ansatz, see e.g. (3.151).

Here we use the first approach and transform the temperature θ_π to a local coordinate system (X^f, Y^f). An exact integration of the right hand side of (3.138) can be performed directly on the edges. In this case triangulation of the face Γ_f is not needed and thus eliminates a possible bias with respect to the choice of the triangular mesh. For such formulation the divergence theorem can be applied on an arbitrary element face $\Gamma_f \in \mathbb{R}^3$

$$\int\limits_{\Gamma_f} \theta_\pi\, d\Gamma = \frac{1}{2}\int\limits_{\gamma_f} \begin{bmatrix} \int \theta_\pi dX^f \\ \int \theta_\pi dY^f \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{e}^{f\,T}_X \\ \boldsymbol{e}^{f\,T}_Y \end{bmatrix} \boldsymbol{N}_e\, d\gamma \tag{4.34}$$

The integrand $\theta_\pi(X, Y, Z)$ has to be expressed by the local coordinate system of a face $\boldsymbol{X}^f = (X^f, Y^f) = \mathbf{R}^f(\boldsymbol{X} - \mathbf{X}_m)$ with $\mathbf{R}^f = \{\boldsymbol{e}^f_X, \boldsymbol{e}^f_Y\}^T$ being a (2×3) transformation matrix. The local coordinate system is defined by (3.142), see also Fig. 3.11 in Sect. 3.2.2. The unit normal $\boldsymbol{N}_f$ is related to the face and defined by (3.143) while the normal at the edges is given by (3.144).

Analogous to the projection for the solid the projection related to each face yields

$$\begin{aligned}\begin{Bmatrix} a_2^f \\ a_3^f \end{Bmatrix} = \nabla\theta_\pi^f = \frac{1}{\Gamma_f}\int\limits_{\Gamma_f} \mathbf{R}^r \nabla\theta_h^f d\Gamma = \frac{1}{\Gamma_f}\int\limits_{\gamma_f} \theta_h^f\, \mathbf{R}^f \boldsymbol{N}_\gamma \\ = \frac{1}{\Gamma_f}\sum_{e=1}^{n_E}\int\limits_{\gamma_e} \theta_h^f\, d\gamma\, \mathbf{R}^f \boldsymbol{N}_e = \frac{1}{\Gamma_f}\sum_{e=1}^{n_E}\frac{l_e}{2}(\theta_k + \theta_{k+1})\, \mathbf{R}^f \boldsymbol{N}_e\end{aligned} \tag{4.35}$$

where θ_k are the nodal values at the edge γ_e and

$$a_1^f = \frac{1}{n_V^f}\sum_{k=1}^{n_V^f}\left(\theta_k - [\nabla\theta_\pi^f]^T\, \mathbf{X}_k^f\right) = \frac{1}{n_V^f}\sum_{k=1}^{n_V^f}\theta_k \tag{4.36}$$

Due to the choice of the origin $\mathbf{X}_m$ of the local coordinate system the second term in (3.148) vanishes: $\sum_{k=1}^{n_V^f}\mathbf{X}_k^f = \mathbf{0}$. The projected quantity θ_π is then given at any point in the face by the ansatz

$$\begin{aligned}\theta_\pi^f &= \mathbf{N}_\pi^f\, \mathbf{a}^f \quad \text{with} \\ \mathbf{N}_\pi^f &= \langle 1\, X^f\, Y^f \rangle \quad \text{and} \quad \mathbf{a}^f = \langle\, a_1^f\, a_2^f\, a_3^f\, \rangle^T\end{aligned} \tag{4.37}$$

which can now be applied in (4.33) to compute the integral which leads to

$$\nabla\theta_\pi = \frac{1}{\Omega_v}\sum_{f=1}^{n_F}\int\limits_{\Gamma_f}\theta_\pi^f\, \mathrm{d}\Gamma\, \boldsymbol{N}_f\,. \tag{4.38}$$

The first constant a_1 can be determined as in the two-dimensional case, see Sect. 3.2.2. This completes the ansatz for the temperature (4.30). With the known constants a_2, a_3 and a_4 from (4.38) the constant a_1 follows according to (3.140) as

$$a_1 = \frac{1}{n_V}\sum_{k=1}^{n_V}\left[\theta_k - [\nabla\theta_\pi]^T\mathbf{X}_k\right]. \tag{4.39}$$

With this result the projection is complete for the virtual element Ω_v. It depends on the nodal variables $\boldsymbol{\theta}_v = \langle\theta_1\, \theta_2\, \dots\, \theta_{n_V}\rangle^T$ and can be generally written as

$$\theta_\pi = \mathbf{H}_\theta^{(3,1)}(X, Y, Z)\, \mathbb{P}_\theta^{(3,1)}\boldsymbol{\theta}_v\,. \tag{4.40}$$

References

Beirão da Veiga, L., F. Brezzi, A. Cangiani, G. Manzini, L. Marini, and A. Russo. 2013. Basic principles of virtual element methods. *Mathematical Models and Methods in Applied Sciences* 23 (01): 199–214.

Beirão da Veiga, L., F. Brezzi, L.D. Marini, and A. Russo. 2014. The hitchhiker's guide to the virtual element method. *Mathematical Models and Methods in Applied Sciences* 24 (8): 1541–1573.

Chapter 5
Construction of the Virtual Element

So far we have formulated ways to compute the projection of an ansatz function $\boldsymbol{u}_h$ onto a polynomial space $\boldsymbol{u}_\pi$ for the virtual element method using a linear or a quadratic order interpolation. These projections provide the basis for the discretization of linear and nonlinear partial differential equations related to the Poisson equations and many applications in solid mechanics. Once the approximate field is known its gradients can be computed and with this all equations can be discretized using the ansatz functions derived in Chap. 3 for solids and in Chap. 4 for the Poisson equation.

Classically a discretization technique is based on a weak form or potential of the underlying partial differential equation. In this line of argument, we will employ weak forms, potentials and pseudo-potentials as starting point of the different applications. For generality, we will use a vector valued function $\boldsymbol{u}_h$ next, but the equations hold in a similar way for scalar valued functions.

Before we insert $\boldsymbol{u}_h$ in the weak form, we recall that the approximate solution and the test function can be split into the projected part $\boldsymbol{u}_\pi = \Pi \boldsymbol{u}_h$ and a remainder

$$\boldsymbol{u}_h = \boldsymbol{u}_\pi + (\boldsymbol{u}_h - \boldsymbol{u}_\pi)\,. \tag{5.1}$$

With the arguments related to the projection we can now state the weak form and a potential:

- A weak form can be presented in an abstract way, see (2.85) or (2.93), by including the projection $\boldsymbol{u}_\pi = \Pi \boldsymbol{u}_h$ and the test functions $\boldsymbol{v}_\pi$. With (5.1) one can write

  $$a(\boldsymbol{u}_\pi\,,\,\boldsymbol{v}_\pi) + a(\boldsymbol{u}_h - \boldsymbol{u}_\pi\,,\,\boldsymbol{v}_h - \boldsymbol{v}_\pi) = f(\boldsymbol{v}_\pi)\,. \tag{5.2}$$

  We note that the mixed term $a(\boldsymbol{u}_\pi\,,\,\boldsymbol{v}_h - \boldsymbol{v}_\pi) = a(\boldsymbol{u}_h - \boldsymbol{u}_\pi\,,\,\boldsymbol{v}_\pi)$ is zero due to the orthogonality of the projection, see (3.19).
- In the same way the potential U, see (2.82) or (2.94), can be expressed in terms of the projection $\boldsymbol{u}_\pi$

P. Wriggers et al., *Virtual Element Methods in Engineering Sciences*,
https://doi.org/10.1007/978-3-031-39255-9_5

$$U(\boldsymbol{u}_\pi) + U(\boldsymbol{u}_h - \boldsymbol{u}_\pi) \longrightarrow EXTR. \tag{5.3}$$

In the following we will denote the first terms $a(\boldsymbol{u}_\pi\,,\boldsymbol{v}_\pi)$ and $U(\boldsymbol{u}_\pi)$ in the weak form and the potential *consistency* term while the second part in the above equations is called *stability* term.

A construction of a virtual element which is based only on the consistency term leads to a rank deficient element once the number of vertices n_V is $>$ 3 for two dimensions and exceeds $n_V > 4$ for three dimensions for a linear ansatz. This means that e.g. a two-dimensional solid virtual element of linear ansatz order with n_V nodes has $2\,n_V - 3$ zero eigenvalues—off which 3 zero eigenvalues are related to the rigid body modes—and thus the formulation has to be stabilized.[1] This requirement is equivalent to the case of finite elements when a one-point integration (under integration) is applied. Such underintegrated elements were developed by Flanagan and Belytschko (1981), Belytschko and Bindeman (1991) and many variants were constructed, see e.g. Reese et al. (1999), Reese and Wriggers (2000), Nadler and Rubin (2003), Reese (2003), Mueller-Hoeppe et al. (2009), Korelc et al. (2010), Krysl (2015b). All of these formulations are in one or another way stabilized.

The stabilization is naturally present in (5.2) and (5.3) by the remainder which can be formulated in a generic way where S_{stab} and U_{stab} denote the stability term. This leads to

$$a(\boldsymbol{u}_\pi\,,\boldsymbol{v}_\pi) + S_{stab}(\boldsymbol{u}_h - \boldsymbol{u}_\pi\,,\boldsymbol{v}_h - \boldsymbol{v}_\pi) = f(\boldsymbol{v}_\pi)\,. \tag{5.4}$$

and

$$U(\boldsymbol{u}_\pi) + U_{stab}(\boldsymbol{u}_h - \boldsymbol{u}_\pi) \longrightarrow EXTR. \tag{5.5}$$

Due to the use of the difference $\boldsymbol{u}_h - \boldsymbol{u}_\pi$ in S_{stab} and U_{stab} it is guaranteed that for a very fine mesh the influence of the stabilization on the results vanishes. For a rigorous mathematical description in the case of small strains, see e.g. Beirão da Veiga et al. (2013) and Beirão da Veiga et al. (2014).

Next we will discuss the formulation of the consistency part of the virtual element that stems from the weak form $a(\boldsymbol{u}_\pi\,,\boldsymbol{v}_\pi)$ or potential $U(\boldsymbol{u}_\pi)$. After that, different possibilities for stabilizing the virtual element method will be introduced.

5.1 Consistency Part

This section is based on the most simple case—the Poisson equation for heat conduction, see (2.42),—to illustrate the discretization scheme for virtual elements. The derivation is based on the weak form (2.93) and the potential form (2.94). Both

[1] Higher order ansatz function lead to consistency terms with more non-zero eigenvalues, e.g. a two-dimensional quadratic virtual solid element has $12 - 3 = 9$ non-zero eigenvalues. Nevertheless the elements become rank deficient once they have more vertices than a triangle or in three dimensions than a tetrahedra.

approaches yield, of course, identical results. However implementation might lead for one of these forms to more efficient computing code and thus both possibilities are stated here.

5.1.1 Weak Form

Let us recall the weak form for the heat conduction equation that was stated in (2.93) over the region Ω

$$\int_{\Omega} \left[\operatorname{Grad} \vartheta \cdot k \operatorname{Grad} \theta - \vartheta \, \varrho_0 R \, \right] \, \mathrm{d}\Omega + \int_{\Gamma_N} \vartheta \; \bar{\boldsymbol{Q}} \cdot \boldsymbol{n} \, \mathrm{d}\Gamma = 0 \tag{5.6}$$

where the scalar variable θ is the temperature and ϑ the test function, both function depend in the two-dimensional case on the coordinates X, Y. The thermal conductivity k is assumed to be constant as well as the source term $\varrho_0 R$.

Generally when writing this weak form as $a(\vartheta, \theta) - f(\vartheta) = 0$ the discretization leads to a process where all contributions of the elements Ω_v that are employed to discretize the region have to be assembled

$$a(\vartheta_h, \theta_h) - f(\vartheta_h) = \mathop{\mathsf{A}}_{v=1}^{n_v} \left[a_v(\vartheta_h, \theta_h) - f_v(\vartheta_h) \right] = 0 \tag{5.7}$$

where $a_v(\vartheta_h, \theta_h)$ and $f_v(\vartheta_h)$ are the contribution of each virtual element Ω_v and n_v is the total number of elements.

The gradients in (5.6) can now be discretized by using the ansatz functions resulting from the projections in Sect. 3.1.2. This will be detailed next for a linear and a quadratic ansatz leading to the matrix form of one element Ω_v. For this task the projected gradients are approximated for the variable and the test function, $\nabla \theta_\pi$ and $\nabla \vartheta_\pi$, respectively.

- **Linear ansatz**. For a linear ansatz the projected gradient is given by (4.13) as $\nabla \theta_\pi = \mathbb{P}_{\nabla\theta}^{(2,1)} \boldsymbol{\theta}_v$ which can also be utilized for the test function $\nabla \vartheta_\pi = \mathbb{P}_{\nabla\theta}^{(2,1)} \boldsymbol{\vartheta}_v$. This yields for the first term in the above equation

$$\int_{\Omega_v} \operatorname{Grad} \vartheta_\pi \cdot k \operatorname{Grad} \theta_\pi \, \mathrm{d}\Omega = \Omega_v \, \boldsymbol{\vartheta}_v^T [\mathbb{P}_{\nabla\theta}^{(2,1)}]^T \mathbb{P}_{\nabla\theta}^{(2,1)} \boldsymbol{\theta}_v = \boldsymbol{\vartheta}_v^T \mathbf{K}_v^{(2,1)} \, \boldsymbol{\theta}_v \tag{5.8}$$

since the projection matrix $\mathbb{P}_{\nabla\theta}^{(2,1)}$ is constant. Due to the fact that the nodal contributions of the test functions cancel out in the assembly process, the vector $\boldsymbol{\vartheta}_v$can be neglected in the final equation system. The size of the element matrix $\mathbf{K}_v^{(2,1)}$ is in this case of the size of the nodal values at the element $(n_V \times n_V)$. We note, that no integration has to be performed and thus the computation of the element matrix of the virtual element is very fast and can be performed component wise by exploiting the structure of $\mathbb{P}^{(2,1)}_{\nabla\theta}$, see (4.12).

The volumetric part of the right hand side in (5.7) is given by

$$\int\limits_{\Omega_v} \vartheta_h\, \varrho_0 R \;\mathrm{d}\Omega\,. \tag{5.9}$$

Since the test function is not know it will be replaced by its projection (4.17), see the arguments related to (3.37), which then leads to

$$\int\limits_{\Omega_v} \vartheta_\pi\, \varrho_0 R \;\mathrm{d}\Omega = \boldsymbol{\vartheta}_v^T [\mathbb{P}^{(2,1)}_\theta]^T \int\limits_{\Omega_v} [\mathbf{H}^{(2,1)}_\theta(X,Y)]^T \varrho_0 R \;\mathrm{d}\Omega = \boldsymbol{\vartheta}_v^T \mathbf{f}_v^{(2,1)}\,. \tag{5.10}$$

The remaining integral can now be evaluated exactly using the relations in Appendix A for any polynomial order. This leads to the loading vector $\mathbf{f}_v^{(2,1)}$ of size $(n_V \times 1)$ for an element.

Note, that the surface integral in (2.93) can be easily evaluated over the edges of elements where the flux is prescribed since the ansatz for ϑ_h is known at the edges.

- **Quadratic ansatz**. In Remark 3.8 in Sect. 3.1.6 the quadratic ansatz and its projection was considered for the heat conduction equation. This ansatz can now be inserted into (2.93). Using Eq. (4.24) the first term of the weak form follows

$$\begin{aligned}&\int\limits_{\Omega_v} \operatorname{Grad}\vartheta_\pi \cdot k \operatorname{Grad}\theta_\pi \;\mathrm{d}\Omega\\ &\quad= \boldsymbol{\vartheta}_v^T [\mathbb{P}^{(2,2)}_{\nabla\theta}]^T \int\limits_{\Omega_v} k\,[\mathbf{B}^{(2,2)}_{\theta\,\pi}(X,Y)]^T\,\mathbf{B}^{(2,2)}_{\theta\,\pi}(X,Y)\;\mathrm{d}\Omega\,\mathbb{P}^{(2,2)}_{\nabla\theta}\,\boldsymbol{\theta}_v = \boldsymbol{\vartheta}_v^T \mathbf{K}_v^{(2,2)}\,\boldsymbol{\theta}_v\,.\end{aligned} \tag{5.11}$$

The integral in this equation includes polynomials up to second order inside and can be evaluated exactly by the formulae provided in Appendix A. The resulting stiffness matrix $\mathbf{K}_v^{(2,2)}$ of the virtual element has the size $(2n_V + 1 \times 2n_V + 1)$.

The volumetric loading in (5.7) can be written by using the arguments for the linear ansatz. Since the test function is not know it will be replaced by its projection (4.29) leading to

$$\int\limits_{\Omega_v} \vartheta_\pi\, \varrho_0 R \;\mathrm{d}\Omega = \boldsymbol{\vartheta}_v^T [\mathbb{P}^{(2,2)}_\theta]^T \int\limits_{\Omega_v} [\mathbf{H}^{(2,2)}_\theta(X,Y)]^T \varrho_0 R \;\mathrm{d}\Omega = \boldsymbol{\vartheta}_v^T \mathbf{f}_v^{(2,2)}\,. \tag{5.12}$$

The integral contains the second order polynomial $\mathbf{H}^{(2,2)}_\theta(X,Y)$, additionally polynomials that are used to describe $\varrho_0 R$ have to be considered which might enlarge the polynomial order. The evaluation is possible for any order polynomials using (A.8) in Appendix A.

5.1.2 Potential

Instead of using the weak form, the virtual element formulation can be based on the potential function (2.94). Thus we have, by summing up (assembling) all element contributions for the n_v virtual elements,

$$U(\boldsymbol{u}) = \mathop{\mathsf{A}}_{v=1}^{n_v} \left[U_c^v(\theta_\pi) + U_{stab}^v(\theta_h - \theta_\pi) \right] . \tag{5.13}$$

where the first term is the consistency part and the second term is the potential energy for the stabilization of the virtual element. The element contributions for the consistency part can be formulated as

$$U_c^v(\theta_\pi) = \frac{1}{2} \int\limits_{\Omega_v} \operatorname{Grad} \theta_\pi \cdot k \operatorname{Grad} \theta_\pi \; \mathrm{d}\Omega \, . \tag{5.14}$$

The discretization of this integral only needs the introduction of the projected gradient for the linear and quadratic ansatz.

- **Linear ansatz**. From (5.8) one derives immediately

$$\frac{1}{2} \int\limits_{\Omega_v} \operatorname{Grad} \theta_\pi \cdot k \operatorname{Grad} \theta_\pi \; \mathrm{d}\Omega = \frac{k}{2} \boldsymbol{\theta}_v^T \, \Omega_v \, [\mathbb{P}_{\nabla\theta}^{(2,1)}]^T \mathbb{P}_{\nabla\theta}^{(2,1)} \boldsymbol{\theta}_v = \frac{1}{2} \boldsymbol{\theta}_v^T \mathbf{K}_v^{(2,1)} \, \boldsymbol{\theta}_v \tag{5.15}$$

since the projection matrix $\mathbb{P}_{\nabla\theta}^{(2,1)}$ is constant.
The volumetric part of the right hand side in (5.7) is given by

$$\int\limits_{\Omega_v} \theta_\pi \, \varrho_0 R \; \mathrm{d}\Omega = \boldsymbol{\theta}_v^T [\mathbb{P}_\theta^{(2,1)}]^T \int\limits_{\Omega_v} [\mathbf{H}_\theta^{(2,1)}(X, Y)]^T \varrho_0 R \; \mathrm{d}\Omega = \boldsymbol{\theta}_v^T \, \mathbf{f}_v^{(2,1)} \, . \tag{5.16}$$

- **Quadratic ansatz**. Following (5.11) we obtain analogously

$$\begin{aligned} &\frac{1}{2} \int\limits_{\Omega_v} \operatorname{Grad} \theta_\pi \cdot k \operatorname{Grad} \theta_\pi \; \mathrm{d}\Omega \\ &\quad = \frac{1}{2} \boldsymbol{\theta}_v^T \, [\mathbb{P}_{\nabla\theta}^{(2,2)}]^T \int\limits_{\Omega_v} k \, [\mathbf{B}_{\theta\,\pi}^{(2,2)}(X, Y)]^T \, \mathbf{B}_{\theta\,\pi}^{(2,2)}(X, Y) \, \mathrm{d}\Omega \; \mathbb{P}_{\nabla\theta}^{(2,2)} \, \boldsymbol{\theta}_v \\ &\quad = \frac{1}{2} \boldsymbol{\theta}_v^T \, \mathbf{K}_v^{(2,2)} \, \boldsymbol{\theta}_v \, . \end{aligned} \tag{5.17}$$

The polynomial order in the integral is up to second order. In the same way as in (5.12) the volumetric loading term follows, only ϑ_π has to be replaced by θ_π. This yields

$$\int_{\Omega_v} \theta_\pi\, \varrho_0 R\, \mathrm{d}\Omega = \boldsymbol{\theta}_v^T\, [\mathbb{P}_\theta^{(2,2)}]^T \int_{\Omega_v} [\mathbf{H}_\theta^{(2,2)}(X,Y)]^T \varrho_0 R\, \mathrm{d}\Omega = \boldsymbol{\theta}_v^T\, \mathbf{f}_v^{(2,2)}\,. \tag{5.18}$$

Again, the evaluation of these integrals is possible for any order of the polynomial, see Appendix A.

5.2 Stabilization Techniques for Virtual Elements

It is clear from the construction of the virtual element, that a stabilization is necessary. Stabilization is known in many other discretization techniques for a long time. Several approaches were introduced within finite element methodologies to stabilize formulations. Among them are hour-glass stabilization techniques and mixed approaches, like enhanced and assumed strain methods.

The latter formulations do not need to introduced stability parameters but have not been explored in depth in virtual element methods, for first attempts see Lamperti et al. (2023). Other approaches that yield stabilization free formulations of the virtual element method can be found in e.g. D'Altri et al. (2021) and Chen and Sukumar (2023a, b). These methods require an increased computational effort on element level because an enriched ansatz space is used to obtain the proper rank for a virtual element. However they circumvent the need to select a stabilization parameter which eventually can spoil the accuracy of the analysis.

Several techniques for the stabilization of virtual elements that are appropriate for different applications, especially for solid mechanics, were developed in e.g. Beirão da Veiga et al. (2013), Beirão da Veiga et al. (2015), Chi et al. (2017), Artioli et al. (2017) and Wriggers et al. (2017). As already mentioned in the beginning of this chapter, the methodologies can be basically split into two approaches. These are a stabilization by

- a bi-linear form

$$S_{stab} = \frac{\alpha_{|\Omega_v}}{2} \int_{\Omega_v} [(\boldsymbol{u}_h - \boldsymbol{u}_\pi) \cdot \mathbf{S}\, (\boldsymbol{v}_h - \boldsymbol{v}_\pi)]\, \mathrm{d}\Omega \tag{5.19}$$

  and variants depending on the choice of $\alpha_{|\Omega_v}$ and the matrix $\mathbf{S}$, see e.g. Beirão da Veiga et al. (2013), Beirão da Veiga et al. (2017) and Artioli et al. (2017).
- A difference between a new potential energy $\widehat{U}$, evaluated for $\boldsymbol{u}_h$ and $\boldsymbol{u}_\pi$

$$U_{stab} = \widehat{U}(\boldsymbol{u}_h) - \widehat{U}(\boldsymbol{u}_\pi)\,, \tag{5.20}$$

  see e.g. Wriggers et al. (2017). The potential energy $\widehat{U}$ can be freely chosen. An obvious choice would be to pick the same form as U_c, however with different material constants, which can be selected such that the response of the virtual element is enhanced.

Stabilization methods depend usually on one or more parameters that have to be selected by the user. Good techniques provide for different types of applications correct solutions for one parameter set. Thus a stabilization technique has to be designed on the basis that it is independent of the problem to be solved. Additionally, in non-linear formulations, the stabilization term has to be linearized such that quadratic convergence of a Newton type solution algorithm is achieved. In the following both approaches are discussed in more detail.

5.2.1 Stabilization by a Discrete Bi-Linear Form

The first possibility to formulate the stabilization part S_{stab} is based on the split in Eq. (5.4), see Beirão da Veiga et al. (2013). The stabilization term (5.19) is then approximated as a sum over all vertex nodes n_V of the virtual element Ω_v

$$S_{stab} = \gamma(\Omega_v) \sum_{k=1}^{n_V} [\mathbf{u}_k - \mathbf{u}_\pi(\mathbf{X}_k)] \cdot [\mathbf{v}_k - \mathbf{v}_\pi(\mathbf{X}_k)] \; . \tag{5.21}$$

This stabilization depends on the nodal degrees of freedom (d.o.f.) and thus it is called sometimes DOFI-stabilization. in (5.21), γ is a stabilization parameter, $\mathbf{u}_\pi$ is the ansatz function for the projection evaluated at the vertex $\mathbf{X}_k$, see Chap. 3 for the computation of the projection in the two-and three-dimensional case.[2] The vector $\mathbf{v}$ denotes the test function. Since all terms are given as functions of the unknown nodal displacements the element residual and tangent stiffness can be computed directly without any numerical integration.

A different approach to formulate the stability term can be found in Artioli et al. (2017). They use the fact that a linear relation exists between the nodal variables at the vertices Ω_v and the parameters $\mathbf{a}$ that define the gradient and the projection $\boldsymbol{u}_\pi$ in terms of the ansatz function $\mathbf{H}(X, Y)$, see e.g. for the two-dimensional linear case (3.52). Thus this equation can be evaluated at each vertex and be written in general for any order of ansatz (here for the case of two-dimensional solid mechanics)

$$\mathbf{u}_v = \begin{Bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \dots \\ \mathbf{u}_{n_V} \end{Bmatrix} = \begin{bmatrix} \mathbf{H}(\mathbf{X}_1) \\ \mathbf{H}(\mathbf{X}_2) \\ \dots \\ \mathbf{H}(\mathbf{X}_{n_V}) \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ \dots \\ a_{2n_p} \end{Bmatrix} \quad \rightarrow \mathbf{u}_v = \mathbf{D}\,\mathbf{a} \tag{5.22}$$

leading to a $(2n_V \times 2n_p)$ matrix where n_p depends on the degree n of the ansatz polynomial. In general n_p follows from $n_p = \sum_{i=1}^{n+1} i$. In Eq. (5.22) $\mathbf{u}_k$ is the displacement at vertex k and $\mathbf{H}(\mathbf{X}_k)$ the ansatz evaluated at the vertex $\mathbf{X}_k$. Now the stabilization matrix for a virtual element Ω_v is defined directly as

[2] Note, that there is an alternative way of writing $\mathbf{H}$ by using the different deformation modes. For details of this approach, see Gain et al. (2014) and Wriggers et al. (2016a).

$$\mathbf{K}_v^{sD} = \gamma(\Omega_v)\,\left[\mathbf{I} - \mathbf{D}\,(\mathbf{D}^T\,\mathbf{D})^{-1}\,\mathbf{D}^T\right] \tag{5.23}$$

and has the size $(2n_V \times 2n_V)$ in the two-dimensional case.

What remains is the choice of the stabilization parameter γ for an element Ω_v in (5.21) and (5.23). Good choices were proposed in Beirão da Veiga et al. (2013) and Artioli et al. (2017) for two-dimensional situations. The stabilization parameter can generally be written as

$$\gamma(\Omega_v) = \alpha_{|\Omega_v}\,\frac{h_v^2}{\mathrm{tr}\,[\mathbf{H}(\mathbf{X}_k)^T\mathbf{H}(\mathbf{X}_k)]} \tag{5.24}$$

and depends in nonlinear situations on the primal variable. The ansatz function for the projection $\mathbf{H}(\mathbf{X}_k)$ is evaluated at the vertex $\mathbf{H}(\mathbf{X}_k)$, h_v is a characteristic element diameter and $\alpha_{|\Omega_v}$ is a parameter that can be picked as follows:

- *Constant parameter*: $\alpha_{|\Omega_v} = \alpha_0\,\mathrm{tr}\,[\mathbb{C}]$ where $\mathbb{C}$ is the constant constitutive tensor related to the problem and α_0 is a constant to be selected by the user. In many cases it can be chosen as $\alpha_0 = 1$. This approach works well in a wide range of applications.
- *Distributed parameters*: In some applications it makes sense to adjust the stabilization parameter related to the entries of the stiffness matrix which leads to a distribution of parameter values within on element. This can be achieved by introducing a digonal matrix $\mathbb{S}_d$

$$S_{stab}^d = \sum_{k=1}^{n_V}[\mathbf{u}_k - \mathbf{u}_\pi(\mathbf{X}_k)]\cdot\mathbb{S}_{d\,i}\,[\mathbf{v}_k - \mathbf{v}_\pi(\mathbf{X}_k)]\ . \tag{5.25}$$

  where now the entries of $\mathbb{S}_{d\,i}$ are given by $\max\{\alpha_{|\Omega_v}\ ,\ K_{v\,ii}\}$ with $K_{v\,ii}$ being the diagonal entries of the stiffness matrix $\mathbf{K}_v$ stemming from the projection within a virtual element. As discussed in Beirão da Veiga et al. (2017), this approach yields better accuracy and has advantages for higher order ansatz spaces.
- *Variable parameter*: The parameter $\alpha_{|\Omega_v}$ depends on the primal variable $\boldsymbol{u}$ if the problem at hand is nonlinear. A good choice is to make it dependent on the trace of the constitutive tensor leading to e.g. $\alpha_{|\Omega_v}(\boldsymbol{u}) = \alpha_0\,\mathrm{tr}\,[\mathbb{C}(\boldsymbol{C}(\boldsymbol{u}))]$ where $\boldsymbol{C}(\boldsymbol{u})$ is the right Cauchy-Green tensor.
- *Stiffness related parameter*: Another possibility is to use the trace of the tangent matrix for the stabilization parameter. With N being the total number of unknowns of the global equation system the parameter is given by the norm of the diagonal of the stiffness matrix

$$\alpha_{|\Omega_v} = \frac{1}{2N}\sqrt{\sum_{i=1}^{N}(\mathbf{K}_{ii})^2} = \frac{1}{2N}||\mathbf{K}_D||\,. \tag{5.26}$$

  This type of stabilization works well for linear problems. Instead of N also $\sqrt{N}$ can be used in (5.26). In nonlinear problems $\mathbf{K}$ depends on the primal variable $\boldsymbol{u}$

and it is not easily possible to linearize this term in a consistent manner. In such case it is advisable to use the tangent matrix of the last loading step.

Still, all choices for the stabilization parameter present a challenge when linearizing in a Newton procedure, even with the use of the symbolic tool *AceGen*, see Korelc and Wriggers (2016), which automatically provides the code for the residual vector and tangent stiffness matrix. If linearization is not possible, like in (5.26), then the parameter has to be computed at the last converged load step and kept constant during the iteration within the load step. This leads often to incremental load stepping, even if the physical problem allows the computation of the solution in one step.

Another drawback in using the form (5.21) in nonlinear computations, without keeping the parameter $\alpha_{|\Omega_v}$ constant within a load step, is related to the fact that the linearization of S_{stab} yields a non-symmetric tangent matrix. A remedy is to write the bi-linear forms S_{stab} in (5.21) and (5.25) as potentials which yields

$$U_{stab} = \frac{\gamma(\Omega_v)}{2} \sum_{k=1}^{n_V} [\mathbf{u}_k - \mathbf{u}_\pi(\mathbf{X}_k)] \cdot [\mathbf{u}_k - \mathbf{u}_\pi(\mathbf{X}_k)] \tag{5.27}$$

and

$$U^d_{stab} = \frac{1}{2} \sum_{k=1}^{n_V} [\mathbf{u}_k - \mathbf{u}_\pi(\mathbf{X}_k)] \cdot \mathbb{S}_d\, [\mathbf{u}_k - \mathbf{u}_\pi(\mathbf{X}_k)]\ . \tag{5.28}$$

and linearizations produce symmetric tangent matrices.

In problems that involve inelastic or other history depending deformations one has to take care of the change in stiffness due to the non-linear constitutive behaviour in a virtual element when selecting the stabilization parameter. Details are provided in Chap. 8.

5.2.2 Energy Stabilization

A different stabilization technique was proposed in Wriggers et al. (2017) for problems in solid mechanics. The essence of the approach is to introduce for stabilization a new, positive definite strain energy $\hat{\Psi}$. The total energy is then the sum of $U_c(\boldsymbol{u}_\pi)$, the original energy in the consistency part, and the stabilization term $U_{stab}(\boldsymbol{u}_h - \boldsymbol{u}_\pi)$ constructed from the strain energy $\hat{\Psi}$

$$U_{stab} = \widehat{U}(\boldsymbol{u}_h) - \widehat{U}(\boldsymbol{u}_\pi)\,. \tag{5.29}$$

The second term on the right side ensures the consistency of the total potential energy, which is now given by

$$U(\boldsymbol{u}_h) = U_c(\boldsymbol{u}_\pi) + \widehat{U}(\boldsymbol{u}_h) - \widehat{U}(\boldsymbol{u}_\pi)\,. \tag{5.30}$$

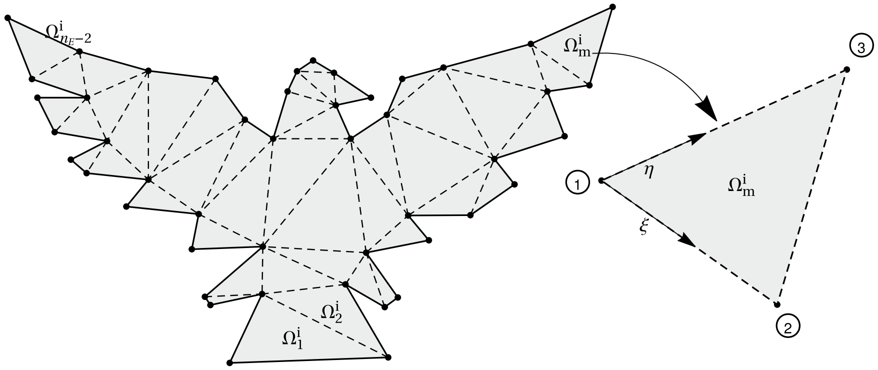


Fig. 5.1 Internal triangular mesh

The potential $\widehat{U}$ is defined by the elastic strain energy as

$$\widehat{U}(\boldsymbol{u}_h) = \sum_{v=1}^{n_v} \int_{\Omega_v} \widehat{\Psi}(\boldsymbol{u}_h)\, d\Omega \quad \text{and} \quad \widehat{U}(\boldsymbol{u}_\pi) = \sum_{v=1}^{n_v} \int_{\Omega_v} \widehat{\Psi}(\boldsymbol{u}_\pi)\, d\Omega\,. \tag{5.31}$$

The terms involving $\boldsymbol{u}_\pi$ can be integrated for the linear case as in (5.15) and analogously for higher order ansatz spaces. However, the term involving the displacement $\boldsymbol{u}_h$ is not computable since $\boldsymbol{u}_h$ is not known within the element.

The integration of $\widehat{U}(\boldsymbol{u}_h)$ can be approximated by inscribing a triangular mesh in the virtual element with nodal points being equivalent to the nodal points of the original virtual element Ω_v, see Fig. 5.1. It consists of n_{int} linear three-noded triangles that are connected to the nodes of the virtual element. By this construction the displacement field $\boldsymbol{u}_h$ in (5.30) can be approximated in each triangle Ω^i_m by a finite element ansatz $\boldsymbol{u}_h = \sum_{i=1}^{3} N_i(\xi, \eta)\, \mathbf{u}_i$ with local coordinates of a triangle (ξ, η) and shape functions $N_1 = \xi$, $N_2 = \eta$ and $N_3 = 1 - \xi - \eta$. With this interpolation of $\boldsymbol{u}_h$ the stabilization energy can be computed. The specific choice of the mesh does not introduce extra degrees of freedom since it just uses the vertices describing the virtual element.

Note that the subtraction of the piecewise-constant part $\widehat{U}(\boldsymbol{u}_\pi)$ in (5.30) is necessary to ensure that the stabilization energy does not influence the convergence with respect to the mesh size. Once the element size is very small, the strain tends to be constant in each virtual element. In that case $\widehat{U}(\boldsymbol{u}_h)$ will approach $\widehat{U}(\boldsymbol{u}_\pi)$ and thus vanish in the limit for small virtual elements. Hence stabilization does not influence the final result.

It remains to define the strain energy $\widehat{\Psi}$. This has to be done in relation to the problem that has to be analysed. Thus the strain energy $\widehat{\Psi}$ will be introduced in the later chapters that are devoted to specific applications. In this respect it is interesting that the stabilization energy has not to be the same as the strain energy of the original problem defining the physical behaviour. Also the constitutive parameters can be different and, actually, have to be selected in a specific way to obtain an efficient and accurate performance of the formulation.

Remark 5.1 Let us assume a very special case where the stabilization energy $\widehat{U}$ has the same form as the energy U_c that describes the physics in the consistency part. Furthermore we assume that the choice of the constitutive parameters in $\widehat{U}$ is such, that we can write $\widehat{U} = \gamma \, U_c$ with γ being a constant parameter. This leads with (5.30) to

$$U = (1-\gamma)\, U_c(\boldsymbol{u}_\pi) + \gamma \, U_c(\boldsymbol{u}_h)\,. \tag{5.32}$$

In this special case we have a sum of parts stemming from the virtual element method (consistency part $U_c(\boldsymbol{u}_\pi)$) and from the finite element method (stabilization energy $U_c(\boldsymbol{u}_h)$). Thus the energy stabilization leads to a mixture of virtual and finite element discretization with the extreme cases

- $\gamma = 1$: pure finite element solution, using just the internal mesh. It can be shown that these results are not as good as the ones when using a mixture of VEM and FEM, see the subsection "VEM versus FEM" in an example in Sect. 6.2.5.1.
- $\gamma = 0$: pure virtual element solution. This however yields a rank deficient tangent matrix and thus solutions cannot be computed.

Hence the coupling of both energies is necessary to solve a problem in solid mechanics correctly with the virtual element method. ■

Remark 5.2 The energy stabilization can also be used for heat conduction. The potential form (2.94) yields

$$\widehat{Q} = \int\limits_{\Omega} \frac{\widehat{k}}{2} (\nabla \, \theta_h \cdot \nabla \, \theta_h) \, \mathrm{d}\Omega\,. \tag{5.33}$$

As before—with $\widehat{k} = \gamma \, k$—the virtual element formulation for the heat conduction problem is a sum of virtual and finite element parts

$$Q = (1-\gamma)\, Q_c(\theta_\pi) + \gamma \, Q_c(\theta_h)\,. \tag{5.34}$$

Here a linear ansatz is formulated to approximate the temperature field θ_h within each triangle Ω^i_m of the inscribed mesh, see Fig. 5.1. The gradient in (5.33) can be easily computed based on an isoparametric formulation for the three noded triangle. The latter yields with the ansatz

$$\theta_h = \sum_{I=1}^{3} N_I(\xi\,, \eta)\, \theta_I \longrightarrow \nabla\theta_h = \sum_{I=1}^{3} \nabla_x \, N_I(\xi\,, \eta)\, \theta_I \tag{5.35}$$

with $N_1 = \xi\,, \quad N_2 = \eta \quad$ and $N_3 = 1 - \xi - \eta$. Using this interpolation in (5.34) for the second term we can compute the stabilization energy for the virtual element. Note that the gradient $\nabla \, \theta_h$ is constant over each inscribed triangle Ω^i_m which yields in the linear case a very efficient code. Let us denote the resulting stabilization matrix stemming from $Q_c(\theta_h)$ by $\mathbf{K}_v^s$. This matrix follows from the assembly of all contributions of the triangles Ω_m^i, see Fig. 5.1. ■

Remark 5.3 When the energy stabilization is employed for higher order ansatz functions with $n \geq 2$ then internal nodes have to be defined for the internal triangularization. This yields a formulation with additional unknowns which is not as efficient, see De Bellis et al. (2019). ■

5.3 Assembly to the Global Equation System

All element contributions have now to be assembled for a specific discretization with n_v virtual elements. Either the first variation of Q is performed, see (5.15), (5.18) and (5.34), or the discretized weak form is used directly based on (5.8) or (5.11) for linear and quadratic ansatz functions, respectively. Naturally, variation of the potential and weak form lead to the same result.

In the previous sections the element stiffness matrices for virtual elements were derived for linear or quadratic ansatz functions ($n = 1, 2$). For these ansatz orders we can assemble all element matrices

$$\mathop{\mathsf{A}}_{v=1}^{n_v} \boldsymbol{\vartheta}_v^T \left[\left\{ (1-\gamma)\, \mathbf{K}_v^{(2,n)} + \gamma\, \mathbf{K}_v^s \right\} \boldsymbol{\theta}_v - \mathbf{f}_e^{(2,n)} \right] = 0 \tag{5.36}$$

to obtain a global matrix system that has to be solved in a heat conduction problem

$$\left[(1-\gamma)\, \mathbf{K}_c^{(2,n)} + \gamma\, \mathbf{K}_s \right] \boldsymbol{\theta} = \mathbf{f}_c^{(2,n)} \tag{5.37}$$

where $\boldsymbol{\theta}$ is the global unknown vector and

$$\mathbf{K}_c^{(2,n)} = \mathop{\mathsf{A}}_{v=1}^{n_v} \mathbf{K}_v^{(2,n)}\,, \quad \mathbf{K}_s = \mathop{\mathsf{A}}_{e=1}^{n_v} \mathbf{K}_v^s \quad \text{and } \mathbf{f}_c^{(2,n)} = \mathop{\mathsf{A}}_{v=1}^{n_v} \mathbf{f}_v^{(2,n)} \tag{5.38}$$

are the global matrices and global loading vector, respectively.

5.4 Numerical Example for the Poisson Equation

The use of the virtual element method will be illustrated by a numerical example in which we employ the Poisson equation to compute the deflection w of a membrane.[3] For the membrane the conductivity coefficient k of the thermal equation has the

[3] The residuals and matrices for this example were obtained with *AceGen* and the computations are performed with *AceFEM*, see e.g. Korelc (2002) and Korelc and Wriggers (2016). However, we

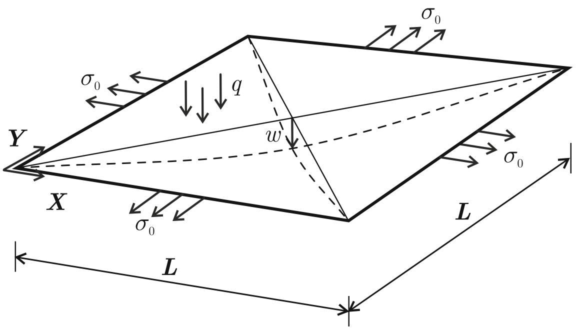


Fig. 5.2 Pre-stressed, simply supported membrane under a load q

meaning of a pre-stress σ_0 acting in the plane of the membrane. Furthermore the source term $\rho_0\, R$ in (5.9) has to be replaced by the scaled load $\bar{q} = q\,/\,t$ where t is the thickness of the membrane. The structure of the membrane is depicted in Fig. 5.2 which shows the uniform pre-stress σ_0, the load q on a square of length and width L. The membrane is simply supported at the entire boundary.

Within the computations the variables are set to $\sigma_0 = 5$, $L = 2$ and $\bar{q} = 1$.

The deflection w is approximated by a linear ansatz within the virtual element. Thus, the virtual element is based on the potential formulation provided in Eqs. (5.15) and (5.16) which now are written with the constants and variables related to the membrane

$$\frac{1}{2}\int\limits_{\Omega_v} \text{Grad}\, w_\pi \cdot \sigma_0\, \text{Grad}\, w_\pi \, \mathrm{d}\Omega = \frac{\sigma_0}{2}\, \boldsymbol{w}_v^T\, \Omega_v\, [\mathbb{P}_{\nabla w}^{(2,1)}]^T \mathbb{P}_{\nabla w}^{(2,1)}\, \boldsymbol{w}_v = \frac{1}{2}\, \boldsymbol{w}_v^T \mathbf{K}_{m\,v}^{(2,1)}\, \boldsymbol{w}_v \tag{5.39}$$

where $\mathbf{K}_{m\,v}^{(2,1)}$ is the element stiffness of the membrane element which is constant since the projection matrix $\mathbb{P}_{\nabla w}^{(2,1)} \equiv \mathbb{P}_{\nabla \theta}^{(2,1)}$ is constant.

The volumetric part of the right hand side in (5.16) is given with $\mathbb{P}_{w}^{(2,1)} \equiv \mathbb{P}_{\theta}^{(2,1)}$ by

$$\int\limits_{\Omega_v} w_\pi\, \frac{q}{t}\, \mathrm{d}\Omega = \boldsymbol{w}_v^T [\mathbb{P}_{w}^{(2,1)}]^T \int\limits_{\Omega_v} [\mathbf{H}_{w}^{(2,1)}(X, Y)]^T \frac{q}{t}\, \mathrm{d}\Omega = \boldsymbol{w}_v^T\, \mathbf{f}_{m\,v}^{(2,1)}\,. \tag{5.40}$$

In case of a constant load q the load vector can be computed with the linear ansatz $\mathbf{H}_{w}^{(2,1)} = \{1\,,\, X\,,\, Y\}$ and making use of (4.14) and with (A.10) it follows

$$\mathbf{f}_{m\,v}^{(2,1)} = [\mathbb{P}_{w}^{(2,1)}]^T \left\{ \begin{matrix} \Omega_v \\ \int\limits_{\Omega_v} X\, \mathrm{d}\Omega \\ \int\limits_{\Omega_v} Y\, \mathrm{d}\Omega \end{matrix} \right\} = [\mathbb{P}_{w}^{(2,1)}]^T \left\{ \begin{matrix} 1 \\ X_b \\ Y_b \end{matrix} \right\} \frac{q\, \Omega_v}{t}\,. \tag{5.41}$$

note that *MATLAB* codes related to the virtual element method are freely available for the Poisson equation, see e.g. Sutton (2017) and Yu (2022).

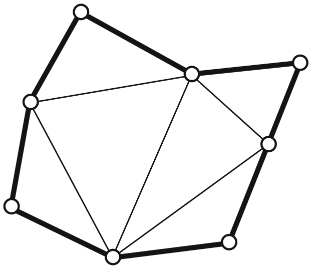


Fig. 5.3 A typical virtual element with 8 vertices and an internal mesh for stabilization

The integrals can equivalently be evaluated as a sum over the vertices of the virtual element using A.3 and A.4 but it may be more efficient to use the evaluation at the barycenter (X_b, Y_b).

The stabilization is performed according to Sects. 5.2.1 and 5.2.2. For the stabilization with the bi-linear form the specific ansatz described in (5.21) was applied with a stabilization parameter $\gamma = \text{tr}\,\mathbf{K}^{(2,1)}_{m\,v}$. For the energy stabilization, which is based on the potential, an internal triangular mesh is constructed in every virtual element. In the examples, a virtual element with linear ansatz is employed for all computations. It consists of 8 vertices, as depicted in Fig. 5.3. The internal mesh that is inscribed for the energy stabilization according to (5.35) can be defined by 6 triangles as shown in Fig. 5.3. This internal triangular mesh can be set up in a form that optimizes the element shape.[4]

5.4.1 Quadratic Membrane

The analysis is performed for different meshes, see Fig. 5.4. A uniform mesh with equal spacing is depicted on the left side and a mesh where the virtual elements have randomly generated shape is shown on the right side. Both meshes have the same number of elements and nodes using Q2S elements with eight nodes, for the definition of Q2S see Fig. B.1 in Appendix B. The randomly generated mesh includes elements that have a non-convex shape which is generally not possible within a finite element analysis.

Since the problem is linear the load can be applied in one step. This leads to a deflection in the membrane as depicted in Fig. 5.5 for the numerical simulation using the bi-linear stabilization. Also the energy stabilization with the inscribed triangular mesh yields the same deflection. Hence only one result is shown for the two mesh types.

[4] It is well known in finite element theory that acute triangular elements lead to a high constant in the error estimate and influence the convergence behaviour negatively. These degenerated triangles can occur in an internal mesh depending on the shape of the virtual element. However since the elements are only used as stabilization they do not affect the overall convergence properties of the virtual element mesh.

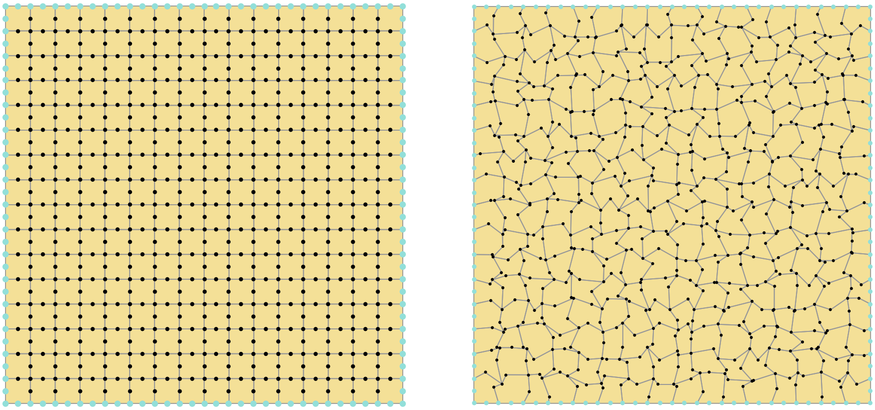


Fig. 5.4 Uniform mesh (16 × 16), Random mesh (16 × 16)

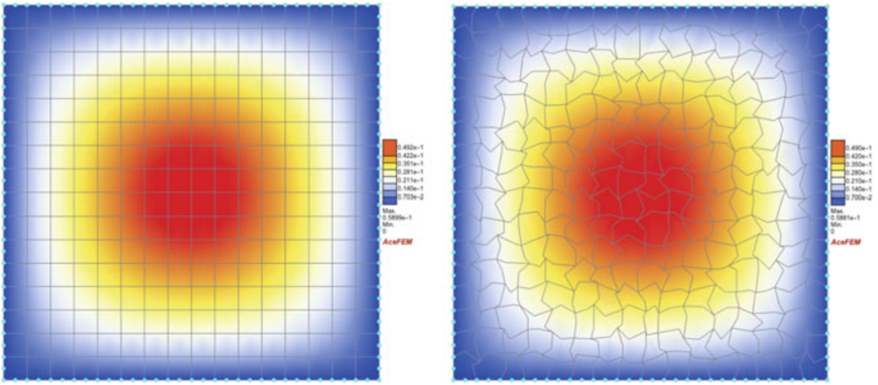


Fig. 5.5 Deflection of the membrane. On the left: uniform mesh (16 × 16), on the right: random mesh (16 × 16)

The solution is the same for the uniform and the random mesh, see Fig. 5.5. Thus the distortion of the elements on the right side of Fig. 5.4 does not influence the solution behaviour. This can also be seen in a convergence study with different number of elements ranging from a (2 × 2) mesh with 8 elements to a (128 × 128) mesh with 16384 elements. The results were compared with an analytical solution[5] that is denoted by w_{ex} and has the value $w_{ex} = 0.0589371$. Figure 5.6 shows a log-log plot of the scaled maximum norm of the deflection at the center of the membrane $(|w - w_{ex}|)/w_{ex}$ versus the number of elements. The uniform mesh depicts a linear

[5] For this example it is possible to compute an exact solution, see Timoshenko and Goodier (1970). The deflection follows from the series

$$w_{ex} = \frac{16\bar{q}L^2}{\sigma_0\pi^3} \sum_{n=1,3,5,\ldots}^{\infty} \frac{1}{n^3}(-1)^{(n-1)/2}\left[1 - \frac{\cosh(n\pi y/2L)}{\cosh(1/2n\pi)}\right]\cos\frac{n\pi x}{2L}$$

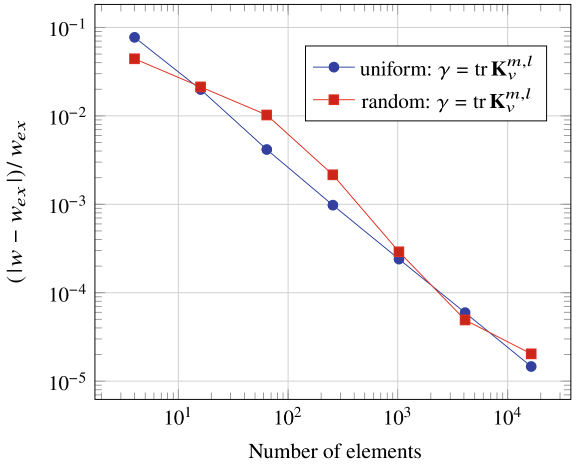


Fig. 5.6 Convergence Study: VEM element with stabilization $\gamma = \operatorname{tr} \mathbf{K}_v^{m,l}$ for the uniform and random mesh

convergence behaviour while the random mesh deviates from the straight line. Due to the randomness of the mesh there is no node in the center of the membrane and the deflection at that point was approximated by an interpolation which contributed to the deviation.

Next the results of the energy stabilized version of the virtual element are reported for three different values of the stabilization parameter $\gamma = \frac{1}{5}$, $\gamma = \frac{2}{5}$ and $\gamma = \frac{3}{5}$. These results are compared with the solution using triangular finite elements with linear shape functions denoted by T1. The mesh consists of the inscribed elements, see Fig. 5.3 and thus uses the same points as the virtual element mesh. Besides some deviations for very coarse meshes (2×2) and (4×4) all formulations depict linear convergence, see Fig. 5.7. The influence of the parameter γ is negligible. Actually, a more detailed study shows that γ can be selected from the wide range $(1 > \gamma > \frac{1}{10})$ and still yields stable solutions for all meshes.

5.4.2 *L-shaped Membrane*

A more complex mesh of an L-shaped membrane, see Fig. 5.8, with mixed boundary conditions is provided in Fig. 5.9. The problem is discretized using 8 noded Q2S elements with linear ansatz (for the definition of such elements see Appendix B). This choice leads to a discretization of 1024 virtual elements. The length parameter of the membrane is set as $L = 5$. Furthermore a pre-stress of $\sigma_0 = 5$ acts in the membrane

Summation up to $n = 101$ yields results that have a sufficient precision and were used as reference solution in the convergence analysis. The deflection in the mid point of the membrane follows by evaluation of this formula at $x = y = 0$.

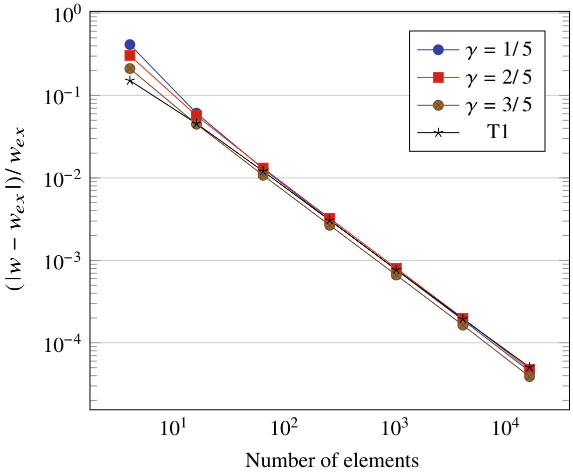


Fig. 5.7 Convergence Study: VEM element with energy stabilization $\gamma = \{\frac{1}{5}, \frac{2}{5}, \frac{3}{5}\}$ and T1 finite element for the random mesh

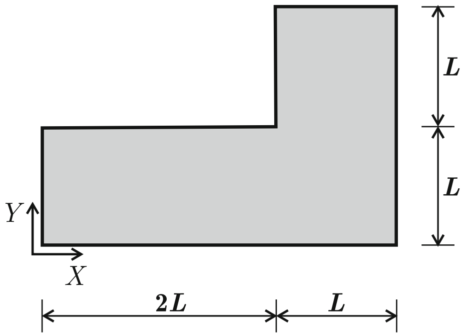


Fig. 5.8 L-shaped membrane

which is loaded by $\bar{q} = 1$. The boundary conditions are depicted by turquoise points in Fig. 5.9. At $X = 0$ and $Y = 2L$ as well as within $X = L$ and $L \leq Y \leq 2L$ a free boundary is introduced where no gradient of the deflection in normal direction is prescribed. Note that in normal direction the deflection has a horizontal tangent at the free boundary.

The numerical simulation is computed with virtual elements using the energy stabilization with $\gamma = \frac{2}{5}$. The deflection of the membrane is shown in the contour plot in Fig. 5.10. The maximal deflection of $w = 1.245$ occurs at the free boundary at $(X = 7.8\,, Y = 5)$.

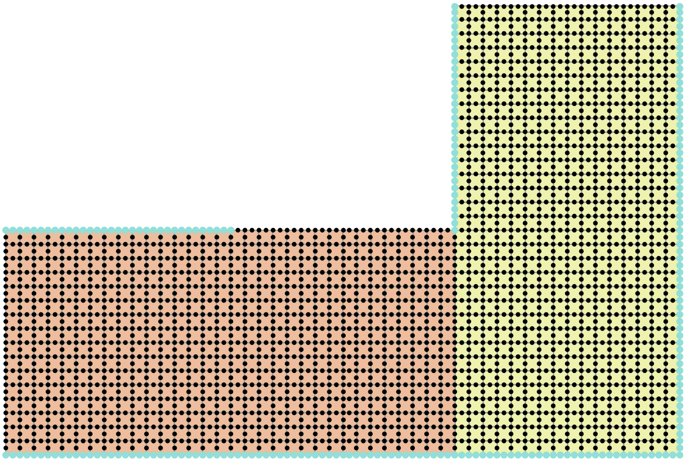


Fig. 5.9 Mesh of the L-shaped membrane using virtual elements with 8 nodes

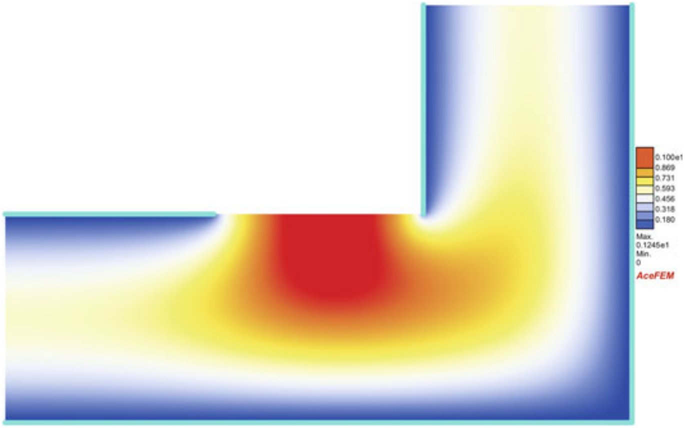


Fig. 5.10 Contour plot of the deflection w in the L-shaped membrane

References

Flanagan, D., and T. Belytschko. 1981. A uniform strain hexahedron and quadrilateral with orthogonal hour-glass control. *International Journal for Numerical Methods in Engineering* 17: 679–706.

Belytschko, T., and L.P. Bindeman. 1991. Assumed strain stabilization of the 4-node quadrilateral with 1-point quadrature for nonlinear problems. *Computer Methods in Applied Mechanics and Engineering* 88 (3): 311–340.

Reese, S., M. Kuessner, and B.D. Reddy. 1999. A new stabilization technique to avoid hourglassing in finite elasticity. *International Journal for Numerical Methods in Engineering* 44: 1617–1652.

Reese, S., and P. Wriggers. 2000. A new stabilization concept for finite elements in large deformation problems. *International Journal for Numerical Methods in Engineering* 48: 79–110.

Nadler, B., and M. Rubin. 2003. A new 3-d finite element for nonlinear elasticity using the theory of a cosserat point. *International Journal of Solids and Structures* 40: 4585–4614.

Reese, S. 2003. On a consistent hourglass stabilization technique to treat large inelastic deformations and thermo-mechanical coupling in plane strain problems. *International Journal for Numerical Methods in Engineering* 57: 1095–1127.

Mueller-Hoeppe, D.S., S. Loehnert, and P. Wriggers. 2009. A finite deformation brick element with inhomogeneous mode enhancement. *International Journal for Numerical Methods in Engineering* 78: 1164–1187.

Korelc, J., U. Solinc, and P. Wriggers. 2010. An improved EAS brick element for finite deformation. *Computational Mechanics* 46: 641–659.

Krysl, P. 2015. Mean-strain eight-node hexahedron with stabilization by energy sampling. *International Journal for Numerical Methods in Engineering* 103: 437–449.

Beirão da Veiga, L., F. Brezzi, and L. Marini. 2013. Virtual elements for linear elasticity problems. *SIAM, Journal of Numerical Analysis* 51: 794–812.

Beirão da Veiga, L., F. Brezzi, L.D. Marini, and A. Russo. The hitchhiker's guide to the virtual element method. *Mathematical Models and Methods in Applied Sciences* 24(8): 1541–1573 (2014).

Lamperti, A., M. Cremonesi, U. Perego, A. Russo, and C. Lovadina. 2023. A Hu–Washizu variational approach to self-stabilized virtual elements: 2D linear elastostatics. *Computational Mechanics* 1–21.

D' Altri, A.M., S. de Miranda, L. Patruno, and E. Sacco. 2021. An enhanced VEM formulation for plane elasticity. *Computer Methods in Applied Mechanics and Engineering* 376: 113663.

Chen, A., and N. Sukumar. 2023. Stabilization-free serendipity virtual element method for plane elasticity. *Computer Methods in Applied Mechanics and Engineering* 404: 115784.

Chen, A., and N. Sukumar. 2023. Stabilization-free virtual element method for plane elasticity. *Computer Methods in Applied Mechanics and Engineering* 404: 115784.

Beirão da Veiga, L., C. Lovadina, and D. Mora. 2015. A virtual element method for elastic and inelastic problems on polytope meshes. *Computer Methods in Applied Mechanics and Engineering* 295: 327–346.

Chi, H., L. Beirão da Veiga, and G. Paulino. 2017. Some basic formulations of the virtual element method (VEM) for finite deformations. *Computer Methods in Applied Mechanics and Engineering* 318: 148–192.

Artioli, E., L. Beirão da Veiga, C. Lovadina, and E. Sacco. 2017a. Arbitrary order 2d virtual elements for polygonal meshes: Part I, elastic problem. *Computational Mechanics* 60:355–377.

Wriggers, P., B. Reddy, W. Rust, and B. Hudobivnik. 2017. Efficient virtual element formulations for compressible and incompressible finite deformations. *Computational Mechanics* 60: 253–268.

Beirão da Veiga, L., F. Dassi, and A. Russo. 2017. High-order virtual element method on polyhedral meshes. *Computers and Mathematics with Applications* 74(5): 1110–1122.

Beirão da Veiga, L., F. Brezzi, A. Cangiani, G. Manzini, L. Marini, and A. Russo. 2013. Basic principles of virtual element methods. *Mathematical Models and Methods in Applied Sciences* 23(01): 199–214.

Gain, A.L., C. Talischi, and G.H. Paulino. 2014. On the virtual element method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes. *Computer Methods in Applied Mechanics and Engineering* 282: 132–160.

Wriggers, P., W. Rust, and B. Reddy. 2016. A virtual element method for contact. *Computational Mechanics* 58: 1039–1050.

Korelc, J., and P. Wriggers. 2016. *Automation of Finite Element Methods*. Berlin: Springer.

De Bellis, M., P. Wriggers, and B. Hudobivnik. 2019. Serendipity virtual element formulation for nonlinear elasticity. *Computers and Structures* 223: 106094.

Korelc, J. 2002. Multi-language and multi-environment generation of nonlinear finite element codes. *Engineering with Computers* 18: 312–327.

Sutton, O.J. 2017. The virtual element method in 50 lines of matlab. *Numerical Algorithms* 75 (4): 1141–1159.

Yu, Y. 2022. mVEM: A MATLAB software package for the virtual element methods. arXiv:2204.01339.

Timoshenko, S.P., and J.N. Goodier. 1970. *Theory of Elasticity*. New York: McGraw-Hill.

Chapter 6
Virtual Elements for Elasticity Problems

One of the first applications of virtual elements in engineering was related to elasticity. It had the aim to construct a general methodology that could be applied to problems involving two-dimensional linear elastic solids under general loading conditions, see Beirão da Veiga et al. (2013). After these first developments and formulations of the new method many follow up contributions appeared. An extension to three-dimensional problems was then provided in Gain et al. (2014). A paper that explained the application of the virtual element method in elasticity in more details is due to Artioli et al. (2017a). In this line of work fits the paper by Mengolini et al. (2019) who discussed implementation issues and compared virtual and finite elements. A mixed approach based on the Hellinger-Reissner functional was presented in Artioli et al. (2017c, 2020c) for linear elasticity in two dimensions and in Dassi et al. (2020) for three dimensional applications. A mixed-enhanced virtual element formulation was developed in D'Altri et al. (2021). Chen and Sukumar (2023a, b) designed a stabilization free formulation of a two-dimensional linear virtual element for linear elasticity. In the same line is the paper of Lamperti et al. (2023) who used the Hu-Washizu variational approach to design sel-stabilized virtual elements for linear elastic solids.

Non-conforming elements for linear elasticity were developed in Kwak and Park (2022) which are on one hand related to a discontinuous Galerkin approach, see Hansbo and Larson (2003), and on the other hand to the approach in Kouhia and Stenberg (1995). A non-conforming formulation of virtual elements for incompressible elasticity can be found in Zhang et al. (2019). Comparison of virtual element with finite element solutions are provided in e.g. Berbatov et al. (2021). This list is incomplete and focusses more on the engineering side of the development of virtual element methods in elasticity, however there is substantial mathematical work for two- and three-dimensional applications in compressible and incompressible elasticity, like e.g. Beirão da Veiga et al. (2013), Brezzi et al. (2014), Brenner et al. (2017), Mora and Rivera (2020), Li et al. (2022), Lovadina and Visinoni (2022) and Frerichs and Merdon (2022), which is essential for understanding convergence, stability and performance of virtual element discretizations.

P. Wriggers et al., *Virtual Element Methods in Engineering Sciences*,
https://doi.org/10.1007/978-3-031-39255-9_6

First nonlinear formulations in elasticity and plasticity where discussed in the mathematical paper by Beirão da Veiga et al. (2015) and in Chi et al. (2017), Botti et al. (2017) and applications to engineering problems with compressible and incompressible response were considered in Wriggers et al. (2017) who also introduced the energy stabilization for low order ansatz functions. After that, numerous papers followed in which virtual elements were applied to hyperelastic materials, see e.g. De Bellis et al. (2019) for a serendipity formulation with quadratic ansatz, van Huyssteen and Reddy (2020) for a variant of the energy stabilization and Wriggers et al. (2021a) for a Taylor-Hood-type mixed method for incompressible behaviour.

This chapter provides the background for the development of virtual elements for boundary value problems assuming linear and nonlinear elastic response of solids and structures. The basic theoretical background of elasticity can be found in Chap. 2. First we will discuss the two-dimensional formulation for small and finite strain and later the three-dimensional case using the ansatz spaces that were developed in Chap. 3.

6.1 Linear Elastic Response of Two-Dimensional Solids

The linear elastic response of a two-dimensional solid is governed by the kinematical relation (2.9), the momentum equation (2.22) and the constitutive equation (2.43) which can be replaced by the weak form (2.85) or the potential (2.82). Again the virtual element scheme needs the discretization of the consistency term and a stabilization term.

Here we start with the consistency term. There are basically two ways that can be followed to construct the consistency part of a virtual element:

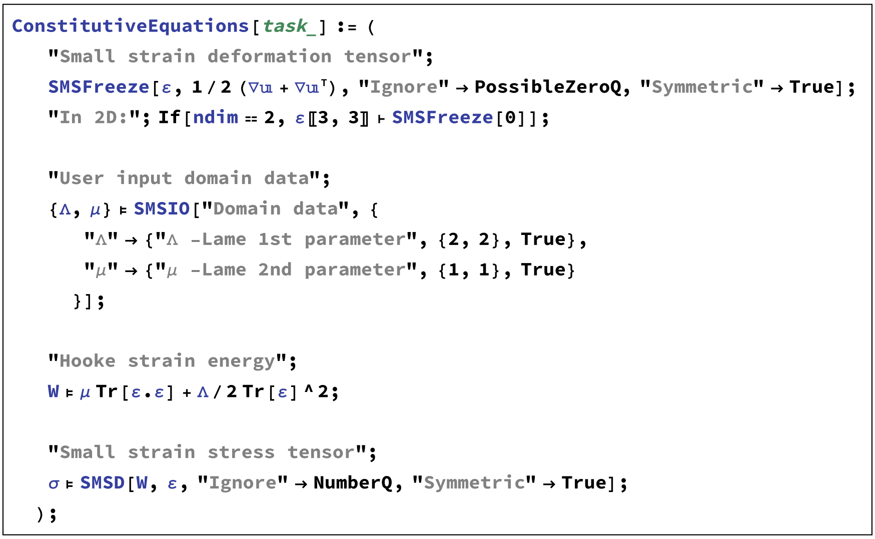
1. Start from the weak form or potential written in matrix form, here we use the potential U_c, see e.g. (5.14),

$$U_c = \frac{1}{2} \int\limits_{\Omega} \hat{\boldsymbol{\varepsilon}}_\pi^T \mathbb{C}\, \hat{\boldsymbol{\varepsilon}}_\pi \, \mathrm{d}\Omega - \int\limits_{\Omega} \mathbf{u}_\pi^T \, \bar{\boldsymbol{f}} \, \mathrm{d}\Omega - \int\limits_{\Gamma_\sigma} \mathbf{u}_\pi^T \, \bar{\boldsymbol{t}} \, \mathrm{d}\Gamma \tag{6.1}$$

   where Voigt notation, like in the stress strain relation (2.46) is used. The strain $\hat{\boldsymbol{\varepsilon}}$ is defined as

$$\hat{\boldsymbol{\varepsilon}}_\pi = \left\{ \begin{array}{c} u_{\pi x,x} \\ u_{\pi y,y} \\ u_{\pi x,y} + u_{\pi y,x} \end{array} \right\} \tag{6.2}$$

   which deviates from the tensorial strain by a factor of 2 in the shear part. This approach is in close relationship to the way classical finite elements are derived.
2. Use the weak form or potential in tensor notation. This yields with (2.82) and (2.43)



```
ConstitutiveEquations[task_] := (
   "Small strain deformation tensor";
   SMSFreeze[ε, 1/2 (∇𝕦 + ∇𝕦ᵀ), "Ignore" → PossibleZeroQ, "Symmetric" → True];
   "In 2D:"; If[ndim == 2, ε⟦3, 3⟧ ⊢ SMSFreeze[0]];

   "User input domain data";
   {Λ, μ} ⊨ SMSIO["Domain data", {
      "Λ" → {"Λ -Lame 1st parameter", {2, 2}, True},
      "μ" → {"μ -Lame 2nd parameter", {1, 1}, True}
     }];

   "Hooke strain energy";
   W ⊨ μ Tr[ε.ε] + Λ/2 Tr[ε]^2;

   "Small strain stress tensor";
   σ ⊨ SMSD[W, ε, "Ignore" → NumberQ, "Symmetric" → True];
  );
```

Fig. 6.1 Part of *AceGen* code for the elastic strain energy and stresses undergoing small deformations

$$U_c = \int\limits_{\Omega} \underbrace{\left[\frac{\Lambda}{2}\, [\mathrm{tr}(\boldsymbol{\varepsilon}_\pi)]^2 + \mu\, \mathrm{tr}(\boldsymbol{\varepsilon}_\pi\, \boldsymbol{\varepsilon}_\pi) \right]}_{W} \mathrm{d}\Omega - \int\limits_{\Omega} \boldsymbol{u}_\pi \cdot \bar{\boldsymbol{f}}\, \mathrm{d}\Omega - \int\limits_{\Gamma_\sigma} \boldsymbol{u}_\pi \cdot \bar{\boldsymbol{t}}\, \mathrm{d}\Gamma\,. \tag{6.3}$$

The related *AceGen* code[1] that defines the strain energy U_c and also the stress, see (2.45), is depicted in Fig. 6.1.

When using the first formulation then an ansatz for $\boldsymbol{u}_\pi$ and $\hat{\boldsymbol{\varepsilon}}_\pi$ has to be inserted into (6.1). In case of the tensor notation the displacement gradient $\nabla \mathbf{u}_\pi$ is used to compute $\boldsymbol{\varepsilon}_\pi$ which the is inserted in (6.3). Depending on the interpolation order we obtain different matrix forms.

6.1.1 *Consistency Term Using Voigt Notation*

The virtual element part related to the consistency term is here given for the case of linear and quadratic polynomial interpolations.

- In case of the linear interpolation one can use the results for the projection in Sect. 3.1.4. The components of the gradients needed in (6.2) can be computed

[1] All examples that are shown in this book are computed using the software tools *AceGen* and *AceFEM*, developed by Korelc (2000), see also Korelc and Wriggers (2016) and the web-site www.http://symech.fgg.uni-lj.si.

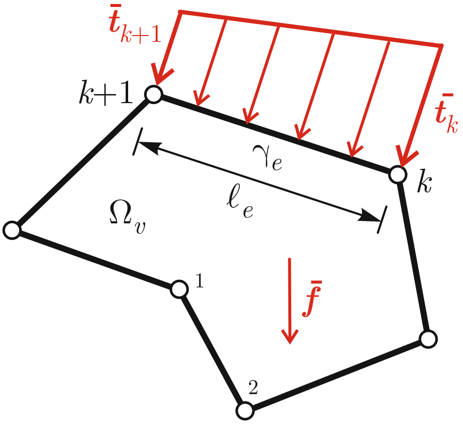


Fig. 6.2 Loading of a VEM element with linear ansatz

with the projection matrices (3.49) and (3.50) together with the unknown vector (3.48) at element level as

$$\left\{ \begin{array}{c} u_{\pi x,x} \\ u_{\pi y,y} \\ u_{\pi x,y} + u_{\pi y,x} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{L}_{x1}^T \\ \mathbf{L}_{y2}^T \\ \mathbf{L}_{x2}^T + \mathbf{L}_{y1}^T \end{array} \right\} \mathbf{u}_v = \mathbb{P}_{\nabla\varepsilon}^{(2,1)} \, \mathbf{u}_v \tag{6.4}$$

where the vectors $\mathbf{L}_{x\alpha}$ and $\mathbf{L}_{y\alpha}$ ($\alpha = 1, 2$) are constants and only depend on the element geometry. Now the strain energy part U_{cs} in the potential (6.1) can be written for an element Ω_v as

$$U_{cs}^v = \frac{\Omega_v}{2} \mathbf{u}_v^T [\mathbb{P}_{\nabla\varepsilon}^{(2,1)}]^T \mathbb{C} \, \mathbb{P}_{\nabla\varepsilon}^{(2,1)} \, \mathbf{u}_v = \frac{1}{2} \mathbf{u}_v^T \mathbf{K}_v^{(2,1)} \, \mathbf{u}_v \, . \tag{6.5}$$

where $\mathbf{K}_v^{(2,1)}$ is the element stiffness matrix.

As depicted in Fig. 6.2 the virtual element Ω_v is loaded by a volume load $\bar{\boldsymbol{f}}$ and a traction load $\bar{\boldsymbol{t}}$ at the edge γ_e. The part of the potential related to the volume load can be discretized using (3.58) which leads at element level to

$$\int\limits_{\Omega_v} \boldsymbol{u}_\pi^T \, \bar{\boldsymbol{f}} \, \mathrm{d}\Omega = \mathbf{u}_v^T \, [\mathbb{P}_u^{(2,1)}]^T \int\limits_{\Omega_v} [\mathbf{H}_u^{(2,1)}(X, Y)]^T \, \bar{\mathbf{f}} \, \mathrm{d}\Omega \, . \tag{6.6}$$

Here $\mathbf{H}_u^{(2,1)}(X, Y)$ is a linear function. If the volume load is constant or a polynomial function the integral can be evaluated over the edges of the virtual element with the formulae provided in Appendix A.

The part of the potential that includes the surface loads can be directly integrated by employing the linear ansatz (3.5) that was introduced along the edges of the virtual element. The integral can then be evaluated directly at an edge γ_e with the vertices k and $k+1$ and edge length l_e that is loaded by the traction $\bar{\mathbf{t}}$

$$\int\limits_{\Gamma_{\sigma e}} \boldsymbol{u}_h \cdot \bar{\boldsymbol{t}}\, \mathrm{d}\Gamma = \int\limits_{\gamma_e} [(1-\xi)\, \mathbf{u}_k^T + \xi\,,\, \mathbf{u}_{k+1}^T]\, \bar{\mathbf{t}}(\xi)\, \mathrm{d}\gamma \tag{6.7}$$

where ξ is the local coordinate along the edge $0 \le \xi \le 1$. For a constant or linear traction load the computation of the integral can be performed simply with a two-point Gauss-Lobatto rule leading to

$$\int\limits_{\gamma_e} [(1-\xi)\, \mathbf{u}_k^T + \xi\,,\, \mathbf{u}_{k+1}^T]\, \bar{\mathbf{t}}(\xi)\, \mathrm{d}\gamma = \{\mathbf{u}_k\,,\, \mathbf{u}_{k+1}\}^T \, \frac{l_e}{2} \begin{Bmatrix} \bar{\mathbf{t}}_k \\ \bar{\mathbf{t}}_{k+1} \end{Bmatrix} = \{\mathbf{u}_k\,,\, \mathbf{u}_{k+1}\}^T \, \mathbf{P}_e\,. \tag{6.8}$$

For more complex dependencies of the distribution of the loading, like sine-loads, Gauss integration can be employed.
Differentiation of the terms (6.5), (6.6) and (6.8) with respect to the unknown nodal displacements $\mathbf{u}_v$ yields the residual

$$\begin{aligned} \mathbf{R}_v^{(2,1)} = \frac{\partial U_c^v}{\partial \mathbf{u}_v} &= \Omega_v\, [\mathbb{P}_{\nabla\varepsilon}^{(2,1)}]_v^T \mathbb{C}\, [\mathbb{P}_{\nabla\varepsilon}^{(2,1)}]_v\, \mathbf{u}_v \\ &\quad - [\mathbb{P}_u^{(2,1)}]_v^T \int\limits_{\Omega_v} [\mathbf{H}_u^{(2,1)}(X,Y)]^T\, \bar{\mathbf{f}}\, \mathrm{d}\Omega - \sum_{e=1}^{n_t} \mathbf{P}_e \end{aligned} \tag{6.9}$$

where $\mathbf{P}_e$was defined in (6.8). It contains the traction load components $\bar{\mathbf{t}}_k$ and $\bar{\mathbf{t}}_{k+1}$ at the locations that are related to the loaded vertices at the edge γ_e. The sum goes over all edges n_t loaded by surface traction. In case of linear analysis we can assemble all virtual elements that are used to discretize the domain Ω of a problem. Then (6.9) leads to the equation for the unknown displacements

$$\mathbf{R} = \mathbf{0} \Leftrightarrow \mathbf{K}\, \mathbf{u} = \mathbf{F} \tag{6.10}$$

with

$$\mathbf{K} = \mathop{\mathsf{A}}_{v=1}^{n_v} \Omega_v\, [\mathbb{P}_{\nabla\varepsilon}^{(2,1)}]_v^T \mathbb{C}\, [\mathbb{P}_{\nabla\varepsilon}^{(2,1)}]_v \quad \text{and} \tag{6.11}$$

$$\mathbf{F} = \mathop{\mathsf{A}}_{v=1}^{n_v} [\mathbb{P}_u^{(2,1)}]_v^T \int\limits_{\Omega_v} [\mathbf{H}_u^{(2,1)}(X,Y)]^T\, \bar{\mathbf{f}}\, \mathrm{d}\Omega + \mathop{\mathsf{A}}_{e=1}^{n_v} \mathbf{P}_e \tag{6.12}$$

where n_v are the number of elements and n_t the number of all edges with traction loading.

Due to the fact that the strains in (6.4) are constant the stresses within the linear virtual element v are constant as well. They can be recovered at element level in a postprocessing step by using relation (2.46) together with (6.4)

$$\boldsymbol{\sigma}_v = \mathbb{C}\,[\mathbb{P}^{(2,1)}_{\nabla\varepsilon}]_v\,\mathbf{u}_v\,. \tag{6.13}$$

- For a virtual element formulation with a quadratic ansatz the projection operators stated in Sect. 3.1.6 can be employed. Now one has to write the strains in (6.4) in matrix form which can already be found in (3.76) and (3.77), see Remark 3.3 in Sect. 3.1.6. Based on Eq. (3.77) the potential (6.1) can be written for the virtual element Ω_v

$$\begin{aligned} U^v_{cs} &= \frac{1}{2}\mathbf{u}_v^T\,[\mathbb{P}^{(2,2)}_{\nabla}]_v^T \int\limits_{\Omega_v} [\mathbf{B}^{(2,2)}_{u\,\pi}(X,Y)]^T\,\mathbb{C}\,\mathbf{B}^{(2,2)}_{u\,\pi}(X,Y)\,\mathrm{d}\Omega\;[\mathbb{P}^{(2,2)}_{\nabla}]_v\mathbf{u}_v \\ &= \frac{1}{2}\mathbf{u}_v^T\,\mathbf{K}^{(2,2)}_v\,\mathbf{u}_v \end{aligned} \tag{6.14}$$

which includes in the integral only polynomials up to second order. The element stiffness matrix $\mathbf{K}^{(2,2)}_v$ can be obtained by integration over the edges, see Appendix A. Figure 6.3 shows the volume loading by $\bar{\boldsymbol{f}}$ and the loading by surface traction $\bar{\boldsymbol{t}}$ at the edge γ_e. The volume loading term is computed by inserting the projected displacement field from (3.84)

$$\int\limits_{\Omega_v} \boldsymbol{u}_\pi^T\,\bar{\boldsymbol{f}}\,\mathrm{d}\Omega = \mathbf{u}_v^T \int\limits_{\Omega_v} [\mathbb{P}^{(2,2)}_u(X,Y)]^T\,\bar{\boldsymbol{f}}\,\mathrm{d}\Omega\,. \tag{6.15}$$

Again the integral can be evaluated for any polynomial, see Appendix A.
The term related to the surface load follows in the same way as the term in the linear case, see (6.7). Here again we can evaluate the integral over the edge γ_e

$$\int\limits_{\Gamma_{\sigma e}} \boldsymbol{u}_h \cdot \bar{\boldsymbol{t}}\,\mathrm{d}\Gamma = \int\limits_{\gamma_e} \left[\sum_{i=1}^{3} M_i(\xi)\,\mathbf{u}_i^T\right] \bar{\mathbf{t}}(\xi)\,\mathrm{d}\gamma\,. \tag{6.16}$$

Using the Gauss-Lobatto rule for the straight edge yields

$$\int\limits_{\gamma_e} \left[\sum_{i=1}^{3} M_i(\xi)\,\mathbf{u}_i^T\right] \bar{\mathbf{t}}(\xi)\,\mathrm{d}\gamma = \frac{l_e}{6}[\mathbf{u}_{2k-1}^T\,\bar{\mathbf{t}}_{2k-1} + 4\mathbf{u}_{2k}^T\,\bar{\mathbf{t}}_{2k} + \mathbf{u}_{2k+1}^T\,\bar{\mathbf{t}}_{2k+1}] \tag{6.17}$$

which is exact for a constant and a linear distribution of the traction loads.
Residual and the equation system for the unknown displacements follow in the same way as (6.9) and (6.10).

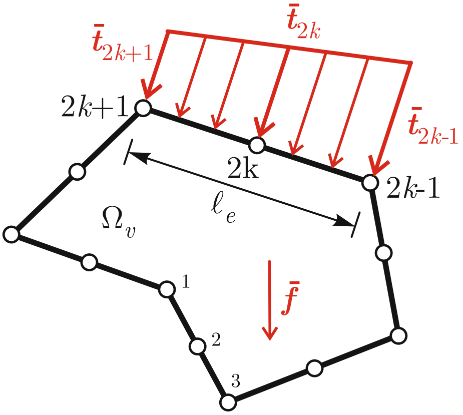


Fig. 6.3 Loading of a VEM element with quadratic ansatz

6.1.2 Consistency Term Using Tensor Notation

There are different ways to formulate the virtual elements. The choice which formulation is most convenient depends on the software tool that is employed to obtain the virtual element residual and stiffness matrix. For tools that are able to automatically differentiate weak forms or potentials like *AceGen*, see e.g. Korelc and Wriggers (2016), it is advantageous to stay within the tensor notation of continuum mechanics. As an example, the virtual element with linear ansatz is discussed.

The projected gradient $\nabla \boldsymbol{u}_\pi$ was already stated in (3.61) in tensor notation for the linear ansatz. This (2×2) tensor can be more explicitly written as

$$\nabla \boldsymbol{u}_\pi = \frac{1}{\Omega_v} \sum_{e=1}^{n_E} \begin{bmatrix} (u_{xk} + u_{xk+1})(Y_1 - Y_2)_e & (u_{xk} + u_{xk+1})(X_2 - X_1)_e \\ (u_{yk} + u_{yk+1})(Y_1 - Y_2)_e & (u_{yk} + u_{yk+1})(X_2 - X_1)_e \end{bmatrix} \tag{6.18}$$

where k and $k+1$ denote the vertices of an edge e and n_E is the number of edges of the virtual element with area Ω_v. Now the strain is simply given by, see also the code in Fig. 6.1,

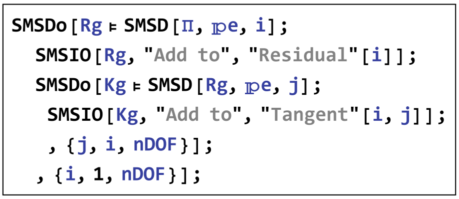
$$\boldsymbol{\varepsilon}_\pi = \frac{1}{2}(\nabla \boldsymbol{u}_\pi + \nabla \boldsymbol{u}_\pi^T)\,. \tag{6.19}$$

This can be inserted in the potential (6.3) and evaluated for an element Ω_v, note that (6.18) is constant with respect to X and Y,

$$U_{cs}^v = \int\limits_{\Omega_v} \left[\frac{\Lambda}{2}\, [\mathrm{tr}(\boldsymbol{\varepsilon}_\pi)]^2 + \mu\, \mathrm{tr}(\boldsymbol{\varepsilon}_\pi\, \boldsymbol{\varepsilon}_\pi) \right] \mathrm{d}\Omega = \Omega_v \left[\frac{\Lambda}{2}\, [\mathrm{tr}(\boldsymbol{\varepsilon}_\pi)]^2 + \mu\, \mathrm{tr}(\boldsymbol{\varepsilon}_\pi\, \boldsymbol{\varepsilon}_\pi) \right]. \tag{6.20}$$

The residual related to the strain energy U_{cs}^v is obtained for the virtual element by simply differenciating U_{cs}^v with respect to the unknowns $\mathbf{u}_k$ at the vertices $(1 \leq k \leq n_V)$. Thus the residual follows with (3.48) as

Fig. 6.4 Part of *AceGen* code for the derivation of the residual and stiffness matrix



```
SMSDo[Rg ⊨ SMSD[Π, 𝕡e, i];
  SMSIO[Rg, "Add to", "Residual"[i]];
  SMSDo[Kg ⊨ SMSD[Rg, 𝕡e, j];
   SMSIO[Kg, "Add to", "Tangent"[i, j]];
   , {j, i, nDOF}];
  , {i, 1, nDOF}];
```

$$\mathbf{R}_v^{(2,1)} = \frac{\partial U_{cs}^v}{\partial \mathbf{u}_v} = \begin{Bmatrix} \frac{\partial U_{cs}^v}{\partial \mathbf{u}_1} \\ \frac{\partial U_{cs}^v}{\partial \mathbf{u}_2} \\ \dots \\ \frac{\partial U_{cs}^v}{\partial \mathbf{u}_{n_V}} \end{Bmatrix} \tag{6.21}$$

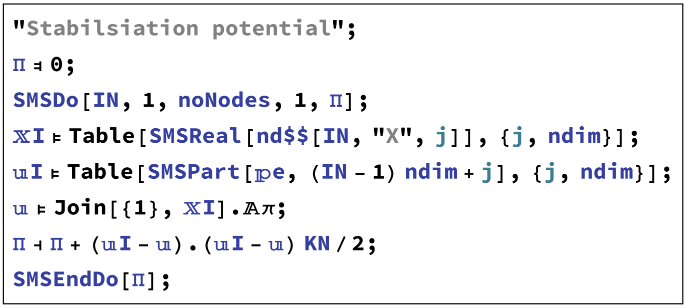
with $\mathbf{u}_k^T = \{u_{xk}\,, u_{yk}\}$. Since the sum in (6.18) includes all vertices this differentiation will automatically pick up the associated displacements and allocate these correctly in the residual vector $\mathbf{R}_v^{(2,1)}$. The element stiffness matrix follows then by a second differentiation

$$\mathbf{K}_v^{(2,1)} = \frac{\partial \mathbf{R}_v^{(2,1)}}{\partial \mathbf{u}_v}\,. \tag{6.22}$$

The *AceGen* code, related to the computation of residual and stiffness matrix, is illustrated in Fig. 6.4. This code is the third ingredient for the generation of a virtual element with linear shape functions. The first part that defines the projection onto the polynomial space and thus the ansatz $\boldsymbol{u}_\pi$ is provided in Fig. 3.5 in Sect. 3.1.5. This ansatz is then, in the second part of the code, inserted in the strain measure $\boldsymbol{\varepsilon}_\pi$ and the potential as shown in Fig. 6.1. Based on this potential Fig. 6.4 shows the derivation of $\mathbf{R}_v^{(2,1)}$ and $\mathbf{K}_v^{(2,1)}$.

6.1.3 Stabilization

The consistency part for linear elasticity yields a stiffness matrix that has only 3 non-zero eigenvalues, reflecting the constant strain state. Thus for a virtual element with more than 3 nodes the consistency part has zero energy modes, like a one-point integrated finite element. To avoid these zero energy modes the element has to be stabilized. Of course, this observation applies also to elements with higher order ansatz and was discussed in general in Sect. 5.2. Here we will employ both approaches for stabilzation of virtual elements, the discrete bi-linear form (5.19) and the energy stabilization (5.20).



```
"Stabilsiation potential";
ℿ ⊣ 0;
SMSDo[IN, 1, noNodes, 1, ℿ];
𝕩I ⊨ Table[SMSReal[nd$$[IN, "X", j]], {j, ndim}];
𝕦I ⊨ Table[SMSPart[𝕡e, (IN - 1) ndim + j], {j, ndim}];
𝕦 ⊨ Join[{1}, 𝕩I].𝔸π;
ℿ ⊣ ℿ + (𝕦I - 𝕦).(𝕦I - 𝕦) KN / 2;
SMSEndDo[ℿ];
```

Fig. 6.5 Part of *AceGen* code for the derivation of the bi-linear form of the stabilization energy

Discrete bi-linear form. The bi-linear form of the stabilization energy[2] is used

$$U_{stab} = \frac{\gamma(\mathbf{u}_v)}{2} \sum_{k=1}^{n_V} [\mathbf{u}_k - \boldsymbol{u}_\pi(\mathbf{X}_k)] \cdot [\mathbf{u}_k - \boldsymbol{u}_\pi(\mathbf{X}_k)] \tag{6.23}$$

with $\gamma(\mathbf{u}_v) = \operatorname{tr} \mathbf{K}_v$. The nodal displacements $\mathbf{u}_k$ are given at the vertices of the virtual element and $\boldsymbol{u}_\pi(\mathbf{X}_k)$ is the projected displacement field, evaluated at the vertices $\mathbf{X}_k$. The essential part of the *AceGen* code, related to this stabilization method is shown in Fig. 6.5 which is the fourth code part needed to generate a virtual element. For the linear ansatz we can formulate (6.23) more explicit as

$$U_{stab} = \frac{||\mathbf{K}_D||}{2\sqrt{N}} \sum_{k=1}^{n_V} \mathbf{u}_v^T \left(\mathbf{I}^T - [\mathbb{P}_u^{(2,1)}]^T [\mathbf{H}_u^{(2,1)}(\mathbf{X}_k)]^T\right) \left(\mathbf{I} - \mathbf{H}_u^{(2,1)}(\mathbf{X}_k)\, \mathbb{P}_u^{(2,1)}\right) \mathbf{u}_v \tag{6.24}$$

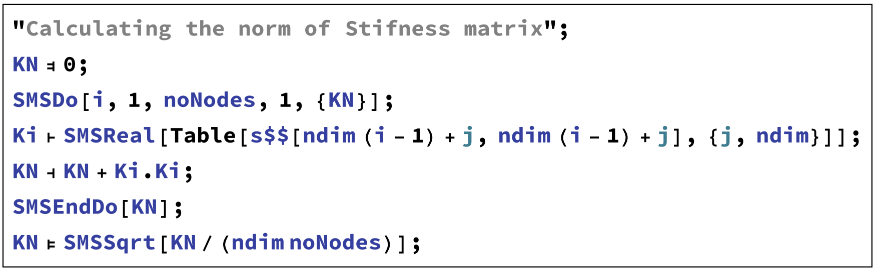
where the (2×6) matrix $\mathbf{H}(\mathbf{X})$ and the $(6 \times 2n_V)$ matrix $\mathbb{P}_u^{(2,1)}$ are known from (3.58) and $N = 2\,n_v$ is the total number of unknowns used to scale the norm of the diagonal matrix $\mathbf{K}_D$ of $\mathbf{K}_v$, see also (5.26) and the code in Fig. 6.6. The matrix $\mathbf{I}$ is given by

$$\mathbf{I} = \begin{bmatrix} 1\;0\;1\;0\;\ldots\;1\;0 \\ 0\;1\;0\;1\;\ldots\;0\;1 \end{bmatrix}$$

which has the size $(2 \times 2n_V)$. The symbolic code for the computation of $\operatorname{tr} \mathbf{K}_v$ in (6.24) is provided in Fig. 6.6.

In case of the quadratic ansatz function the stabilization energy is the same as in (6.23) only the explicit form changes with (3.78) and (3.84) to

[2] In a way this stabilization potenrial can be viewed as a penalty term that tries to minimize the distance between the solution at the vertices and the projected displacement field $\mathbf{u}_\pi$.



```
"Calculating the norm of Stifness matrix";
KN ⊣ 0;
SMSDo[i, 1, noNodes, 1, {KN}];
Ki ⊢ SMSReal[Table[s$$[ndim (i - 1) + j, ndim (i - 1) + j], {j, ndim}]];
KN ⊣ KN + Ki.Ki;
SMSEndDo[KN];
KN ⊨ SMSSqrt[KN / (ndim noNodes)];
```

Fig. 6.6 Part of *AceGen* code for the computation of the scaled norm $\frac{1}{\sqrt{N}}||\mathbf{K}_D||$

$$U^q_{stab} = \frac{||\mathbf{K}_D||}{2\sqrt{N}} \sum_{k=1}^{n_V} \mathbf{u}_v^T \left[\mathbf{I}^T - \mathbb{P}_u^{(2,2)}(\mathbf{X}_k)^T \right] \left[\mathbf{I} - \mathbb{P}_u^{(2,2)}(\mathbf{X}_k) \right] \mathbf{u}_v \tag{6.25}$$

where $\mathbf{I}$ and $\mathbb{P}_u^{(2,2)}$ are now a $(2 \times 4n_V)$ matrices.

Energy stabilization. For this stabilization one has to introduce a stabilization energy $\widehat{U}$, see (5.30), which is added to the consistency part $U_c(\boldsymbol{u}_\pi)$, see e.g. Wriggers et al. (2017). This yields the total potential energy

$$U(\boldsymbol{u}_h) = U_c(\boldsymbol{u}_\pi) + \widehat{U}(\boldsymbol{u}_h) - \widehat{U}(\boldsymbol{u}_\pi)\,, \tag{6.26}$$

where $\widehat{U}$ is related to a selected strain energy. The terms involving $\boldsymbol{u}_\pi$ can be integrated in the same way as the consistency term, see e.g. (6.20).

For the linear elastic case the form of the potential strain energy $\widehat{U}$ is chosen to be the same as for U_{cs} leading for a virtual element Ω_v to

$$\widehat{U}(\boldsymbol{u}) = \int\limits_{\Omega_v} \left[\frac{\widehat{\Lambda}}{2}\, [\mathrm{tr}(\boldsymbol{\varepsilon}(\boldsymbol{u}))]^2 + \widehat{\mu}\, \mathrm{tr}[\boldsymbol{\varepsilon}(\boldsymbol{u})\, \boldsymbol{\varepsilon}(\boldsymbol{u})] \right] \mathrm{d}\Omega\,. \tag{6.27}$$

where $\boldsymbol{u}$ can stand for either $\boldsymbol{u}_\pi$ or for $\boldsymbol{u}_h$. In case of elasticity it makes sense to use extra knowledge for the determination of the constitutive parameters $\widehat{\Lambda}$ and $\widehat{\mu}$. The idea is to find a parameter set that will enhance the bending behaviour for virtual elements with linear ansatz which is discussed below. Once the parameters are determined, the part $\widehat{U}(\boldsymbol{u}_\pi)$ is easily computed using the same evaluation as for $U_c(\boldsymbol{u}_\pi)$.

Hence it remains to compute the term involving $\widehat{U}(\boldsymbol{u}_h)$. This is achieved by constructing an internal mesh in the element with nodal points being equivalent to the nodal points of the original element to approximate the displacement field $\boldsymbol{u}_h$ in (5.30) by an inscribed triangular finite element mesh, see Fig. 6.7. It consists of n_{int} linear three-noded triangles that are connected to the nodes of the virtual element. This mesh can then be employed to compute the stabilization energy. The specific

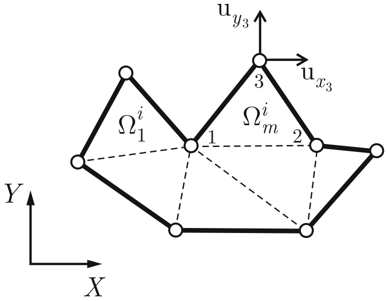


Fig. 6.7 Internal triangular mesh of a virtual element Ω_v

choice of the mesh does not introduce extra degrees of freedom since it just uses the vertices describing the virtual element. The stabilization energy has now to be computed as a sum over all internal triangles

$$\widehat{U}(\boldsymbol{u}_h) = \sum_{m=1}^{n_{int}} \int\limits_{\Omega_m^i} \left[\frac{\widehat{\Lambda}}{2} \, [\mathrm{tr}(\boldsymbol{\varepsilon}(\boldsymbol{u}_h^m))]^2 + \widehat{\mu} \, \mathrm{tr}[\boldsymbol{\varepsilon}(\boldsymbol{u}_h^m) \, \boldsymbol{\varepsilon}(\boldsymbol{u}_h^m)] \right] \mathrm{d}\Omega \,. \tag{6.28}$$

A linear ansatz is formulated to approximate the displacement field within each triangle Ω_m^i of the inscribed mesh, denoted by $\boldsymbol{u}_h^m$. The gradient $\nabla \boldsymbol{u}_h^m$ and with this the strain $\boldsymbol{\varepsilon}(\boldsymbol{u}_h^m)$ can be easily computed using an isoparametric formulation for the three noded triangle

$$\boldsymbol{u}_h^m = \sum_{I=1}^{3} N_I(\xi \, , \eta) \, \mathbf{u}_I \, , \tag{6.29}$$

with

$$N_1 = 1 - \xi - \eta \, , \quad N_2 = \xi \quad \text{and } N_3 = \eta \, .$$

Now the stabilization energy $\widehat{U}(\mathbf{u}_h)$ can be computed for the virtual element by employing this ansatz in (6.28). Note that the gradient $\nabla \boldsymbol{u}_h^m$ is constant over each inscribed element. Hence the strain $\boldsymbol{\varepsilon}$ is constant as well within each element Ω_m^i that is needed to evaluate (6.28).

The displacement gradient $\nabla \boldsymbol{u}_h^m$ can be computed using a direct evaluation of the ansatz functions for a triangular element Ω_m^i which yields with

$$\begin{Bmatrix} u_x \\ u_y \end{Bmatrix}_m = \begin{Bmatrix} d_1 \\ d_4 \end{Bmatrix}_m + \begin{Bmatrix} d_2 \\ d_5 \end{Bmatrix}_m X_m + \begin{Bmatrix} d_3 \\ d_6 \end{Bmatrix}_m Y_m \tag{6.30}$$

an explicit expression for the displacement gradient in the reference configuration

$$\nabla \, \boldsymbol{u}_h^m = \begin{bmatrix} d_2 \; d_3 \\ d_5 \; d_6 \end{bmatrix}_m \, . \tag{6.31}$$

The values d_2, d_3, d_5 and d_6 depend on the nodal displacements of the triangle. The strain tensor in each sub triangle Ω^i_m follows now as

$$\boldsymbol{\varepsilon}^m_h = \frac{1}{2}(\nabla\, \boldsymbol{u}^m_h + \nabla^T \boldsymbol{u}^m_h)\,. \tag{6.32}$$

Hence the stabilization energy can be computed using (6.28)

$$\widehat{U}(\mathbf{u}_h) = \sum_{m=1}^{n_{int}} \int\limits_{\Omega^i_m} \left[\frac{\widehat{\Lambda}}{2}\, [\mathrm{tr}(\boldsymbol{\varepsilon}^m_h)]^2 + \widehat{\mu}\, \mathrm{tr}(\boldsymbol{\varepsilon}^m_h\, \boldsymbol{\varepsilon}^m_h) \right] \mathrm{d}\Omega \tag{6.33}$$

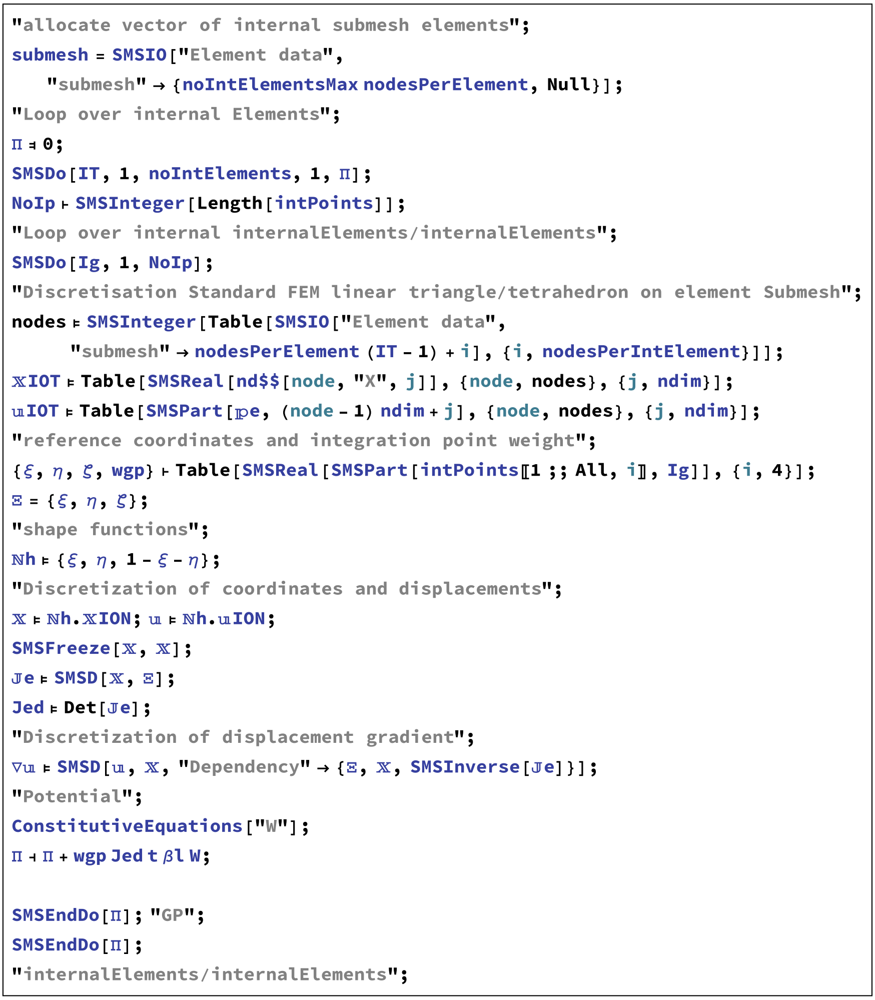
which is a linear function of the nodal displacements related to the interior triangular mesh. The symbolic code for the computation of the energy stabilization is depicted in Fig. 6.8 which consists of a loop over all internal elements and another loop regarding the Gauss integration of each internal element where a one point integration is sufficient, see also (6.31).

Next we have to find a procedure to select the constitutive parameters $\widehat{\Lambda}$ and $\widehat{\mu}$. There exist different approaches to choose the parameters in order to not only stabilize the element but also to obtain optimal response for specific applications. Since solids are often subjected to bending deformations it is preferable to have a parameter choice that enhances bending performance, especially for elements with linear ansatz functions. Enhanced bending properties can be obtained by integrating analytical solutions of specific deformations modes into the stabilization, see Nadler and Rubin (2003) or using enhanced modes, see Mueller-Hoeppe et al. (2009), or simplified variants, see e.g. Krysl (2016), who suggested a procedure that compares the strain energy of a bending mode for a hexahedra with the analytical solution. This yields the Lamé parameters $\widehat{\Lambda}$ and $\widehat{\mu}$ for the stabilization energy $\widehat{U}$ that depend on the geometrical shape of the elements. Since the above mentioned procedures are usually based on rectangular or cuboid element shapes they are not directly applicable to arbitrary virtual elements. It was suggested in Wriggers et al. (2017) to derive the parameters from the basic geometric data of the virtual element, as shown in Fig. 6.9.[3]

For each virtual element the Lamé parameters Λ and μ are converted to Young's modulus E and Poisson ratio ν. The Young's modulus for the stabilization potential can now be obtained based on the length to height ratio of the individual element by comparing the bending energy of a rectangular block with that of a beam in order to enhance the bending behaviour of the element, see e.g. Krysl (2015a). This yields the correction

$$\widehat{E} = \frac{\alpha}{1+\alpha}\, E = \beta\, E \tag{6.34}$$

[3] van Huyssteen and Reddy (2020) suggested to use instead of the outer and inner circles in Fig. 6.9 ellipses which are closer to the actual shape of the virtual element. However, as will be seen in the numerical simulations, the energy stabilzation of the virtual element method is not very sensitive with respect to the stabilization parameter and hence we choose the simpler computation described in Fig. 6.9.



```
"allocate vector of internal submesh elements";
submesh = SMSIO["Element data",
   "submesh" → {noIntElementsMax nodesPerElement, Null}];
"Loop over internal Elements";
Π ⫤ 0;
SMSDo[IT, 1, noIntElements, 1, Π];
NoIp ⊢ SMSInteger[Length[intPoints]];
"Loop over internal internalElements/internalElements";
SMSDo[Ig, 1, NoIp];
"Discretisation Standard FEM linear triangle/tetrahedron on element Submesh";
nodes ⊨ SMSInteger[Table[SMSIO["Element data",
     "submesh" → nodesPerElement (IT - 1) + i], {i, nodesPerIntElement}]];
𝕏IOT ⊨ Table[SMSReal[nd$$[node, "X", j]], {node, nodes}, {j, ndim}];
𝕦IOT ⊨ Table[SMSPart[𝕡e, (node - 1) ndim + j], {node, nodes}, {j, ndim}];
"reference coordinates and integration point weight";
{ξ, η, ζ, wgp} ⊢ Table[SMSReal[SMSPart[intPoints⟦1 ;; All, i⟧, Ig]], {i, 4}];
Ξ = {ξ, η, ζ};
"shape functions";
ℕh ⊨ {ξ, η, 1 - ξ - η};
"Discretization of coordinates and displacements";
𝕏 ⊨ ℕh.𝕏ION; 𝕦 ⊨ ℕh.𝕦ION;
SMSFreeze[𝕏, 𝕏];
𝕁e ⊨ SMSD[𝕏, Ξ];
Jed ⊨ Det[𝕁e];
"Discretization of displacement gradient";
∇𝕦 ⊨ SMSD[𝕦, 𝕏, "Dependency" → {Ξ, 𝕏, SMSInverse[𝕁e]}];
"Potential";
ConstitutiveEquations["W"];
Π ⊣ Π + wgp Jed t βl W;

SMSEndDo[Π]; "GP";
SMSEndDo[Π];
"internalElements/internalElements";
```

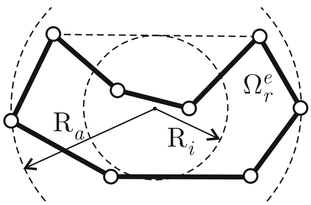
Fig. 6.8 Part of *AceGen* code for the computation of the energy stabilization

for the Young's modulus. The Poisson ratio is not changed since it does not influence the convergence behaviour leading to $\widehat{\nu} = \nu$. Only for $\nu > 0.3$ Krysl (2015a) suggests to use $\widehat{\nu} = \frac{1}{2}(\nu + 0.3)$ to avoid locking for the case of incompressibility when $\nu \rightarrow 0.5$.

The factor α depends on the height to length ratio of the virtual element. Due to the arbitrary shape this ratio depends on the inner and outer radii, R_i^2 , R_a^2 respectively, see Fig. 6.9

$$\alpha = 2\sqrt{2}\,(1+\nu)\,\frac{R_i^2}{R_a^2 - R_i^2}\,. \tag{6.35}$$

Fig. 6.9 Inner and outer radius of a virtual element



The inner radius is computed by using the distance from the baricenter to the convex hull of the virtual element while the outer radius is defined by the maximum distance between nodes defining the virtual element. The computation of α takes into account that for a square shaped element the ratio of length to height is 1.

Once $\widehat{E}$ and $\widehat{\nu}$ are known the Lamé parameters for the stabilization energy follow from

$$\widehat{\Lambda} = \frac{\widehat{E}\,\widehat{\nu}}{(1+\widehat{\nu})(1-2\widehat{\nu})} \qquad \widehat{\mu} = \frac{\widehat{E}}{2(1+\widehat{\nu})}\,. \tag{6.36}$$

Remark 6.1 In linear elasticity we have a very special case where the stabilization energy U_{stab} has the same form as the energy U_c that describes the physics in the consistency part. Assuming furthermore that the Poisson ratio ν is the same in both energies, $\hat{\nu} = \nu$, one can define a relation between the Youngs's moduli in the consistency and stabilization part $\widehat{E} = \beta\, E$, see (6.34). It follows that $\widehat{U}(\boldsymbol{u}_\pi) = \beta\, U_c$. This leads with (5.30) to

$$U = U_c + \beta\,\widehat{U} = (1-\beta)\,U(\boldsymbol{u}_\pi) + \beta\,\widehat{U}(\boldsymbol{u}_h)\,. \tag{6.37}$$

In this special case we have a sum of parts stemming from the virtual element method (consistency part $U(\boldsymbol{u}_\pi)$) and from the finite element method (stabilization energy $\widehat{U}(\boldsymbol{u}_h)$ which is computed in the code depicted in Fig. 6.8). Thus the energy stabilization leads to a mixture of virtual and finite element discretization with the extreme cases

- $\beta = 1$: pure finite element solution, using just the internal mesh. It can be shown that these results are not as good as the ones using a mixture of VEM and FEM. This was investigated for the nonlinear case, see the results in "VEM versus FEM" in Sect. 6.2.5.1.
- $\beta = 0$: pure solution based on the consistency term. This however yields a rank deficient tangent matrix and thus solutions cannot be computed.

Hence the coupling of both energies is necessary to solve a problem in solid mechanics correctly.

6.1.4 Numerical Example

Many test cases are known for elasticity problems in solid mechanics that can be applied to evaluate the quality of a discretization scheme. An often employed test, well known from finite element analysis, is the Cook's membrane problem, see e.g. Cook et al. (1989). It includes in-plane and bending behaviour and is used here to compare the virtual element formulation with standard finite elements for linear elasticity. Different discretizations are employed which are defined in Appendix B.

A tapered beam is subjected to an end load $q_0 = 1$ N/mm, see Fig. 6.10, which also includes the geometrical data. The Young's modulus is chosen to be $E = 100$ N/mm^2 and the Poisson ratio is $\nu = 0.3$.

Two different Voronoi meshes with uniformly and spatially distributed element sizes and regular meshes with four and eight nodes per virtual element were employed within the numerical solution. The deformed configurations for different Voronoi meshes are depicted in Fig. 6.11 together with the stress contours of the von Mises stress σ_{VM}. The mesh on the left hand side has a uniform element size while on the right hand side a mesh with different, spatially distributed sizes of the virtual elements was used. The Cook's membrane problem has a weak singularity at the optuse-angled corner ($X = 0\,,\, Y = 44$ mm) which leads to a high normal stress σ_{VM} at this point.

Due to this singularity one cannot expect more than linear asymptotic convergence in the displacement field when using a regular mesh. This is revealed by the convergence study depicted in Fig. 6.12. Here the results of linear triangular (FE-T1) and quadrilateral (FE-Q1) as well as a quadratic serendipity (FE-Q2S) finite elements are compared with the results of virtual elements with linear ansatz having four

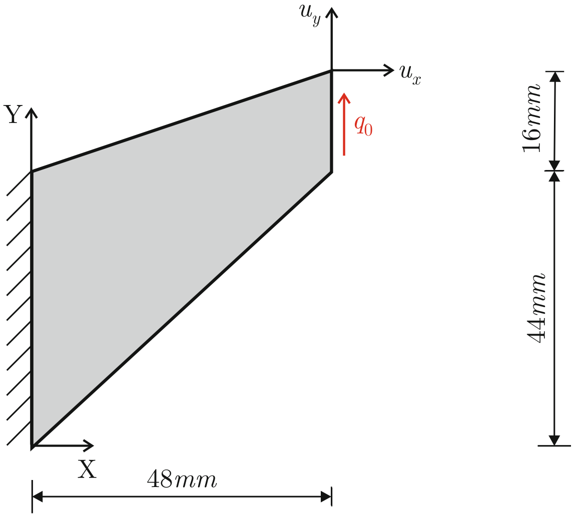


Fig. 6.10 Cook's membrane problem

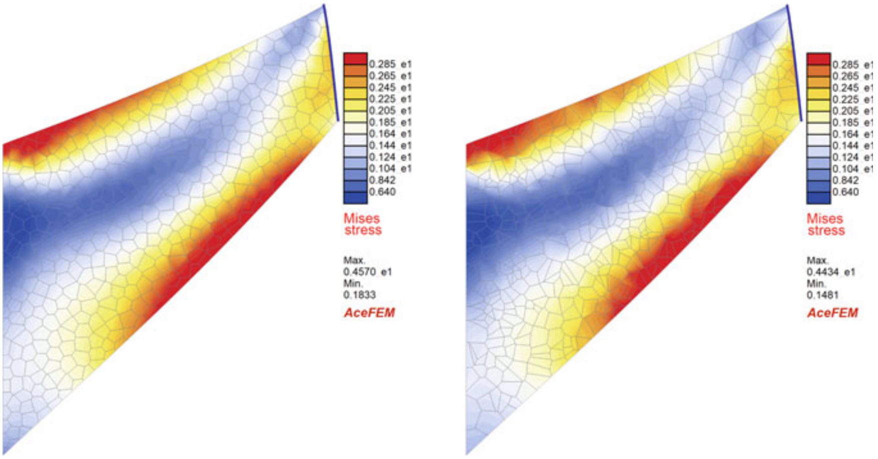


Fig. 6.11 Stress distribution of σ_{VM} using two different Voronoi meshes

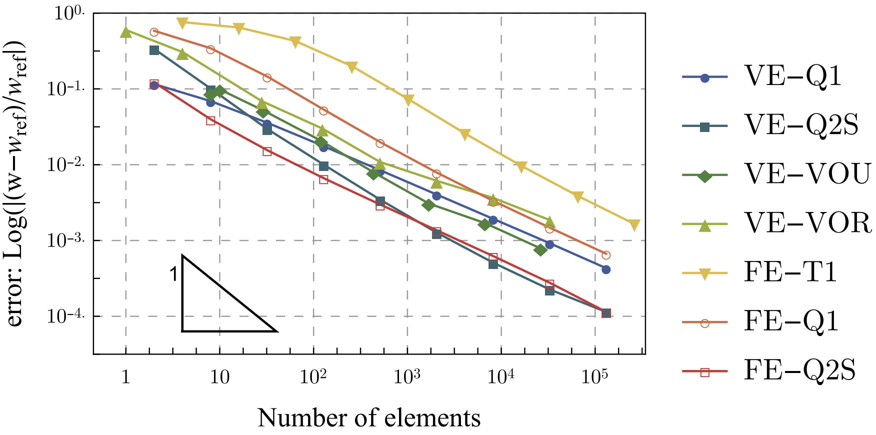


Fig. 6.12 Convergence Study: VEM element with internal triangular stabilization for uniform meshes

nodes (VE-Q1), eight nodes (VE-Q2S) and using Voronoi meshes with uniformly (VE-VOU) and randomly (VE-VOR) distributed element sizes.

Figure 6.12 depicts the same convergence behaviour for all elements. An overkill solution with linear Pian-Sumihara finite elements, see e.g. Pian and Sumihara (1984), with 19 Mio unknowns is used as reference solution, which yields a vertical displacement $u_{Y\,ref} = 3.5111916$ mm at the middle node $(X = 48\,, Y = 52)$ mm of the right side, see Fig. 6.10. We observe a sub linear convergence for finite and virtual elements with basically the same convergence rate. Interestingly, the linear virtual element (VE-Q2S) has almost the same convergence behaviour as the quadratic serendiptiy finite element (FE-Q2S). Furthermore, the graph illustrates that

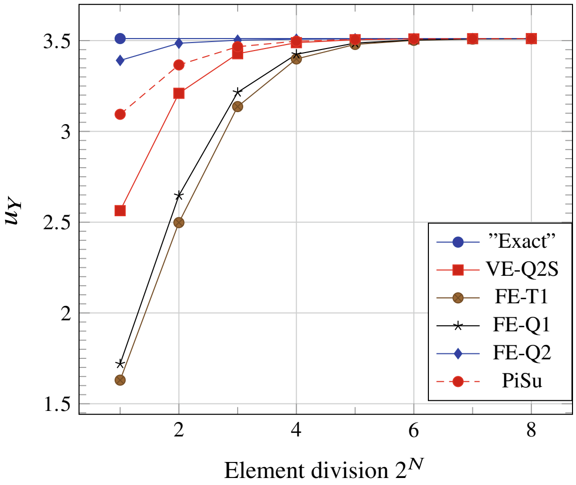


Fig. 6.13 Convergence Study: VE-Q2S, T1, Q1, Q2 and Pian-Sumihara (PiSu) elements

the coarse mesh accuracy of the linear virtual elements for all mesh types is superior to that one of the linear finite elements. In detail it can be seen in Fig. 6.12 that the same accuracy of 10^{-2} is achieved for a mesh of about 8 10^2 unknowns for VE-Q1 as with a mesh of about 2.5 10^3 unknowns using FE-Q1 finite elements. Thus, virtual elements seem more efficient when compared to standard finite elements.

It is well know that there exist special finite element formulations, like enhanced finite elements, see e.g. Simo and Rifai (1990), or the mixed Pian-Sumihara element that have a far better coarse mesh accuracy and very good bending behaviour, but it is not easy to compare these formulations with a pure displacement based virtual element since associated mixed formulations do not exist so far for virtual elements. Nevertheless a comparion is made and the results are shown in Fig. 6.13 which demonstrates the good convergence behaviour of the linear virtual element (VE-Q2S). For comparison, the results of four different finite elements are depicted. These are the linear (FE-T1, FE-Q1) and quadratic (FE-Q2) isoparametric elements as well as the mixed Pian-Sumihara (PiSu) element, see Pian and Sumihara (1984). The latter two elements exhibit excellent bending behaviour, that is related for the element FE-Q2 to the quadratic ansatz and for the Pian-Sumihara element to the special mixed formulation. Figure 6.13 reveals that the VE-Q2S element is the second best linear element behind the mixed Pian-Sumihara finite element, while both cannot reach the coarse grid accuracy of the FE-Q2 element for predicting the displacement u_Y at the mid node $(X = 48\,, Y = 52)$ mm. The reason for that behaviour is related to the Cook's membrane problem which results basically in a bending response of a thick tapered beam and FE-Q2 has superior bending behaviour. Since the construction of the mixed Pian-Sumihara finite element and the stabilization of the VE-Q2S take bending effects into account, for the latter see (6.34), both outperfom the FE-Q1 and FE-T1 finite elements. We note that he quadratic ansatz of FE-Q2 includes the bending modes used in the construction of the PiSu and VE-Q2S elements.

The complete *AceGen* code related to the numerical examples can be obtained form the associated software package, see Korelc (2023), using the function *AceShare*. However it is also possible to use other implementations, like the one provided in Yu (2022) for the *MATLAB* environment.

6.2 Finite Strain: Compressible Elasticity

The formulation of virtual elements for finite deformations of hyperelastic materials is straightforward. We can apply the projections from Sects. 3.1.2 and 3.2.2 with no changes. Again, the computation of the displacement gradient $\nabla \boldsymbol{u}_\pi$ which is needed for the consistency part of the elements can be performed in a straightforward manner as in the linear case. An especially simple formulation follows for low order virtual elements since the consistency part then yields a constant strain and stress field at element level. Higher order formulations need a special treatment of the volume integrals related to the nonlinear strain energy. We will start with a virtual element formulation with a linear ansatz. As in the previous section, we provide details of the source code used together with *AceGen* to generate the associated virtual element.

6.2.1 Consistency Term

The consistency part is based on the potential U that was formulated in (2.82). Together with the strain energy function (2.55) the discrete form of the potential yields

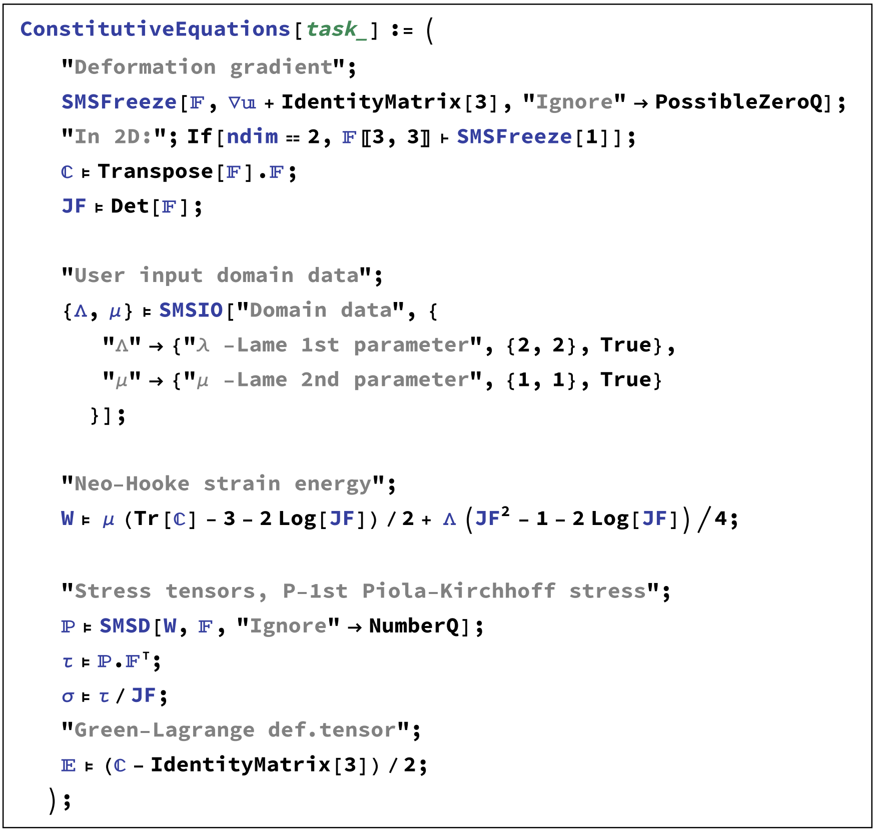
$$U^c(\boldsymbol{u}_\pi) = \int\limits_{\Omega_v} \left[\Psi^c(\nabla \boldsymbol{u}_\pi) - \bar{\boldsymbol{f}} \cdot \boldsymbol{u}_\pi \right] \, \mathrm{d}\Omega - \int\limits_{\Gamma_e^\sigma} \bar{\boldsymbol{t}} \cdot \boldsymbol{u}_\pi \, \mathrm{d}\Gamma \, . \tag{6.38}$$

Note, that arbitrary strain energy functions can be used in U^c. The below strain energy function $\Psi^c(\nabla \boldsymbol{u}_\pi)$ for compressible materials (2.55) is only specified as an illustrative example. The strain energy function

$$\Psi^c(\nabla \boldsymbol{u}_\pi) = \frac{\lambda}{4}[J(\boldsymbol{u}_\pi)^2 - 1 - 2 \ln J(\boldsymbol{u}_\pi)] + \frac{\mu}{2}[\operatorname{tr} \boldsymbol{C}(\boldsymbol{u}_\pi) - 3 - 2 \ln J(\boldsymbol{u}_\pi)] \tag{6.39}$$

depends with $\boldsymbol{F}_\pi = \boldsymbol{1} + \nabla \boldsymbol{u}_\pi$, $J = \det \boldsymbol{F}_\pi$ and $\boldsymbol{C}_\pi = \boldsymbol{F}_\pi^T \boldsymbol{F}_\pi$ on the gradient of $\boldsymbol{u}_\pi$. The part of the *AceGen* code that defines the nonlinear hyperelastic potential and computes 1st Piola-Kirchoff, Kirchhoff and Cauchy stresses and Green-Lagrange strains is provided in Fig. 6.14.

The first part in (6.38) has to be integrated. This integration depends on the ansatz order n and the spatial dimension d. In general one can use a Gauss integration with weights w_g, n_g Gauss points $\boldsymbol{\xi}_g$ and $\mathrm{d}\Omega = \det \boldsymbol{J}_e(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi}$ leading to



```
ConstitutiveEquations[task_] := (
   "Deformation gradient";
   SMSFreeze[𝔽, ∇𝕦 + IdentityMatrix[3], "Ignore" → PossibleZeroQ];
   "In 2D:"; If[ndim == 2, 𝔽⟦3, 3⟧ ⊢ SMSFreeze[1]];
   ℂ ⊨ Transpose[𝔽].𝔽;
   JF ⊨ Det[𝔽];

   "User input domain data";
   {Λ, μ} ⊨ SMSIO["Domain data", {
      "Λ" → {"λ -Lame 1st parameter", {2, 2}, True},
      "μ" → {"μ -Lame 2nd parameter", {1, 1}, True}
     }];

   "Neo-Hooke strain energy";
   W ⊨ μ (Tr[ℂ] - 3 - 2 Log[JF]) / 2 + Λ (JF² - 1 - 2 Log[JF]) / 4;

   "Stress tensors, P-1st Piola-Kirchhoff stress";
   ℙ ⊨ SMSD[W, 𝔽, "Ignore" → NumberQ];
   τ ⊨ ℙ.𝔽ᵀ;
   σ ⊨ τ / JF;
   "Green-Lagrange def.tensor";
   𝔼 ⊨ (ℂ - IdentityMatrix[3]) / 2;
  );
```

Fig. 6.14 Part of the code for the definition of the hyperelastic potential for finite strains

$$\int_{\Omega_v} \Psi^c(\nabla \boldsymbol{u}_\pi)\, \mathrm{d}\Omega = \sum_{g=1}^{n_g} \Psi^c\left[\nabla \boldsymbol{u}_\pi(\boldsymbol{\xi}_g)\right] \det \boldsymbol{J}_e(\boldsymbol{\xi}_g) = \Xi_c^{(d,n)}(\mathbf{u}_v)\,. \tag{6.40}$$

where $\Xi_c^{(d,n)}(\mathbf{u}_v)$ depends in a non-linear way on the unknown variables $\mathbf{u}_v$. Details that show how this integration is performed for specific cases are discussed next for two-dimensional applications.

Linear ansatz. In this case the gradient $\nabla \boldsymbol{u}_\pi$ is constant which yields a constant deformation gradient $\boldsymbol{F}_\pi = \mathbf{1} + \mathbb{P}_\nabla^{(d,1)}\,\mathbf{u}_v$, the Jacobian $J = \det \boldsymbol{F}_\pi$ and the right Cauchy–Green tensor $\boldsymbol{C}_\pi = \boldsymbol{F}_\pi^T \boldsymbol{F}_\pi$ as functions of the nodal displacements. Hence the strain energy in (6.39) is constant as well within the virtual element with respect to the coordinates $\mathbf{X}$. It is however a nonlinear function of the projected displacement $\boldsymbol{u}_\pi$.

Thus the first part in integral in (6.38) that is related to the strain energy can be simplified with (3.51) as

$$\int\limits_{\Omega_v} \left[\Psi^c(\nabla \boldsymbol{u}_\pi) \right] \, \mathrm{d}\Omega = \Psi^c(\nabla \boldsymbol{u}_\pi) \, \Omega_v = \Psi^c(\mathbb{P}_\nabla^{d,1} \, \mathbf{u}_v) \, \Omega_v \tag{6.41}$$

where Ω_v is the area of the virtual element. Note, that the strain energy $\Psi^c(\mathbb{P}_\nabla^{d,1} \, \mathbf{u}_v)$ is a nonlinear function with respect to the displacement nodal degrees of freedom.

The loading terms are the same as in the case of linear elasticity and follow from (6.6) and (6.7) in case of two-dimensional deformations.

The residual and the tangent matrix for the consistency part U_c^c, related to the element Ω_v, are computed by differentiation of the potential U_v^c with respect to the nodal displacements $\mathbf{u}_v$ using (3.58) or (3.157) for the two- or three-dimensional case, respectively,

$$\mathbf{R}_v^c = \frac{\partial U_v^c(\mathbf{u}_v)}{\partial \mathbf{u}_v} \quad \text{and} \quad \mathbf{K}_{Tv}^c = \frac{\partial \mathbf{R}_v^c(\mathbf{u}_v)}{\partial \mathbf{u}_v} \, . \tag{6.42}$$

All derivations can be easily executed with the symbolic tool *AceGen*, see Korelc and Wriggers (2016).

Quadratic ansatz, serendipity element. The derivation of the projection for the ansatz of a serendipity element was described in Sect. 3.1.7 for a quadratic interpolation. In Eq. (3.93) the internal moment $\mathbf{m}_1$ was expressed in term of the displacements $\bar{\mathbf{u}}_v$, see (3.85), at the nodes of the virtual element Ω_v

$$\mathbf{m}_1(\bar{\mathbf{u}}_v) = \int_{\Omega_v} \mathbf{H}_u^{(2,2)}(X, Y) \, \mathrm{d}\Omega \; [\mathbf{G}^{(2)}]^{-1} \, [\bar{\mathbf{H}}^{(2)}]^T \bar{\mathbf{u}}_v \tag{6.43}$$

which yields a constant dependency of the internal moment on the nodal displacements. It can be evaluated at element level since $\mathbf{X_k}$ are the known coordinates of the nodal points and the integral in (6.43) can be computed using the formulae in Appendix A. Thus the projection follows in short notation as

$$\mathbf{m}_1(\bar{\mathbf{u}}_v) = \mathbf{P}_{m1} \, \bar{\mathbf{u}}_v \qquad \text{with } \mathbf{P}_{m1} = \int_{\Omega_v} \mathbf{H}(X, Y) \, \mathrm{d}\Omega \; \mathbf{G}^{-1} \, [\bar{\mathbf{H}}^{(2)}]^T \, . \tag{6.44}$$

Based on this result the projection matrix $\mathbf{B}_e^q$ in (3.74) can be adjusted with (3.68) for the serendipity element

$$\mathbf{B}_{s\,v}^{(2)} = \left[\mathbf{P}_1^T \; \mathbf{P}_2^T \; \ldots \; \mathbf{P}_{2k-1}^T \; \mathbf{P}_{2k}^T \; \mathbf{P}_{2k+1}^T \; \ldots \mathbf{P}_{2n_V}^T \right] + \mathbf{B}_{\pi \, div}^{1\,T} \mathbf{P}_{m1} \, . \tag{6.45}$$

It has now the size $(10 \times 4n_V)$. This leads with (3.75) to the expression for the projected gradient, see also (3.94),

$$\nabla \mathbf{u}_\pi = \mathbf{B}^{(2,2)}_{u\,\pi}(X, Y)\,[\mathbf{G}^{(2)}_\pi]^{-1}\,\mathbf{B}^{(2)}_{s\,v}\,\bar{\mathbf{u}}_v = \mathbf{B}^{(2,2)}_{u\,\pi}(X, Y)\,\mathbb{P}^{(2,2)}_{\nabla\,S}\,\bar{\mathbf{u}}_v \tag{6.46}$$

which is a linear function of $\mathbf{X}$ within the serendipity element. We note that the matrix $\mathbf{B}^q_\pi(X, Y)$ has the size (4×10) while the projector $\mathbb{P}^{(2,2)}_{\nabla\,S}$ is of size $(10 \times 4n_V)$. The gradient $\nabla \mathbf{u}_\pi$ in the equation above is given in Voigt notation. For the formulation of finite deformations it is more convenient to express the gradient in tensor notation, e.g. in the 2d case by a 2×2-matrix. This can be achieved by assembling the components of the displacement gradient in a matrix

$$\nabla \mathbf{u}_\pi = \begin{Bmatrix} u_{x,x} \\ u_{x,y} \\ u_{y,x} \\ u_{y,y} \end{Bmatrix} \Longrightarrow \nabla \boldsymbol{u}_\pi = \begin{bmatrix} u_{x,x} & u_{x,y} \\ u_{y,x} & u_{y,y} \end{bmatrix} \tag{6.47}$$

where the components $u_{\alpha,\beta}$ follow by evaluating the matrix $\mathbf{B}^{(2,2)}_{u\,\pi}$, see (3.63), row wise. For example

$$u_{x,x} = \big\langle 1\ 0\ 0\ 0\ X\ 0\ Y\ 0\ 0\ 0 \big\rangle\ \mathbb{P}^{(2,2)}_{\nabla\,S}\,\bar{\mathbf{u}}_v\,.$$

Hence the components $u_{\alpha,\beta}$ of the displacement gradient $\nabla \boldsymbol{u}_\pi$ depend on the nodal degrees of freedom $\bar{\mathbf{u}}_v$ and the coordinates $\mathbf{X} = (X, Y)$ inside the virtual element Ω_v. To simplify notation we introduce for the tensor notation the projector $\mathbb{G}_S(X, Y)$ such that

$$\nabla \boldsymbol{u}_\pi = \mathbb{G}_S(X, Y)\,\bar{\mathbf{u}}_v\,. \tag{6.48}$$

Now the deformation gradient $\boldsymbol{F}_\pi = \mathbf{1} + \nabla \boldsymbol{u}_\pi$ can be computed and with that the right Cauchy Green tensor $\boldsymbol{C}_\pi$ and the Jacobian $J_\pi = \det \boldsymbol{F}_\pi$. Based on these quantities the first part in the integral in (6.38) can be evaluated for the serendipity virtual element. It is related to the strain energy (6.39) and thus will be a nonlinear function of $\bar{\mathbf{u}}_v$ and the coordinates $\mathbf{X}$.

In the geometrically linear case it was possible to compute the integral on the left side of (6.41) exactly by using integration formulae that only needed to be evaluated at the edges of the virtual element. Due to the nonlinearities of the finite strain formulation this is no longer possible. Hence numerical integration has to be applied.[4] This can be performed by subdividing the virtual element into n_{int} internal triangles Ω^i_m, see Fig. 6.15

$$\int\limits_{\Omega_v} \big[\Psi^c(\nabla \boldsymbol{u}_\pi)\big]\,\mathrm{d}\Omega = \sum_{m=1}^{n_{int}} \int\limits_{\Omega^i_m} \big[\Psi^c(\nabla \boldsymbol{u}_\pi)\big]\,\mathrm{d}\Omega\,. \tag{6.49}$$

[4] The approximation $\boldsymbol{u}_h$ is in general not known in the virtual element, see assumption (a0) to (a3) in Sect. 3.1.1 for the space $\mathbf{V}_h$. However the ansatz $\mathbf{u}_\pi$ can be employed in the virtual element by using the equivalent projector of the relaxed space $\mathbf{W}_h$ based on (a3)' and (a3)" in Sect. 3.1.2.

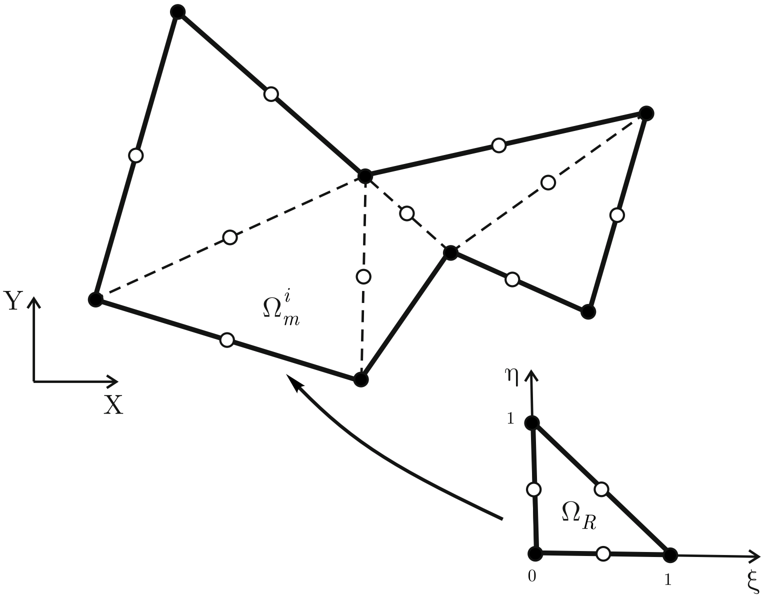


Fig. 6.15 Internal triangular mesh for integration of the consistency part

Once the sub-triangularization is finalised each integral over Ω_m^i is evaluated by a Gauss point integration. For the efficient implementation of the numerical integration a coordinate transformation is performed using a linear isoparametric map,[5] see (6.29), $\mathbf{X} = \sum_{i=1}^{3} N_I(\xi\ ,\eta)\,\mathbf{X}_I$ where $\mathbf{X}_I$ are the vertices of the virtual element, black points in Fig. 6.15. This yields

$$\int\limits_{\Omega_m^i} \Psi^c[\nabla \boldsymbol{u}_\pi(X,Y)]\mathrm{d}\Omega = \int\limits_{\xi=0}^{\xi=1}\int\limits_{\eta=0}^{\eta=1-\xi} \Psi^c[\nabla \boldsymbol{u}_\pi(\xi,\eta)]\ \det \mathbf{J}_e(\xi,\eta)\,\mathrm{d}\eta\,\mathrm{d}\xi$$
$$= \sum_{g=1}^{n_g} w_g\ \Psi^c[\nabla \boldsymbol{u}_\pi(\xi_g,\eta_g)]\det \mathbf{J}_e(\xi_g,\eta_g) \tag{6.50}$$

where $\det \mathbf{J}_e(\xi_g,\eta_g)$ is the Jacobian of the isoparametric mapping from the triangle Ω_m^i to the reference triangle Ω_R in the ξ, η reference space, see Fig. 6.15. The weights

[5] It is not necessary to use a quadratic map based on the vertex and edge nodes of an internal triangle since the edge nodes are midpoints between the vertices and thus a coordinate transformation using a quadratic ansatz function will actually reduce to the linear ansatz provided above.

of the Gauss integration are denoted by w_g and ξ_g , η_g are the Gauss point coordinates. Usually $n_g = 3$ Gauss points are sufficient for the integration.

Expression (6.50) is a nonlinear function of the nodal displacements $\bar{\mathbf{u}}_v$. The residual and tangent matrix for the consistency part of the serendipity virtual element can now be obtained by applying Eq. (6.42).

6.2.2 Stability Term

For the quadratic serendipity virtual element we will use only the bi-linear stabilization. Energy stabilization looses for higher order ansatz functions its advantages for two reasons:

- The order of the internal triangles has to be the same as for the virtual element ansatz. Otherwise the order of convergence is reduced. This leads to internal triangles with extra nodes and thus reduces the efficiency of the element due to the corresponding additional degrees of freedom.
- The bending behaviour of quadratic elements is already very good, see e.g. the example Fig. 6.13, thus there is no need to improve the element by such stabilization.

Hence for higher order virtual elements it only makes sense to use the discrete bilinear form (6.23). The bi-linear stabilization can be employed without changes as well for virtual elements with linear as with quadratic ansatz spaces. Since the stiffness of the solid can change in nonlinear analysis due to the deformation state, the parameters of the stabilization term have to be adjusted accordingly. Some suggestions for deformation depending changes of the stabiity parameters for the bi-linear discrete form in (6.23) can be found below.

The stabilization parameter in (6.23) can be written for a problem of spatial dimension d as

$$\gamma(\mathbf{u}_v) = \alpha(\mathbf{u}_v)\,\frac{h_v^d}{\operatorname{tr}\left[[\mathbf{H}_u^{(d,n)}(\mathbf{X}_k)]^T\mathbf{H}_u^{(d,n)}(\mathbf{X}_k)\right]} \tag{6.51}$$

which depends in nonlinear situations on the deformation $\mathbf{u}_v$. The ansatz function (order n) for the projection $\mathbf{H}_u^{(d,n)}(\mathbf{X}_k)$ is evaluated at the vertex $\mathbf{X}_k$, h_e is a characteristic element diameter and $\alpha(\mathbf{u}_v)$ is a parameter. Two different possibilities exist for the selection of the parameter $\alpha(\mathbf{u}_v)$:

- In Beirão da Veiga et al. (2015) the parameter $\alpha(\mathbf{u}_v)$ is chosen as $\alpha(\mathbf{u}_v) = \|\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}}(\boldsymbol{F}_\pi)\|$, where $\boldsymbol{P}$ is the first Piola-Kirchhoff stress tensor related to the consistency term.
  An alternative stabilization parameter $\alpha(\mathbf{u}_v) = \frac{1}{4}\operatorname{tr}[\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}}(\boldsymbol{F}_\pi)]$, referred to as a trace-based stabilization, was proposed in Chi et al. (2017). Both stabilization parameters take into account the deformation state of the solid and thus are a far better approximations than a constant parameter as selected in the geometrical linear

analysis. For the special case of the strain energy function, like (6.39), the trace of constitutive tangent can be obtained explicitly

$$\alpha(\mathbf{u}_v) = \frac{1}{4}\,\text{tr}[\frac{\partial^2 \Psi}{\partial \boldsymbol{C} \partial \boldsymbol{C}}] = \frac{1}{4}\,\{2\,\mu + \lambda[J_\pi^2[\text{tr}(\boldsymbol{C}_\pi^{-1})]^2 - (J_\pi^2 - 1)\text{tr}(\boldsymbol{C}_\pi^{-1} \cdot \boldsymbol{C}_\pi^{-1})]\}\,. \tag{6.52}$$

- Another possibility is to use the trace of the global tangent matrix $\mathbf{K}_T$ for the stabilization parameter. With N being the total number of unknowns of the global equation system the parameter is given by

$$\alpha(\mathbf{u}_v) = \frac{1}{4\,N}\left(\sum_{i=1}^{N}(\mathbf{K}_{T\,ii}^c)^2\right)^{\frac{1}{2}} \quad \text{or} \quad \alpha(\mathbf{u}_v) = \frac{1}{4\,N}\text{tr}\,\mathbf{K}_T^c = \frac{1}{N}\sum_{i=1}^{N} K_{T\,ii}^c\,. \tag{6.53}$$

  The number of unknowns (degrees of freedom) follow with the dimension d of a solid mechanics problem as $N = d\,n_N$ for an element with n_N nodes . This type of stabilization can be very good since it is direcly related to the current stiffness matrix, but it is not feasible to linearize this term in a consistent manner.

Still, all choices for the stabilization parameter present a challenge when linearizing in a Newton procedure, even with the use of the symbolic tool *AceGen*, see Korelc and Wriggers (2016), which automatically provides the code for the residual vector and tangent stiffness matrix, see e.g. De Bellis et al. (2019) where (6.52) was consistently linearized. If linearization is not possible then the parameter has to be computed at the last converged load step and then kept constant during the iteration within the load step. This leads eventually to incremental load stepping, even if the physical problem allows the computation of the solution in one step.

Again, as in the consistency part, the stabilization energy is a nonlinear function of the nodal displacements. All further derivations leading to the residual vector $\mathbf{R}_v$ and the tangent matrix $\mathbf{K}_{Tv}$ were performed with the symbolic tool *AceGen*, see Korelc and Wriggers (2016). This yields for (6.23)

$$\mathbf{R}_v^s = \frac{\partial U_{stab}(\mathbf{u}_v)}{\partial \mathbf{u}_v} \quad \text{and} \quad \mathbf{K}_{Tv}^s = \frac{\partial \mathbf{R}_v^s(\mathbf{u}_v)}{\partial \mathbf{u}_v}\,. \tag{6.54}$$

Thus the total residual and tangent matrix of the virtual element are given by the sum of expressions (6.110) and (8.14):

$$\mathbf{R}_v = \mathbf{R}_v^c + \mathbf{R}_v^s \quad \text{and} \quad \mathbf{K}_{Tv} = \mathbf{K}_{Tv}^c + \mathbf{K}_{Tv}^s\,.$$

6.2.3 Virtual Elements for Three-Dimensional Problems in Nonlinear Elasticity

Here we discuss the construction of a three-dimensional solid element for a linear ansatz. Basically the derivatiion of residual and tangent stiffness matrix follow along the same lines as in the two-dimensional case. Differences are discussed below.

Consistency term. In the three dimensional case the strain energy part of the consistency term is computed for a linear ansatzv exactly in the same way as (6.41). This is due to the fact that the projected gradient $\nabla \boldsymbol{u}_\pi$ is constant, see (3.156)

$$U^c(\boldsymbol{u}_\pi) = \int\limits_{\Omega_v} \left[\Psi^c(\boldsymbol{u}_\pi)\right] \, d\Omega = \Psi^c(\boldsymbol{u}_\pi)\,\Omega_v\,. \tag{6.55}$$

Hence the only remaining task is the computation of the volume Ω_v of the three-dimensional virtual element. Using the divergence theorem the integral over the n_F faces Γ_f

$$\Omega_v = \frac{1}{3}\sum_{f=1}^{n_F} \int\limits_{\Gamma_f} \boldsymbol{X}_e \cdot \boldsymbol{N}_f \, \mathrm{d}\Gamma = \frac{1}{3}\sum_{f=1}^{n_F} \boldsymbol{N}_f^T \int\limits_{\Gamma_f} \mathbf{X}_e \, \mathrm{d}\Gamma \tag{6.56}$$

has to be evaluated with $\boldsymbol{N}_f$ being the normal vector of a face that is constant and $\mathbf{X}_e$ are the coordinates describing the face.

Loading term. Another difference to the two-dimensional element is related to the loading terms where the volume load has now to be computed over the volume of the element and the surface tractions have to be evaluated at the faces. The volume load is given by

$$U^f = \int\limits_{\Omega_v} \boldsymbol{u}_\pi \cdot \bar{\boldsymbol{f}} \, \mathrm{d}\Omega = \mathbf{u}_v^T \, [\mathbb{P}_u^{(3,1)}]^T \int\limits_{\Omega_v} [\mathbf{H}_u^{(3,1)}(X, Y, Z)]^T \, \bar{\boldsymbol{f}} \, \mathrm{d}\Omega \tag{6.57}$$

where $\mathbb{P}_u^{(3,1)}$ and $\mathbf{H}_u^{(3,1)}(X, Y, Z)$ can be found in (3.157). The volume integral can be evaluated for any polynomial function of the volume load $\bar{\boldsymbol{f}}$ using (A.25) or the algortihm in Mirtich (1996). For the specific case of a constant volume load (A.35) can be applied.

The load related to the surface tractions $\bar{\boldsymbol{t}}$ is given by

$$U^\sigma = \int\limits_{\Gamma_{\sigma e}} \boldsymbol{u}_h \cdot \bar{\boldsymbol{t}} \, \mathrm{d}\Gamma = \sum_{f=1}^{n_{Ft}} \int\limits_{\Gamma_f} \boldsymbol{u}_h \cdot \bar{\boldsymbol{t}} \, \mathrm{d}\Gamma\,. \tag{6.58}$$

Here the evaluation follows over the n_{Ft} faces that are loaded by the traction $\bar{\boldsymbol{t}}$. The integration is carried out by approximating $\int_{\Gamma_f} \boldsymbol{u}_h = \int_{\Gamma_f} \boldsymbol{u}_\pi$ and then transforming the ansatz for $\boldsymbol{u}_\pi$ to the local coordinate system of the face Γ_f, see (3.142), with $\mathbf{u}_f$ unknowns at its vertices. Then we have two possibilities

1. Perform a computation over the face using (3.146). We note that this yields an exact integration for loads that can be described by polynomials, see (3.141),

$$U_F^\sigma = \int\limits_{\Gamma_f} \boldsymbol{u}_\pi \cdot \bar{\boldsymbol{t}}\, \mathrm{d}\Gamma = \frac{1}{2} \int\limits_{\gamma_f} \left(\int \boldsymbol{u}_\pi \cdot \bar{\boldsymbol{t}}\, dX^f\, \boldsymbol{e}_X^{f\,T} \boldsymbol{N}_e + \int \boldsymbol{u}_\pi \cdot \bar{\boldsymbol{t}}\, dY^f\, \boldsymbol{e}_Y^{f\,T} \boldsymbol{N}_e \right) d\gamma \tag{6.59}$$

   where $\boldsymbol{u}_\pi(X^f, Y^f) \cdot \bar{\boldsymbol{t}}(X^f, Y^f)$ has to be expressed by the local coordinate system (3.142) (X^f, Y^f) and $\boldsymbol{N}_e$ is given by (3.144), see Fig. 3.11.
2. Approximate $\boldsymbol{u}_\pi$ at the face by a triangularization $\Gamma_f = \cup_{\tau=1}^{n_\tau} \Gamma_\tau$ using the ansatz (3.151) for each of the triangles Γ_τ with the ansatz $\boldsymbol{u}_h^\tau \in \boldsymbol{u}_f$, see also (3.152). This integration works for general loadings, that are described by arbitrary functions, and yields

$$U_\tau^\sigma = \int\limits_{\Gamma_f} \boldsymbol{u}_\pi \cdot \bar{\boldsymbol{t}}\, \mathrm{d}\Gamma = \sum_{f=1}^{n_F} \sum_{\tau=1}^{n_\tau} \int\limits_{\Gamma_\tau} \boldsymbol{u}_h^\tau \cdot \bar{\boldsymbol{t}}\, \mathrm{d}\Gamma = \sum_{f=1}^{n_F} \sum_{\tau=1}^{n_\tau} \sum_{g=1}^{n_g} w_g \boldsymbol{u}_h^\tau(\boldsymbol{\xi}_g) \cdot \bar{\boldsymbol{t}}(\boldsymbol{\xi}_g) \tag{6.60}$$

   where $\boldsymbol{u}_h^\tau$ is given in (3.151), see Fig. 3.12, and $\boldsymbol{\xi}_g$ are the Gauss points used to integrate over each of the triangles Γ_τ.

Stability term. The energy stabilization technique was presented in Hudobivnik et al. (2018) for three-dimensional virtual elements undergoing finite elasto-plastic deformations.[6] This procedure can, of course, also be applied for hyperelastic response. It is based on the split (6.37) into the consistency and stabilization part of the energy where the stabilization energy (5.31) can be introduced. The only difference to the two-dimensional case is the way how to compute the stabilization integral

$$\widehat{U}(\boldsymbol{u}_h) = \mathop{\mathbf{A}}_{e=1}^{n_e} \int\limits_{\Omega_v} \widehat{\Psi}(\boldsymbol{u}_h) \mathrm{d}\Omega\,. \tag{6.61}$$

In the previous section the integral (6.61) was evaluated using an internal triangular mesh. This idea is now applied to the 3D problem in a similar way. In order to approximate the displacement field $\boldsymbol{u}_h$ in (6.61) an inscribed tetrahedral finite element mesh is introduced, see Fig. 6.16. It consists of n_P linear four-noded tetrahedra that are only connected to the nodes of the virtual element. Again, as in the two-dimensional case, no additional degrees of freedom are introduced. In Hudobivnik et al. (2018)

[6] The discrete bi-linear form (6.23) can be employed for the stabilization of the three-dimensional element as well. Following Sect. 6.2.2, the stabilization parameter depends in the nonlinear case on the solution $\boldsymbol{u}_\pi$ and can be selected in the same manner as in (6.51).

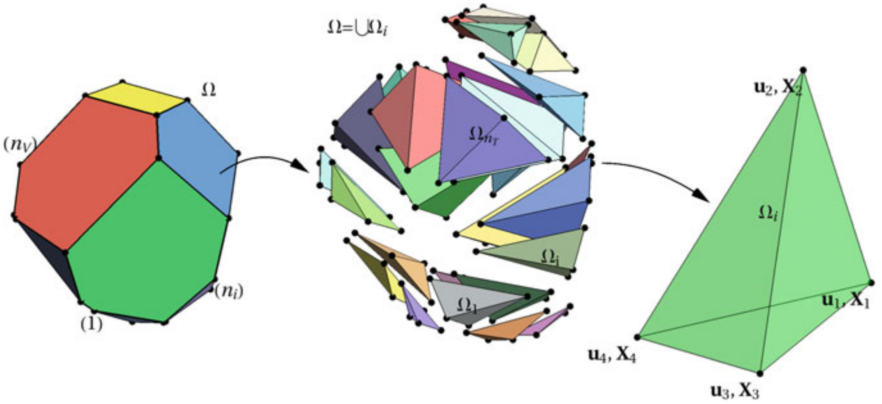


Fig. 6.16 Internal tetrahedral mesh

it was shown that such a mesh is sufficient to compute the stabilization energy in (6.61). The same function as in (6.55) is chosen as stabilization energy. By using a linear ansatz for the displacement field an approximation can be computed for the strains within each tetrahedron Ω_m^i of the inscribed mesh.

Based on this internal mesh the displacement gradient $\nabla\,\boldsymbol{u}_h^m$ can be computed in each tetrahedral element. It is constant within the element due to the linear ansatz for the four-noded thetrahedron. Hence the discrete deformation gradient $\boldsymbol{F}_m^i = \mathbf{1} + \nabla\,\boldsymbol{u}_h^m$ is constant as well within each element Ω_m^i which leads to a very simple integration scheme for the tetrahedra Ω_m^i defining the inscribed mesh.[7]

With the above approximations the pseudo potential $\widehat{U}(\boldsymbol{u}_h)$ within a virtual element v is a result of the assembly of all internal tetrahedral elements

$$\widehat{U}(\boldsymbol{u}_h) = \mathop{\mathbf{A}}_{i=1}^{n_m} \int\limits_{\Omega_m^i} \widehat{\Psi}(\nabla\,\boldsymbol{u}_h^m)\,\mathrm{d}\Omega\,. \tag{6.62}$$

where n_m is the total number of internal tetrahedral elements within a virtual element Ω_v and $\nabla\,\boldsymbol{u}_h^m$ the part of the nodal displacements in a virtual element that is related to the specific tetrahedral element Ω_m^i.

The values of the Lamé parameters in the pseudo potential (6.62) have to be different from the original ones. The same procedure as in the two-dimensional case can be employed to compute the constitutive parameters $\widehat{\Lambda}$ and $\widehat{\mu}$ for the stabilization energy based on (6.37). The only difference is the computation of the parameter β related to the shape of the virtual element. In the three-dimensional case this geometry parameter follows from the inner and outer radii, R_i^2 , R_a^2 respectively, see Fig. 6.17,

[7] The shape of the triangles is not optimal for a finite element discretization. Since we only want to compute a stabilization and thus the classical shape restrictions are not relevant. However, to fulfil the patch test the discretization has to be the same for elements at shared faces.

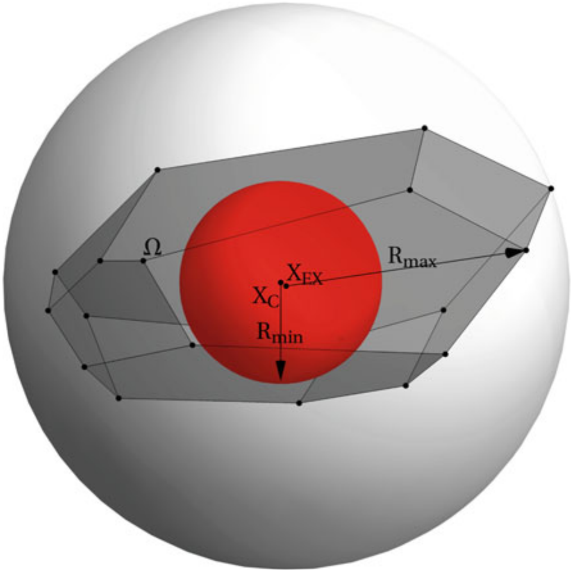


Fig. 6.17 Inner and outer radius of a virtual element

$$\beta = 2\sqrt{2}\,(1+\nu)\,\frac{R_i^2}{R_a^2 - 2\,R_i^2}\,. \tag{6.63}$$

The inner radius is computed by using the distance from the geometrical centre to the convex hull of the virtual element while the outer radius is defined by the maximum distance of nodes related to the virtual element, see Fig. 6.17. Again the for a cube the ratio $\frac{R_i^2}{R_a^2-2\,R_i^2}$ is one.

Influence of integration and stabilization on the loading term It should be noted that stabilization, either using the energy form (6.62) or the bi-linear form (6.23), and the way the integration over the faces is performed leads to different nodal values for the traction loading term. These have to be selected with care in order to fulfil the patch test. In general we have four different cases

1. Bi-linear form of the stabilization and edge integration to obtain $\nabla \boldsymbol{u}_\pi$ for the consistency part, see (3.146) and (3.147). Here one has to use U_F^σ, see (6.59).
2. Bi-linear form of the stabilization and integration of the consistency part to obtain $\nabla \boldsymbol{u}_\pi$ using triangularization of the faces Γ_f, see (3.152). In this case U_τ^σ has to be applied, see (6.60).
3. Energy stabilization and and edge integration to obtain $\nabla \boldsymbol{u}_\pi$ for the consistency part, see see (3.146) and (3.147). Here the weighted term $(1-\beta)\,U_\tau^\sigma + \beta U_F^\sigma$ has to be employed where β is the reduction parameter for the Lamé constants, computed in (6.63).

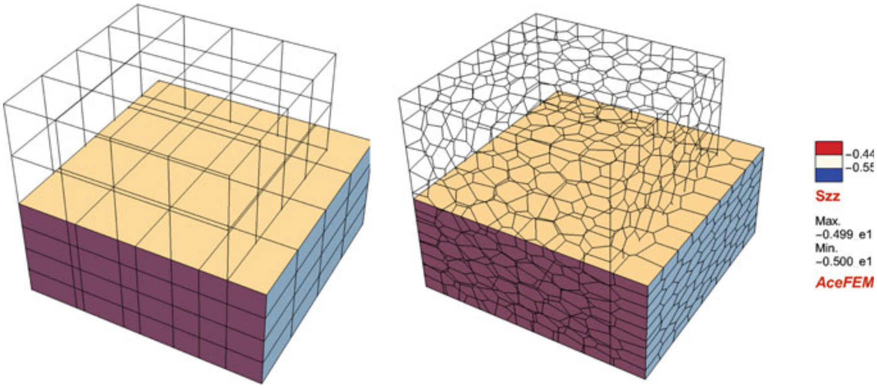


Fig. 6.18 Patch test using a VE-H2S and a Voronoi mesh (VE-VOR)

4. Energy stabilization and integration of the consistency part to obtain $\nabla \boldsymbol{u}_\pi$ using triangularization of the faces Γ_f, see (3.152). This requires the loading term U^σ_τ, see (6.60).

The patch test is fulfilled in all cases where the integration is performed according to the above list. This is demonstrated for two meshes (VE-H2S and VE-VOR) in Fig. 6.18 for a block undergoing finite deformations under constant pressure.

Residual and tangent. All further computations leading to the residual vector $\mathbf{R}_v$ and the tangent matrix $\mathbf{K}_v$ of the element can be performed with the symbolic tool *AceGen*, see Korelc and Wriggers (2016). This yields for all energy contributions the total residual and tangent matrix of the virtual element $U = U^c + U^f + U^\sigma + \widehat{U}$

$$\mathbf{R}_v = \frac{\partial U(\mathbf{u}_v)}{\partial \mathbf{u}_v} \quad \text{and} \quad \mathbf{K}_{Tv} = \frac{\partial \mathbf{R}_v(\mathbf{u}_v)}{\partial \mathbf{u}_v} \,. \tag{6.64}$$

Note that the loading terms U^f and U^σ only contribute to the residuum.[8]

6.2.4 General Solution for Nonlinear Equations

Assembling all element contributions within a discretization yields the global matrix system that is related to the weak form or potential. First the residual vectors in (6.54) for the two-dimensional case or (6.64) for the three-dimensional case lead together with the loading terms to

[8] Contributions to the tangent matrix appear for displacement dependent loading, like follower loads due to fluid pressure, see e.g. Wriggers (2008).

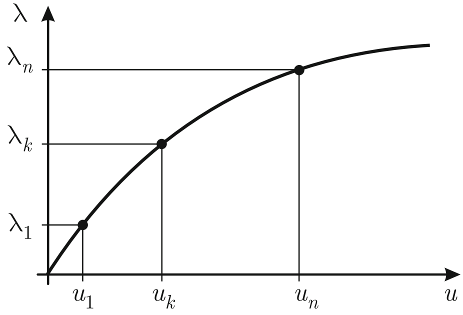


Fig. 6.19 Sketch of a load stepping procedure

$$\mathop{\mathsf{A}}_{v=1}^{n_v}\left[\mathbf{R}_v(\mathbf{u}_v)-\lambda\,\bar{\mathbf{f}}_v\right]=\mathbf{0}\quad\Longrightarrow\quad\mathbf{R}(\mathbf{u})-\lambda\,\bar{\mathbf{f}}=\mathbf{G}(\mathbf{u}\,,\lambda)=\mathbf{0}\tag{6.65}$$

where $\mathbf{u}$ are the nodal unknowns of the global discrete problem and λ is a load multiplier which will be increased in steps up to a final value $\lambda_{n_{steps}}$ in a load stepping procedure, see Fig. 6.19. Load stepping can be necessary for the solution of highly nonlinear problems.

The nonlinear algebraic equation (6.65) can be solved by different methods, see e.g. Zienkiewicz and Taylor (2000b) and Wriggers (2008). For problems undergoing finite elastic deformations the Newton-Raphson method is the most efficient choice.[9] It leads to an iteration within each load step that has quadratic convergence properties, see Algorithm 1.

The quantity $\mathbf{u}_0$ is the displacement field at the beginning of the load stepping procedure. The tolerance ε is usually set to $\varepsilon = 10^{-8}$ since the Newton Raphson algorithm depicts a quadratic convergence. In most applications 4 to 6 iterations per

Algorithm 1: Newton-Raphson algorithm for finite deformations

```
Given: $\mathbf{u}_0$ , $\lambda_1$      Find: $\mathbf{u}_{n+1}$ ;
for $n = 0$ to $(n_{steps} - 1)$ do
    $\mathbf{u}^0_{n+1} = \mathbf{u}_n$ ;
    for $i = 0$ to $n_{iter}$ do
        $\mathbf{K}_T(\mathbf{u}^i_{n+1})\Delta\mathbf{u}^{i+1}_{n+1} = -\mathbf{G}(\mathbf{u}^i_{n+1}\,, \lambda_{n+1})$;
        $\mathbf{u}^{i+1}_{n+1} = \mathbf{u}^i_{n+1} + \Delta\mathbf{u}^{i+1}_{n+1}$;
        $\|\mathbf{G}(\mathbf{u}^{i+1}_{n+1})\| \le \varepsilon \longrightarrow$ STOP
    end
    $\mathbf{u}_{n+1} = \mathbf{u}^{i+1}_{n+1}$;
end
```

[9] Depending on the nonlinearity of the problem, more sofisticated schemes, like line search or arc-length procedures, can be applied to stabilize the convergence behaviour, for more details, see e.g. Wriggers (2008).

load step are necessary to reach convergence. to be on the safe side, n_{iter} can be set to 15. If more than n_{iter} iterations are needed, it is advisable to reduce the load factor, e.g. $\lambda_{k+1} = \frac{1}{2}(\lambda_{k+1} + \lambda_k)$.

Within the Algorithm 1 the matrices and vectors

$$\mathbf{G} = \mathop{\mathsf{A}}\limits_{v=1}^{n_v} \mathbf{R}_v - \lambda_n \mathop{\mathsf{A}}\limits_{v=1}^{n_v} \bar{\mathbf{f}}_v \quad \text{and} \quad \mathbf{K}_T = \mathop{\mathsf{A}}\limits_{v=1}^{n_v} \mathbf{K}_{T\,v} \tag{6.66}$$

are the global residual and the global tangent, respectively.

6.2.5 *Numerical Examples, Compressible Case*

Several examples are considered to compare the low order virtual element formulation and for some cases also the quadratic serendipity virtual element with existing state-of-the-art finite elements. All examples are subjected to loads that lead to finite deformation strain states and thus the depicted deformation states are not scaled. All formulations employ the potential (6.38) when the compressible case is considered. Within the virtual element method it is possible to use different meshes. The general definition of the type of meshes applied in the numerical examples can be found in Sect. B. If a finer distinction between mesh types or virtual elements is needed it will be defined within the sections describing the examples.

The computations are performed using the Newton-Rapson algorithm described above. When necessary a load stepping procedure will be applied. All formulations are linearized in a consistent manner using *AceGen*, hence quadratic convergence is achieved. In the first example, some basic features of the virtual element technology regarding comparisons to FEM and selection of the stabilization parameters are discussed in detail. This example and the next one are related to the work Wriggers et al. (2017).

6.2.5.1 Cook's Membrane Problem

Cook's membrane problem described in Sect. 6.1.4, see Fig. 6.10, is also used to discuss the behaviour of virtual elements in the nonlinear range. The membrane is again loaded at the right end by a constant load q_0 distributed in vertical direction, as depicted in Fig. 6.10.

The selected constitutive parameters for the Lamé constants are $\mu = 40$ and $\Lambda = 100$ which results in a compressible material. The distributed load is given as $q_0 = 4$.

The mesh, used to discretize the membrane, was automatically generated by the meshing tools in *Mathematica*. Different mesh types are employed (a regular mesh, a distorted mesh and a Voronoi mesh) to illustrate the performance of the low order virtual element with energy stabilization.

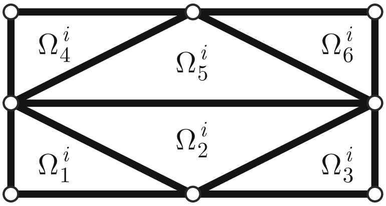


Fig. 6.20 Internal mesh for a regular shaped virtual element with 8 nodes (VE-Q2S)

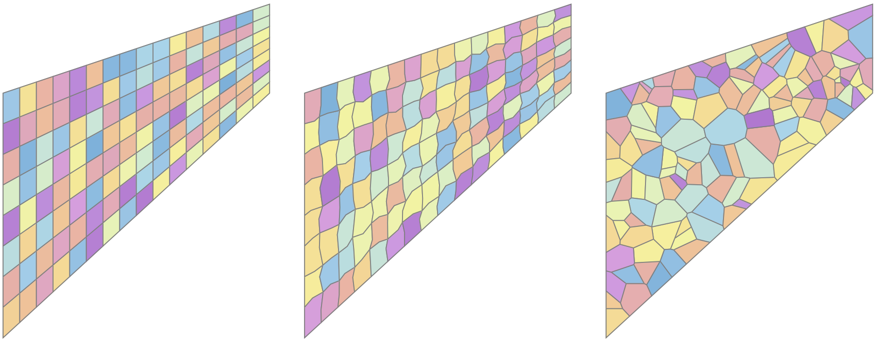


Fig. 6.21 Distorted mesh d1 and d2 (VE-Q2S) and Voronoi mesh (VE-VOR)

The regular mesh is based on the VE-Q2S element, see left side of Fig. 6.21, that has an internal triangular mesh with six triangles Ω_m^i as depicted in Fig. 6.20. The internal mesh is an arbitrary choice, it could have been constructed in a different way.

A selection of the different meshes can be found in Fig. 6.21 where the undistorted and two types of distorted meshes are depicted. The mesh in the middle is a mesh that can be obtained form the standard mesh on the left side by randomly changing the nodal placements of the corner and side nodes of the element. Note that this leads to a mesh with convex and non-convex shaped elements. A Voronoi mesh is shown on the right side of Fig. 6.21 which includes elements with arbitrary number of nodes and element sizes in a random manner. In all distorted cases a special optimization procedure, provided in *Mathematica*, has been applied to generate the internal mesh for the energy stabilization.

Different mesh densities were employed to compute the solution for all generated types of meshes. The mesh refinement is uniform in the sense that the finer meshes are included in the coarser meshes for the regular and the distorted mesh types of type d1 and d2. This enables convergence studies that illustrate differences of the formulations. The number 2^N denotes the mesh division: for example, in the left part of Fig. 6.21, a mesh with division of $2^3 = 8$ is depicted. The sequence of 2^N that was used in this study is $N \in \{1, 2, 3, 4, 5, 6, 7, 8\}$. Note that the final mesh density

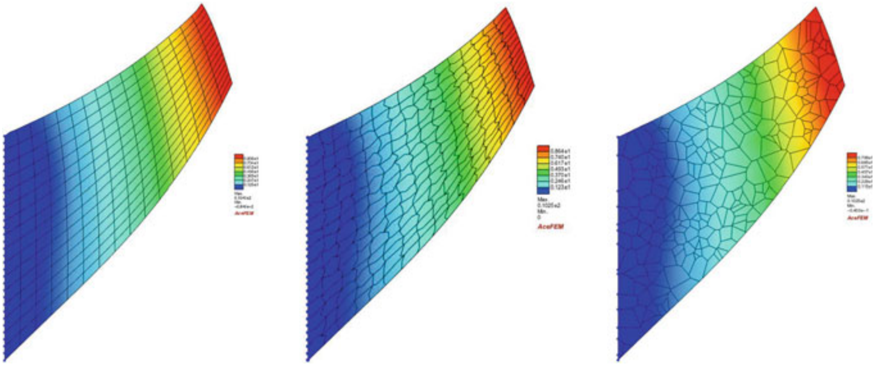


Fig. 6.22 Deformed configuration of the cantilever beam for a regular (VE-Q2S), a distorted (VE-Q2S) and a Voronoi mesh (VE-VOR)

Table 6.1 Convergence rate, load applied in one load step

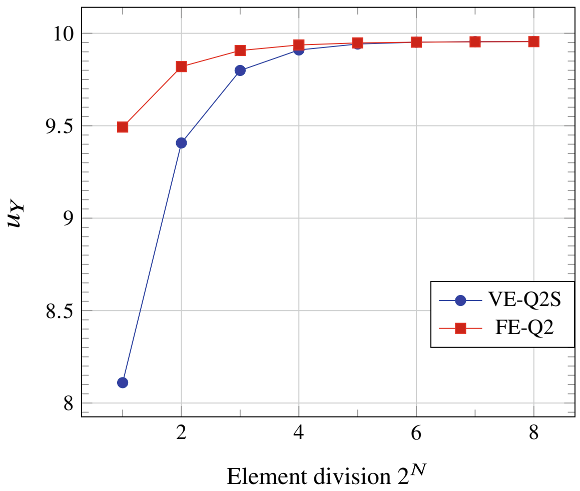
Iteration	3	4	5	6	7
$\|\mathbf{R}(\mathbf{u})\|$	$3.59 \cdot 10^{-1}$	$7,21 \cdot 10^{-2}$	$9.46 \cdot 10^{-4}$	$3.84 \cdot 10^{-7}$	$1.19 \cdot 10^{-12}$

leads to roughly $4 \cdot 10^5$ degrees of freedom. The Voronoi meshes are generated using randomly distributed nodes at each refinement step thus the finer meshes are not embedded in the coarser meshes.

In order to test the robustness of the virtual element formulation the load was applied in only one load step. Due to the consistent linearization of the element residual quadratic convergence is achieved. The deformation state, depicted in Fig. 6.22 for a mesh with subdivision $N = 3$, is computed for all other mesh divisions. All of them need 7 iterations to converge. The convergence rate of the Newton-Rapson method is depicted in Table 6.1 for a Cook's membrane problem with 32×32 elements.

Mesh convergence is investigated using the vertical displacement of the upper node, $u_Y(48, 60)$ at the right end of Cook's membrane. The study is performed for the virtual element formulation VE-Q2S. The Lamé parameters for the stabilization term in the strain energy function were computed using the formulation leading to (6.34) and (6.36), which resulted in $\widehat{\Lambda} = 25$ and $\widehat{\mu} = 17$ for the regular mesh. The virtual element VE-Q2S is compared for the compressible case with a standard biquadratic FE-Q2 finite element that has a theoretical higher convergence rate, see Fig. 6.23. It is obvious that the FE-Q2 element has a superior performance, however the VE-Q2S element for a mesh with $N = 4$, leading to a 16×16 element mesh, is already very close to the solution of the FE-Q2 element. For a mesh with $N > 4$ the solution of the VE-Q2S element is almost not distinguishable from the Q2-element and thus performs almost as well. It can be observed that the VE-Q2S formulation converges from below like classical displacement finite elements.

Fig. 6.23 Convergence Study: VE-Q2S and FE-Q2 element



VEM versus FEM using the internal mesh. One could argue that the construction of an element consisting only of the internal triangular mesh, employed for the stabilization, will lead to the same result as the virtual element since it has the same number of nodes but more elements, see Fig. 6.20. To show that the VE-Q2S is superior to such an element, here called FE-T6 element, the FE-T6 element was coded as a macro-element with the shape of the virtual element, but consisting only of the internal triangles. The internal mesh that was used to create the FE-T6 macro-element is shown in Fig. 6.20. With this FE-T6 macro-element Cook's membrane problem is solved for the compressible case based on the potential (6.38). The results stemming from the FE-T6 element are presented in Fig. 6.24 together with the response of the VE-Q2S virtual element. Again the convergence behaviour of the vertical displacement at point (48, 60) is depicted. Clearly the VE-Q2S element is superior to the FE-T6 element. This can be explained by the fact that the consistency term relaxes the response—like a one point (under) integrated finite element—and the constitutive parameter of the stabilization term are derived from the equivalent bending energy, see (6.34). This together leads to a superior bending performance of the VE-Q2S element. A similar performance can also be observed in case of small strains.

Sensitivity of the solution with respect to the stabilization parameter A sensitivity study with respect to the choice of the constitutive parameters of the stabilization energy (5.31) is performed next. This study was conducted for 32×32 ($N = 5$) and 64×64 ($N = 6$) meshes.

The outcome is shown in Fig. 6.25 where the results of the VE-Q2S element are compared for different values of $\alpha = \widehat{\Lambda} \,/\, \Lambda$ with the solution obtained by the FE-Q2 element. The value of the shear modulus $\widehat{\mu}$ is given by $\widehat{\mu} = \alpha\, \mu$. In a range of the parameter $0.1 \leq \alpha \leq 1$ the deviations from the solutions obtained with the FE-Q2 finite element are acceptable, being less than 1%. Note that the parameters $\widehat{\Lambda}$ and $\widehat{\mu}$

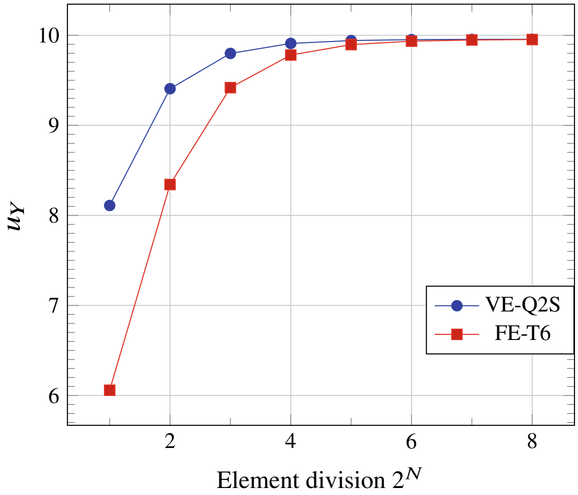


Fig. 6.24 Convergence Study: VE-Q2S and Fe-T6 element

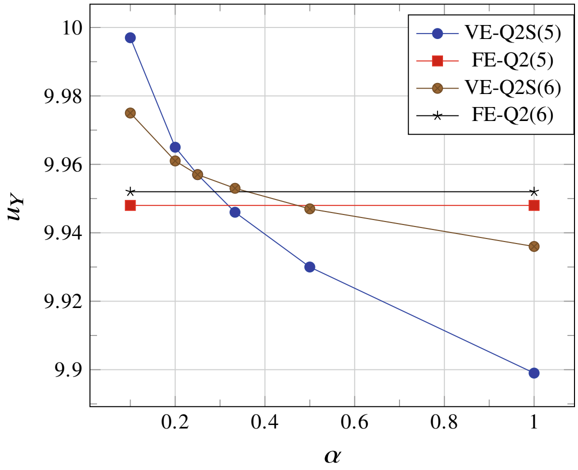


Fig. 6.25 Sensitivity Study for a mesh with $N = 5$ and $N = 6$: VE-Q2S and FE-Q2 element

obtained using the formula (6.34), are different for each element, but are in the range of the smallest deviation from the FE-Q2 results, thus they seem to be optimal.

It should be noted that for $\alpha < 0.1$ the solution is no longer stable, showing light hour-glass modes that become distinct at $\alpha < 0.05$.

6.2.5.2 Thin Beam

In this example virtual element response is investigated for bending situations. Classically, discretization schemes with linear ansatz functions exhibit shear locking and thus provide solutions that are too stiff. However, due to the special choice of the con-

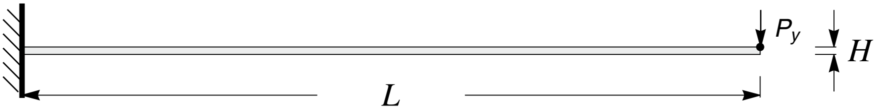


Fig. 6.26 Beam under a point load P_Y; geometry and boundary conditions

stitutive parameters $\widehat{\Lambda}$ and $\widehat{\mu}$ within the energy stabilization it is possible to generate a low order virtual element with good convergence properties in bending dominated structural response. Here the finite elements FE-Q1 and FE-Q2S and the VE-Q2S element are compared.

A beam undergoing large deflections is considered. The beam, depicted in Fig. 6.26, is thin, having a height to length ratio of $H\,/\,L = 1\,/\,100$. It is loaded at the right end by a point load P_Y and clamped at the left end. The constitutive data are selected to be $E = 10^7$ for the Young's modulus and $\nu = 0.3$ for the Poisson ratio which is equivalent to Lamé constants $\Lambda = 5769231$ and $\mu = 384615$. The applied load has the magnitude $P_Y = 48$.

The analysis is performed with virtual elements using a mesh with rectangular shaped elements. Here the 8-node virtual element VE-Q2S is selected which has been already applied in Cook's membrane problem. The results obtained with the virtual element are compared to solutions generated with FE-Q1 and FE-Q2S finite elements. The latter uses a quadratic interpolation for the displacement field and thus has a convergence rate which is one order higher than the convergence rate of the virtual element VE-Q2S. The meshes are selected such that the elements have a length to height ratio larger than 10 in order to demonstrate the applicability of the new virtual element formulation for severe bending situations.

Figure 6.27 reports a mesh convergence study on the basis of a series of meshes with 10 × 1, 20 × 2, 40 × 4, 80 × 6, 160 × 8 and 320 × 10 elements The graphs in Fig. 6.27 depict a convergence behaviour of the VE-Q2S element that is nearly as good as the one of the quadratic FE-Q2S element, despite that it only uses linear ansatz functions. Even for the coarse mesh with only 10 elements the VE-Q2S element deviates only by 3 % from the converged solution. The FE-Q1 element depicts the classical bending locking. Only for a very fine mesh, with 1280 × 14 elements, it converges to the solution.

For the given load the total deflection is computed within one load step using 7 Newton iterations. This leads to the deformation state depicted on the left side of Fig. 6.28. On the right side side of Fig. 6.28 the deflection of the beam is shown for a load that is 10 times larger. In that case 3 load steps with a total of 22 Newton iterations are needed to obtain the final state with the VE-Q2S element. The vertical displacement $u_Y^{10\times 1} = 66.3$ at the beam end deviates for a coarse mesh about 10% from the converged solution $u_Y^{conv} = 72.17$, while a solution $u_Y^{20\times 2} = 70.66$ with a 20×2 mesh is only 2% away from the converged solution.

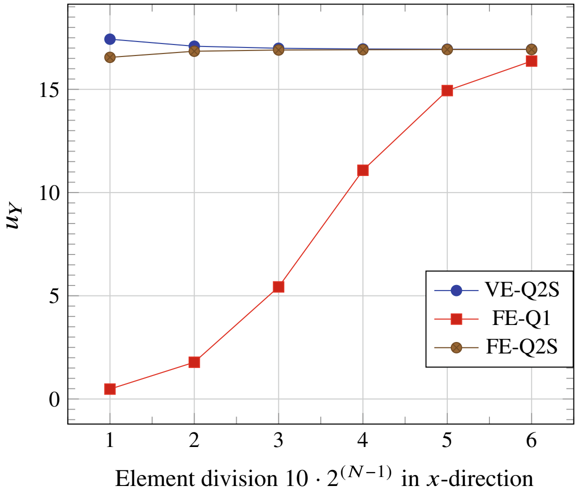


Fig. 6.27 Convergence study for the beam: VE-Q2S-, FE-Q1-, FE-Q2S-element

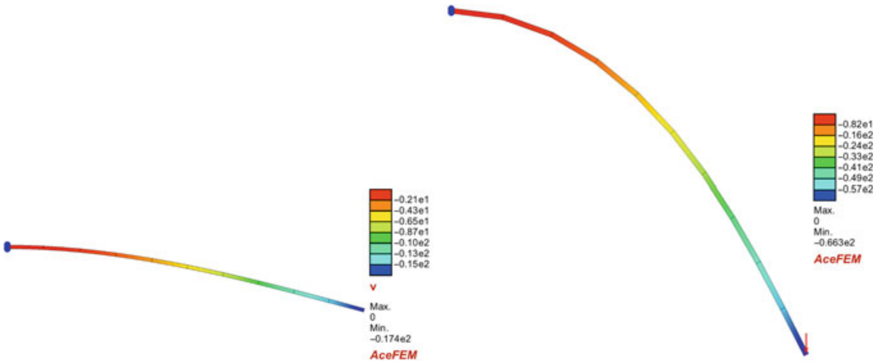


Fig. 6.28 Deflection of the beam for the coarse discretization

6.2.5.3 Block Under Partial Constant Load in Three Dimensions

A cube is subjected to a constant vertical load q on a part of its upper surface, see Fig. 6.29. The geometrical data are given as height $h = 50$, width and depth $a = 100$ and loading surface $a\,/\,2 = 50$. The Young's modulus is selected to be $E = 10$ and the Poisson ratio is set to $\nu = 1\,/3$.

The boundary conditions restrict the displacements at the bottom of the block ($Z = 0 \rightarrow u_Z = 0$) at the top of the block ($z = h \rightarrow u_X = u_Y = 0$). Due to symmetry it is sufficient to model only one quarter of the block. This requires the additional boundary conditions ($X = a\,/\,2 \rightarrow u_X = 0$ and $Y = a\,/\,2 \rightarrow u_Y = 0$).

The cube is discretized using finite and virtual elements. The finite element solutions is obtained with linear and quadratic hexahedral (FE-H1, FE-H2S). Regular

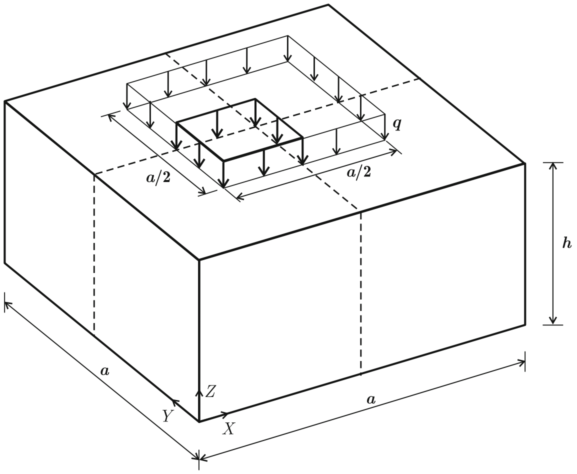


Fig. 6.29 Block under partially constant load q

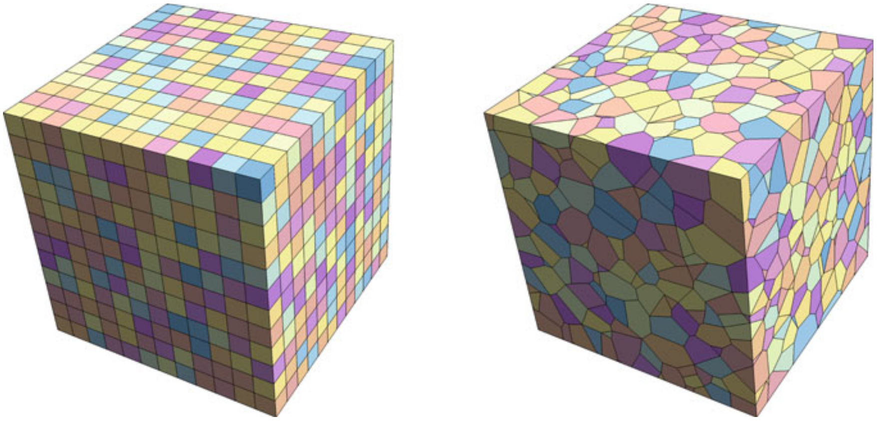


Fig. 6.30 Regular (VE-H1) andVoronoi (VE-VOU) meshes

(VE-H1) and Voronoi meshes (VE-VOU) are employed for the virtual element solutions. The regular and Voronoi meshes are depicted in Fig. 6.30.

The results of the virtual element formulation are computed using the energy stabilization, see Sect. 6.2.3, with a parameter of $\beta = 1 / 3$ for the regular and Voronoi meshes. The load was applied in 5 steps for finite and virtual elements. The Newton algorithm required 6 iteration steps per load step. In total, 30 iterations were needed for both discretizations showing the robustness of the virtual element method. The final deformed configurations using a regular mesh for the finite element discretization (FE-H1) and the virtual element discretization (VE-H1) are depicted in Fig. 6.31. One can observe that both discretizations yield a stable solution despite a very high mesh distortion in the vicinity of the load. Thus the virtual element scheme is able to

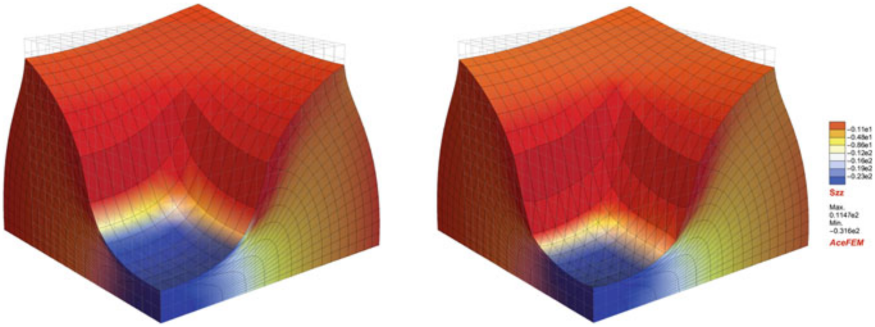
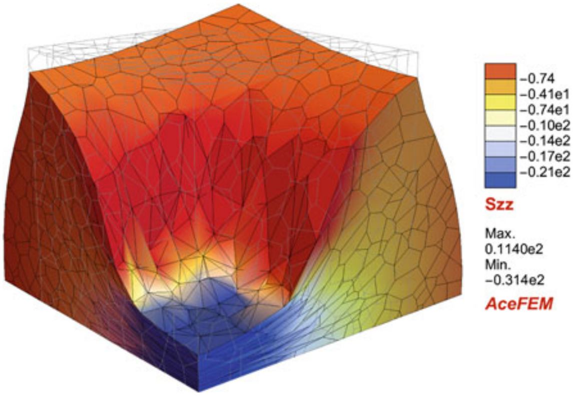


Fig. 6.31 Finite element solutions using finite (FE-H1) and virtual (VE-H1) elements

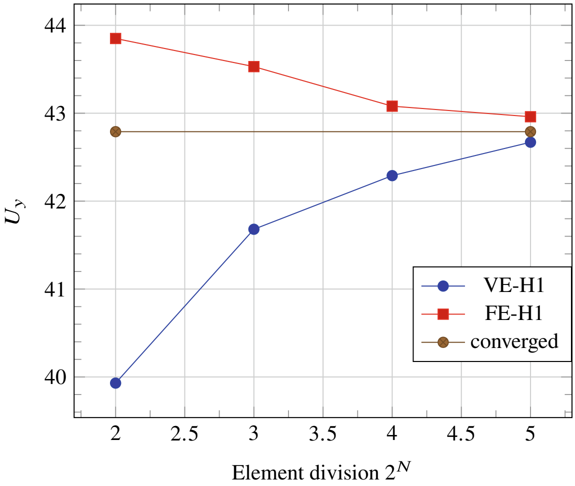
Fig. 6.32 Virtual element solutions (VE-VOU) using a Voronoi mesh



compute numerical solutions as well as finite element schemes for three-dimensional problems undergoing finite deformations. This is also true for a virtual element solution using Voronoi meshes, as depicted in Fig. 6.32. Also here, the elements allow for large distortions and the robustness with respect to the total number of iterations is the same as for the regular meshes. However this is only achieved with the energy stabilization. When using the bi-linear form (6.23), more iterations and load steps are necessary.

Finally the convergence behaviour of the three-dimensional virtual element is demonstrated in Fig. 6.33 in which the results of a FE-H1 finite element are compared with virtual element (VE-H1) solution are compared for a regular H1 mesh. The "converged" solution was computed with a finite element mesh of $N = 6$ with around $8 \cdot 10^5$ unknowns. Both formulations converge smoothly.

Fig. 6.33 Deformation $u_Z(50, 50, 50)$ of the block under constant load q. Convergence study: VE-H1, FE-H1



6.3 Incompressible Elasticity

Materials that do not allow volume change are called incompressible. The formulation of a solid is then based on the inclusion of the volume constraint $J - 1 = 0$, see Sect. 2.3. This can be achieved by

- introduction of a Lagrangian multiplier, the pressure p, that enforces the constraint by the additional term $\int_\Omega p\,[\,J(\boldsymbol{u}) - 1\,]\mathrm{d}V$ in the potential exactly, see (2.89), or
- by using an approximation method, like the penalty approach where the term $\int_\Omega \frac{\varepsilon}{2}\,[\,J(\boldsymbol{u}) - 1\,]^2\,\mathrm{d}V$ is added to the potential. This formulation enforces the incompressibility constraint in a weak sense and leads to a nearly incompressible response, see also (2.90).

Problems with constraints are generally of a mixed nature. Related mixed formulations were discussed in Brezzi et al. (2014) for virtual elements based on the geometrically linear theory of elasticity, in Beirão da Veiga et al. (2018) for Navier-Stokes equations and in Wriggers et al. (2017) for two-dimensional finite strain elasticity problems. The first two articles introduce $H(div)$-spaces to account for the incompressibility constraint Div $\boldsymbol{u} = 0$. Further investigations regarding robustness of divergence free approximations can be found in Frerichs and Merdon (2022). We note that the use of divergence free ansatz spaces is not straightforward in case of the nonlinear constraint $J(\boldsymbol{u}) - 1 = 0$. Thus the following formulations for finite strain elasticity will be based on the use of the spaces L^2 for the pressure field and H^1 for the displacements when enforcing the constraint exactly. This yields Taylor-Hood type elements which are known to be stable for certain ansatz spaces in linear elasticity, see e.g. Brezzi and Fortin (1991).

Strong enforcement of the incompressibility constraint For an exact formulation of the incompressibility constraint the potential (2.89) has to be employed. The first part in this potential can be computed by using the results obtained in the last section. Evaluating the incompressibility constraint by the projection $\boldsymbol{u}_\pi$ leads to

$$U_v^c(\boldsymbol{u}_\pi, p) = \int\limits_{\Omega_v} \left[\Psi^i(\boldsymbol{u}_\pi) - \bar{\boldsymbol{f}} \cdot \boldsymbol{u}_\pi \right] \, d\Omega + \int\limits_{\Omega_v} p \, [J(\boldsymbol{u}_\pi) - 1] \, d\Omega - \int\limits_{\Gamma_e^\sigma} \bar{\boldsymbol{t}} \cdot \boldsymbol{u}_\pi \, d\Gamma \,. \tag{6.67}$$

Here we can use the strain energy (2.60) for the incompressible strain case[10]

$$\Psi^i(\boldsymbol{u}_\pi) = \frac{\mu}{2} [\mathrm{tr}\, \bar{\boldsymbol{C}}(\boldsymbol{u}_\pi) - 3] \tag{6.68}$$

which includes the isochoric Cauchy-Green strain and is free of volume changes. The isochoric right Cauchy-Green strain follows from (2.11) and can be expressed by the projected quantities for the Cauchy-Green strain and the volume change as $\bar{\boldsymbol{C}}(\boldsymbol{u}_\pi) = J_\pi^{-2/3} \, \boldsymbol{C}_\pi$. Thus we have for a virtual element with linear ansatz functions

$$\int\limits_{\Omega_v} \left[\Psi^i(\boldsymbol{u}_\pi) \right] \, d\Omega = \frac{\mu}{2} \left[\mathrm{tr}[\, J_\pi^{-2/3} \, \boldsymbol{C}_\pi] - 3 \right] \, \Omega_v \,. \tag{6.69}$$

Note that the right hand side of (6.69) is still a nonlinear function with respect to the nodal degrees of freedom $\mathbf{u}_v$. In case of higher order ansatz functions the integration of the strain energy (6.68) over the element

$$\int\limits_{\Omega_v} \frac{\mu}{2} \left[\mathrm{tr}[\, J_\pi^{-2/3} \, \boldsymbol{C}_\pi] - 3 \right] \, d\Omega \tag{6.70}$$

has to be performed in the same way as described in Sect. 6.2.1 for the quadratic ansatz.

6.3.1 *Linear Virtual Element with Constant Pressure*

When a linear ansatz space is applied to approximate the displacement field $\boldsymbol{u}_h$ it is best to use a constant pressure p within the virtual element to obtain a locking free behaviour for incompressible problems in solid mechanics. This choice has a long tradition in finite element methods leading to the well known FE-Q1-P0 elements, see e.g. Wriggers (2008). The related virtual element will be derived next.

[10] It is also possible to use a strain energy for a compressible material, like (2.55), since the constraint $J - 1 = 0$ will eliminate in any case the compressible part from the response.

The Jacobian J_π occurs within the incompressibility constraint. It can be computed as $J_\pi = \det \boldsymbol{F}_\pi$. Here we follow a different approach, which was introduced in Chi et al. (2017). For a linear ansatz J_π is constant in Ω_v. Thus, in the two-dimensional case we can use the mass balance analogous to (2.17) and write $J_\pi = \omega_v \,/\, \Omega_v$ where ω_v is the area of the element in the deformed configuration and Ω_v the area with respect to the undeformed configuration. In the spirit of the virtual element method the deformed and undeformed areas are computed using an integral over the boundaries Γ_v and γ_v, respectively:

$$\begin{aligned}
\Omega_v &= \frac{1}{2}\int\limits_{\Gamma_v} \boldsymbol{X}_v \cdot \boldsymbol{N}_v \, \mathrm{d}\Gamma = \frac{1}{2}\sum_{e=1}^{n_E}\int\limits_{\Gamma_e} \boldsymbol{X}_e \cdot \boldsymbol{N}_e \, \mathrm{d}\Gamma \\
\omega_v &= \frac{1}{2}\int\limits_{\gamma_v} \boldsymbol{x}_v \cdot \boldsymbol{n}_v \, \mathrm{d}\gamma = \frac{1}{2}\sum_{e=1}^{n_E}\int\limits_{\gamma_e} \boldsymbol{x}_e \cdot \boldsymbol{n}_e \, \mathrm{d}\gamma \,.
\end{aligned} \tag{6.71}$$

Here Γ_e denotes a straight part of the edge between two vertices, see e.g. Fig. 3.4, of the undeformed element edge while γ_e is the same edge in the deformed configuration of the virtual element. The normal vector $\boldsymbol{N}_e$ is computed using (3.45) for an edge of the virtual element. In the same way the deformed normal $\mathbf{n}_e$ can be computed by inserting in (3.45) the deformed configuration $\boldsymbol{x}_e = \boldsymbol{X}_e + \boldsymbol{u}_e$ where $\boldsymbol{X}_e$ is the vector pointing to the boundary in the undeformed configuration.[11]

As a result, the incompressibility constraint (6.67) can be written

$$\int\limits_{\Omega_v} p\,[\,J(\boldsymbol{u}_\pi) - 1\,]\,\mathrm{d}\Omega = p_v\,[\omega_v(\boldsymbol{u}_\pi) - \Omega_v\,] \tag{6.72}$$

for a constant approximation pressure p_v in each element. The related virtual element, which was described for finite strain hyperelastic material response in Wriggers et al. (2017), fulfills incompressibility at element level and will be called VE-“ET”-I where “ET” is defined in Appendix B.

[11] The same procedure also holds in three dimensions. Only the factor in front of (6.71) changes and the integration has to be carried out over the faces Γ_f. This yields

$$\Omega_v = \frac{1}{3}\int\limits_{\Gamma_v} \boldsymbol{X}_v \cdot \boldsymbol{N}_v \, \mathrm{d}\Gamma = \frac{1}{3}\sum_{e=1}^{n_F}\int\limits_{\Gamma_f} \boldsymbol{X}_f \cdot \boldsymbol{N}_f \, \mathrm{d}\Gamma \,.$$

The same formula can be used for the deformed configuration ω_v, see $(6.71)_2$.

6.3.2 *Quadratic Serendipity Virtual Element with Linear Pressure*

A special mixed serendipity element with a linear ansatz of the pressure can be constructed for incompressible problems. The equivalent finite element is known as Taylor-Hood element (T2-P1), see e.g. Zienkiewicz and Taylor (2000b) and Boffi et al. (2013). This formulation is known to be efficient and robust. It fulfils in the geometrically linear case the Babauska-Brezzi condition which guarantees stability of the element formulation. To meet the Babuska-Brezzi condition a linear ansatz for the pressure p has to be used for virtual elements as well. Analogously to the procedure employed for the approximations of the displacement in a virtual element Ω_v which lead to $\boldsymbol{u}_\pi = \Pi\, \boldsymbol{u}_h$, a projection of the pressure $p_\pi = \Pi\, p_h$ will be applied to project the approximation p_h on a polynomial space. For that the ansatz

$$p_\pi = \mathbf{H}_p(X, Y)\, \bar{\mathbf{p}} \quad \text{with } \mathbf{H}_p(X, Y) = \langle\, 1\; X\; Y\, \rangle \quad \text{and } \bar{\mathbf{p}} = \langle\, a_1\, a_2\, a_3\, \rangle^T \tag{6.73}$$

is introduced. The serendipity element formulation, see Sect. 6.2, and the above approximation for the pressure yields a virtual element with three nodes for the displacement field at each edge γ_k and pressure variables at the vertices, see Fig. 6.34.

Since the pressures are linear the projection described in Sect. 3.1.2 yields the coefficients a_i in terms of the values of the pressure at the vertices $\mathbf{X}_{2k-1}$

$$\int\limits_{\Omega_v} \nabla p_\pi \, \mathrm{d}\Omega = \int\limits_{\Omega_v} \nabla p_h \, \mathrm{d}\Omega \quad \longrightarrow \quad \int\limits_{\Omega_v} \nabla p_\pi \, \mathrm{d}\Omega = \int\limits_{\Gamma_v} p_h\, \boldsymbol{n}\, \mathrm{d}\Gamma, \tag{6.74}$$

and

$$\sum_{k=1}^{n_E} p_\pi(\mathbf{X}_{2k-1}) = \sum_{k=1}^{n_E} p_h(\mathbf{X}_{2k-1}). \tag{6.75}$$

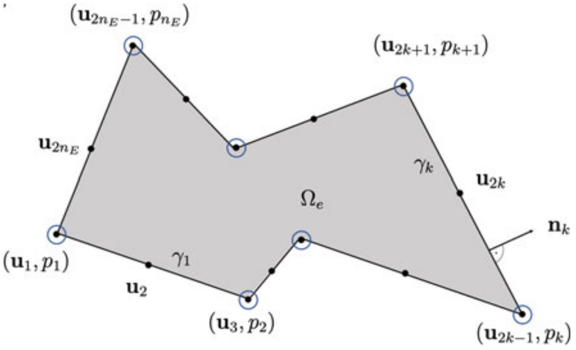


Fig. 6.34 Mixed virtual serendipity T2-P1 element. Displacement and pressure degrees of freedom

By inserting the ansatz (6.73) into (6.74) the gradient and its integral follows as

$$\nabla p_\pi = \begin{Bmatrix} a_2 \\ a_3 \end{Bmatrix}, \qquad \int\limits_{\Omega_v} \nabla p_\pi \, \mathrm{d}\Omega = \Omega_v \begin{Bmatrix} a_2 \\ a_3 \end{Bmatrix}. \tag{6.76}$$

Let us now introduce a linear ansatz for the pressure field $p_h = (1 - \xi)\, p_k + \xi\, p_{k+1}$ along the edge γ_k where p_k are the nodal pressures at the vertices. We further note that the construction of virtual elements is based on straight edges, hence the normal vector $\mathbf{n}_k$ at each edge k is constant, see Fig. 6.34. Insertion of the ansatz in the right hand side of (6.74) yields

$$\int\limits_{\Gamma_v} p_h \boldsymbol{n} \, \mathrm{d}\Gamma = \sum_{k=1}^{n_E} \int\limits_{\gamma_k} [(1 - \xi)\, p_k + \xi\, p_{k+1}]\, l_k \, \mathrm{d}\xi \, \mathbf{n}_k \tag{6.77}$$

where $\xi \in [0, 1]$ is the coordinate along the edge and l_k the length of edge γ_k. The evaluation of (6.77) and insertion into (6.74) leads together with (6.76) to

$$\begin{Bmatrix} a_2 \\ a_3 \end{Bmatrix} = \frac{1}{2\Omega_v} \sum_{k=1}^{n_E} (p_k + p_{k+1})\, l_k \, \mathbf{n}_k = \mathbb{G}_p \, \mathbf{p}_v \,. \tag{6.78}$$

This equation provides a linear map between the nodal pressures p_k and the parameters a_2 and a_3. The associated matrix form on the right side follows by introducing a vector that contains all n_E nodal presures $\mathbf{p}_v = \langle p_1 \; p_2 \; \ldots \; p_{n_V} \rangle^T$ and the matrix

$$\mathbb{G}_p = \frac{1}{2\Omega_v} \Big\langle \, (l_{n_E} \mathbf{n}_{n_V} + l_1 \mathbf{n}_1) \; (l_1 \mathbf{n}_1 + l_2 \mathbf{n}_2) \; \ldots \; (l_{n_V - 1} \mathbf{n}_{n_V - 1} + l_{n_V} \mathbf{n}_{n_V}) \Big\rangle \,. \tag{6.79}$$

Now (6.75) has to evaluated to obtain the coefficient a_1

$$\sum_{k=1}^{n_V} \langle \, 1 \; X_{2k-1} \; Y_{2k-1} \, \rangle \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \sum_{k=1}^{n_V} p_k, \tag{6.80}$$

after some algebra and by defining the vector $\mathbf{I}_s = \langle \, 1 \; 1 \; 1 \; \ldots \; 1 \, \rangle$ the sum $\sum_{k=1}^{n_E} p_k$ can be abbreviated by $\mathbf{I}_s \, \mathbf{p}_e$. With $X_s = \sum_{k=1}^{n_V} X_{2k-1}$ and $Y_s = \sum_{k=1}^{n_V} Y_{2k-1}$ we finally write

$$a_1 = \frac{1}{n_V} \big(\mathbf{I}_s - \langle X_s \; Y_s \rangle \, \mathbb{G}_p \big) \; \mathbf{p}_v \,. \tag{6.81}$$

Combination of the last equation with (6.78) yields the matrix form that determines the coefficients a_i as a function of the nodal pressure values p_k

$$\begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \begin{bmatrix} \frac{1}{n_V}\left(\mathbf{I}_s - \langle X_s\, Y_s \rangle\, \mathbb{G}_p\right) \\ \mathbb{G}_p \end{bmatrix} \mathbf{p}_v = \mathbb{A}_p\, \mathbf{p}_v\,. \tag{6.82}$$

With this map the projected pressure p_π in (6.73) can be linked directly to the nodal pressures $\mathbf{p}_v$

$$p_\pi = \{1\,,\, X\,,\, Y\} \mathbb{A}_p\, \mathbf{p}_v = \mathbf{H}_p(X, Y)\, \mathbb{A}_p\, \mathbf{p}_v\,. \tag{6.83}$$

This result can be inserted into the incompressibility constraint (6.72)

$$\int\limits_{\Omega_v} p_\pi\, [J(\boldsymbol{u}_\pi) - 1]\, \mathrm{d}\Omega = \mathbf{p}_v^T \mathbb{A}_p^T \int\limits_{\Omega_v} [\mathbf{H}_p(X, Y)]^T\, (\det[\mathbf{1} + \mathbb{G}_S(X, Y)\, \mathbf{u}_v] - 1)\, \mathrm{d}\Omega \tag{6.84}$$

where we have used (6.48) leading to the discrete form of the deformation gradient of the serendipity element $\boldsymbol{F}_\pi = \mathbf{1} + \mathbb{G}_S(X, Y)\mathbf{u}_v$, which is linear function of $\boldsymbol{X}$ at the element level and the Jacobian $J_\pi = \det \boldsymbol{F}_\pi = \det[\mathbf{1} + \mathbb{G}_S(X, Y)\mathbf{u}_v]$. The enforcement of incompressibility by this Lagrangian multiplier term leads to a nonlinear function of X , Y and $\mathbf{u}_v$, but can be integrated with Gauss quadrature. It is also possible to use a constant ansatz for p_h along the edge in (6.87). Further details and comparison of the different approaches can be found in Wriggers et al. (2021a).

Remark 6.2 A different discrete form of the incompressibility constraint (6.72) can be developed based on a Taylor series expansion of the pressure at the barycenter $\boldsymbol{X}_b$ of the virtual element Ω_v

$$p_\pi(\boldsymbol{X}_b + \Delta \boldsymbol{X}) = p_\pi(\boldsymbol{X}_b) + \nabla p_\pi(\boldsymbol{X}_b) \cdot \Delta \boldsymbol{X}_b. \tag{6.85}$$

Due the linear ansatz for p_π the gradient ∇p_π is constant in the element and $p_\pi(\boldsymbol{X}_b) = p_{\pi 0}$ is a constant value. Thus we can write with $\Delta \boldsymbol{X}_b = \boldsymbol{X} - \boldsymbol{X}_b$

$$p_\pi(\boldsymbol{X}) = p_{\pi 0} + \nabla p_\pi \cdot (\boldsymbol{X} - \boldsymbol{X}_b)\,. \tag{6.86}$$

Again, a Galerkin projection can be applied to compute ∇p_π

$$\int\limits_{\Omega_v} \nabla p_\pi\, \mathrm{d}\Omega = \int\limits_{\Omega_v} \nabla p_h\, \mathrm{d}\Omega \Longrightarrow \nabla p_\pi = \frac{1}{\Omega_v} \int\limits_{\Gamma_v} p_h \boldsymbol{n}\, \mathrm{d}\Gamma\,. \tag{6.87}$$

The pressure p_h is linear at the edge of the element which leads with (6.77) to

$$\begin{Bmatrix} p_{\pi,x} \\ p_{\pi,y} \end{Bmatrix} = \frac{1}{2\Omega_e} \sum_{k=1}^{n_E} (p_k + p_{k+1})\, l_k\, \mathbf{n}_k\,. \tag{6.88}$$

Thus the pressure field to be inserted in (6.72) follows with (6.78) as

$$p_\pi(\boldsymbol{X}) = p_{\pi 0} + (\boldsymbol{X} - \boldsymbol{X}_b)^T \mathbb{G}_p \, \mathbf{p}_v . \tag{6.89}$$

Similar to (6.84) the Lagrange multiplier term in the potential can be written as

$$\begin{aligned} \int\limits_{\Omega_v} p_\pi \, [J(\boldsymbol{u}_\pi) - 1\,] \, \mathrm{d}\Omega = p_{\pi 0} \int\limits_{\Omega_e} (\det[\mathbf{1} + \mathbb{G}_S(X,Y)\, \mathbf{u}_v] - 1) \;\mathrm{d}\Omega \\ + \mathbf{p}_v^T \mathbb{G}_p^T \int\limits_{\Omega_v} (\boldsymbol{X} - \boldsymbol{X}_b) \; (\det[\mathbf{1} + \mathbb{G}_S(X,Y)\, \mathbf{u}_v] - 1) \;\mathrm{d}\Omega \end{aligned} \tag{6.90}$$

which is different from (6.84). ■

6.3.2.1 Construction of the Quadratic Serendipity Virtual Element with Linear Pressure

For the mixed virtual element a quadratic ansatz is employed for the displacements and a linear ansatz approximates the pressure. The gradient of the displacement field is approximated by a linear part, as discussed in Sect. 6.2 for the serendipity element, and the pressure approximation was introduced above.

The finite deformation formulations of the virtual element method for hyperelstic response of solids has been addressed in the framework of pure displacement elements in Beirão da Veiga et al. (2015), Wriggers et al. (2017) using different stabilizations and for serendipity elements in De Bellis et al. (2019), Wriggers et al. (2021a, b). The mixed potential U in (2.89) is split into the consistency part and a stabilization term. In this contribution we employ the hyperelastic potential functions (2.60)–(2.62) for incompressible elastic material as basis for the constitutive modeling.The potential form for the mixed element is given by

$$U(\boldsymbol{u}, p) = \mathop{\mathsf{A}}_{v=1}^{n_v} \left[U_v^c(\boldsymbol{u}_\pi, p_\pi) + U_v^s(\boldsymbol{u}_h - \boldsymbol{u}_\pi, p_h - p_\pi) \right] \tag{6.91}$$

for a discretization with n_v virtual elements. In the following we will first discuss the formulation of the consistency part $U_v^c(\boldsymbol{u}_\pi, p_\pi)$ that stems from the projection, see last section. Furthermore, a formulation for the stabilization U_v^s of the virtual element method will be considered that was developed in Wriggers et al. (2021a).

6.3.2.2 Consistency Part Due to Projection

The consistency part in the potential (6.91) can be evaluated by inserting the results obtained in Sect. 6.2 and in the last section into the potential (2.89). This yields for a virtual element Ω_v

$$U_v^c(\boldsymbol{u}_\pi, p_\pi) = \int\limits_{\Omega_v} \left[\Psi^i(\boldsymbol{u}_\pi) + p_\pi [J(\boldsymbol{u}_\pi) - 1] - \bar{\boldsymbol{f}} \cdot \boldsymbol{u}_\pi \right] \mathrm{d}\Omega - \int\limits_{\Gamma_v} \bar{\boldsymbol{t}} \cdot \boldsymbol{u}_\pi \mathrm{d}\Gamma \,. \tag{6.92}$$

Here the term related to the pressure p_π can be expressed by (6.72), (6.84) or (6.90). Due to the fact that we are able to compute the projection of the gradient $\nabla\boldsymbol{u}_\pi$ in the initial configuration directly (6.48), we obtain the deformation gradient $\boldsymbol{F}_\pi(\mathbf{u}_v)$ and the Jacobian $J_\pi(\mathbf{u}_v)$. This leads to the deviatoric part of the right Cauchy-Green tensor $\widehat{\boldsymbol{C}}_\pi = J_\pi^{-\frac{2}{3}} \boldsymbol{F}_\pi^T \boldsymbol{F}_\pi$ which enters the strain energy

$$\widehat{\boldsymbol{C}}_\pi(\boldsymbol{u}_v) = \det[\mathbf{1} + \mathbb{G}_S(X,Y)\mathbf{u}_v]^{-\frac{2}{3}} \, [(\mathbf{1} + \mathbb{G}_S(X,Y)\mathbf{u}_v)]^T \, [\mathbf{1} + \mathbb{G}_S(X,Y)\mathbf{u}_v]. \tag{6.93}$$

Strain energy functions of hyperelastic materials, like (2.60), (2.61) and (2.62) can now be used. They have to be computed in terms of the projected deformation measure (6.93) and the projection part of displacements (6.48). All quantities in (6.92) depend only on the projection $\nabla\boldsymbol{u}_\pi$ and the pressure p_π. Note however that (6.92) is a non-linear function with respect to the nodal displacements. Hence the first integral in (6.92) cannot simply be shifted to the boundary and has to be integrated over the area. The integration has to be performed by numerical schemes. As in Sect. 6.2 Gauss quadrature is selected, which means that contributions of energy have to be evaluated at appropriate integration points. This can be done analogous to the compressible case, for details see (6.50). Due to the non-linearity of the strain energy Ψ^i and the Jacobian J, an integration scheme with 12 Gauss points was used for the incompressible case. This number was found to be sufficient in an error analysis related to the integration scheme, see Wriggers et al. (2021b).

The two loading terms in (6.92) have to be considered next. For a constant or polynomial volumetric load $\bar{\boldsymbol{f}}$ the associated integral in (6.92) can be transformed to a boundary integral by using a divergence theorem, see (A.22). The last integral in (6.92) is related to the traction load. The associated surface integral can be directly evaluated at the edges γ_k of the virtual element by transferring it to the local coordinate ξ,

$$\int\limits_{\Gamma_v} \bar{\boldsymbol{t}} \cdot \boldsymbol{u}_h \, \mathrm{d}\Gamma = \sum_{k=1}^{n_E} \int\limits_0^1 \bar{\boldsymbol{t}}_k(\xi) \cdot \boldsymbol{u}_h(\xi) \, l_k \, \mathrm{d}\xi. \tag{6.94}$$

Once the integration of all terms in (6.92) is finalized, the derivations of the potential with respect to the $2\,n_N$ unknown nodal displacements at the vertices of the virtual element $\mathbf{u}_v$ and the n_E pressures $\mathbf{p}_v$ can be carried out leading to the residual vector and tangent matrix. Such approach exploits the fact that $\nabla\boldsymbol{u}_\pi$ depends directly on the nodal displacements, see the mapping (6.48) and that p_π is given explicitly as a function of the nodal values $\mathbf{p}_v$ in (6.83) and (6.89). The displacement and pressure variables are combined in one vector of unknowns $\mathbf{q}_v = \{\mathbf{u}_v \,, \mathbf{p}_v\}$.

All derivations, leading to the residual vector $\mathbf{R}_e^c$ and the tangent matrix $\mathbf{K}_e^c$, can be carried out using the symbolic tool *AceGen*. This yields for (6.92) the residual and tangent

$$\mathbf{R}_v^c = \frac{\partial U_v^c(\nabla \boldsymbol{u}_\pi(\mathbf{u}_v)\,,\boldsymbol{u}_\pi(\mathbf{u}_v),\,p_\pi(\mathbf{p}_v))}{\partial \mathbf{q}_v} \quad \text{and} \quad \mathbf{K}_v^c = \frac{\partial \mathbf{R}_v^c(\mathbf{q}_v)}{\partial \mathbf{q}_v}\,. \tag{6.95}$$

6.3.2.3 Stabilization Techniques for Quadratic Serendipity Virtual Element with Linear Pressure

Here the stabilization by a discrete bi-linear form is employed, as introduced in Sect. 5.2.1, which is directly based on the degrees of freedom. It introduces a point wise error measure, see (6.23), between the nodal values $\mathbf{u}_k$ and the approximation function $\boldsymbol{u}_\pi$ evaluated at the vertices $\boldsymbol{X}_k$, see e.g. Beirão da Veiga et al. (2013, 2015) and Chi et al. (2017).

Note that not only the displacement term, but also the mixed term related to the incompressibility have to be stabilized.

$$U_v^s(\boldsymbol{u}_h - \boldsymbol{u}_\pi,\,p_h - p_\pi) = U_v^u(\boldsymbol{u}_h - \boldsymbol{u}_\pi) + U_v^{pu}(\boldsymbol{u}_h - \boldsymbol{u}_\pi,\,p_h - p_\pi) \tag{6.96}$$

The stabilization $U_v^u(\boldsymbol{u}_h - \boldsymbol{u}_\pi)$ of the serendipity virtual element is based on (6.23) where the stabilization parameter is computed according to (6.53). For details, see Sect. 6.2.

The stabilization for the mixed form $U_s^{pu}(\boldsymbol{u}_h - \boldsymbol{u}_\pi,\,p_h - p_\pi)$ has to correct rank deficiencies in order to fulfill the Babuška-Brezzi condition. Since we are free in choosing the stabilization term, we can introduce the constraint term $p\,\mathrm{Div}\boldsymbol{u}$ from the geometrically linear theory instead of $p(J-1)$. This choice leads to

$$U_v^{pu}(\boldsymbol{u}_h - \boldsymbol{u}_\pi,\,p_h - p_\pi) = \int_{\Omega_v} (p_h - p_\pi)\,\mathrm{Div}(\boldsymbol{u}_h - \boldsymbol{u}_\pi)\,\mathrm{d}\Omega \tag{6.97}$$

However the term $\mathrm{Div}[\boldsymbol{u}_h]$ in Ω_v is not computable and hence an approximation by using the divergence theorem

$$\int_{\Omega_v} (p_h - p_\pi)\,\mathrm{Div}(\boldsymbol{u}_h - \boldsymbol{u}_\pi)\,\mathrm{d}\Omega = \tag{6.98}$$
$$\int_{\Gamma_v} (p_h - p_\pi)\,(\boldsymbol{u}_h - \boldsymbol{u}_\pi)\cdot\boldsymbol{n}\,\mathrm{d}\Gamma - \int_{\Omega_v} (\nabla p_h - \nabla p_\pi)\cdot(\boldsymbol{u}_h - \boldsymbol{u}_\pi)\,\mathrm{d}\Omega$$

follows by neglecting the last term which should be small due to the similarity with (6.74). With the nodal degrees of freedom $\mathbf{q}_v = \{\mathbf{p}_v\,,\mathbf{u}_v\}$, the mappings for the displacement (3.94) and the pressures (6.83) and (6.89) one can write

$$U_v^{pu}(\mathbf{p}_v\,,\mathbf{u}_v) = \beta_p \sum_{k=1}^{n_E} [p_k - p_\pi(\mathbf{X}_k)] \left[\mathbf{u}_k - \mathbb{P}_{uS}^{(2,2)}(\mathbf{X}_k)\mathbf{u}_v\right] \cdot \mathbf{n}_k \tag{6.99}$$

where $\mathbf{n}_k$ is the outward normal of edge k of the virtual element and the scalar value β_p acts as a scaling factor. In Wriggers et al. (2021a) $\beta_p = 1/n_P$ was introduced, with n_P being the number of pressure degrees of freedom of an element. The correct choice of this factor is essential to avoid checkerboard modes for the pressure.

Again the derivations leading to the residual vector $\mathbf{R}_v^{pu}$ and the tangent matrix $\mathbf{K}_v^{pu}$ were performed in the same manner as in (6.95) with the symbolic tool *AceGen*. This leads for (6.99) to the mixed residual and tangent

$$\mathbf{R}_v^s = \frac{\partial U_v^s(\mathbf{u}_v, \mathbf{p}_v)}{\partial \mathbf{q}_v} \quad \text{and} \quad \mathbf{K}_v^s = \frac{\partial \mathbf{R}_v^s(\mathbf{q}_v)}{\partial \mathbf{q}_v}\,. \tag{6.100}$$

The final residual and tangent matrix of the virtual element are given by the sum of expressions (6.95) and (6.100):

$$\mathbf{R}_v = \mathbf{R}_v^c + \mathbf{R}_v^s \quad \text{and} \quad \mathbf{K}_v = \mathbf{K}_v^c + \mathbf{K}_v^s\,. \tag{6.101}$$

6.3.3 Nearly Incompressible Behaviour

If nearly incompressible materials are considered the classical FE-Q1-P0 formulation from the finite element method can be employed leading to a virtual element with linear ansatz for the displacement, as discussed in Sect. 6.2, and constant ansatz for the pressure, see (6.72).

The general formulation is based on the Hu-Washizu principle for the pressure part only, see Simo et al. (1985a). For such formulation two additional unknowns the pressure p and the volume dilatation Θ have to be introduced which relaxes incompressibility constraint $J - 1 = 0$. The constraint term is then given by

$$\int_{\Omega_v} p\,[\,J(\boldsymbol{u}_\pi) - 1\,]\,\mathrm{d}\Omega \Longrightarrow \int_{\Omega_v} \left(p\,[\,J(\boldsymbol{u}_\pi) - \Theta\,] + \frac{K}{2}(\Theta - 1)^2\right)\,\mathrm{d}\Omega \tag{6.102}$$

where K is the bulk modulus. This formulation is classically used in conjunction with low order elements with linear ansatz functions. Here it will be the linear VE-“ET” element. By choosing a constant ansatz Θ_v for the dilatation Θ and a constant pressure p_v the integrals can be evaluated directly, by employing (6.71)

$$\int_{\Omega_v} \left(p\,[\,J(\boldsymbol{u}_\pi) - \Theta\,] + \frac{K}{2}(\Theta - 1)^2\right)\,\mathrm{d}\Omega = p_v\,[\omega_v(\boldsymbol{u}_\pi) - \Theta_v\Omega_v\,] + \frac{K}{2}(\Theta_v - 1)^2\,\Omega_v\,. \tag{6.103}$$

This formulation allows an elimination of the variables p_v and Θ_v at element level and thus does not introduce additional unknowns at the global level of the solution. The element residual vector and tangent matrix are computed as described in (6.100) and (6.101).

Virtual elements that are based on Eq. (6.103) are often used in the analysis of rubber or in J_2-elasto-plasticity where the plastic deformation is assumed to be incompressible $J_p = \det \boldsymbol{F}_p = 1$, see Chap. 8. They will be denoted by VE-ET-P0 in the next chapters.

6.3.4 Numerical Examples, Incompressible Case

The following element types are used for the solution of incompressible and nearly incompressible problems:

- The two-dimensional mixed virtual element with linear ansatz, see (6.67), is denoted as VE-ET-I where ET defines the mesh type as described in Appendix B.
- For nearly incompressible materials the Hu Washizu formulation for the pressure part is used, see (6.103). The associated two-dimensional virtual element is denoted by VE-ET-P0.
- The quadratic Taylor-Hood type virtual element with quadratic ansatz function for the displacements and linear ansatz for the pressure has two variants. Variant (6.84) is named VE-ET-P1c and variant (6.90) VE-ET-P1t, respectively.
- The classical Taylor-Hood mixed triangular finite element FE-T2-P1 with quadratic ansatz for the displacements and linear ansatz for the pressure is used for comparison.

All element abbreviations are supplemented by different indices ET that denote the type of meshes used in an analysis. These will be defined within the sections describing the examples.

Again, all computations are performed by using a Newton-Rapson algorithm with load stepping when necessary. Due to the fact that all formulations are linearized in a consistent manner, using *AceGen*, quadratic convergence is achieved.

6.3.4.1 Cook's Membrane

Cook's membrane problem that was described in Sect. 6.2.5.1, see Fig. 6.20, is investigated now for incompressible response using the potential energy (6.67). The geometry and data are shown on the left side of Fig. 6.35. Two types of meshes are investigated, a regular (Q2S) mesh with eight nodes, see mid part of Fig. 6.35, and an uniform (VOU) Voronoi mesh, see right side of Fig. 6.35. The magnitude of the load is $p_y = 4$. The regular mesh consists of quadrilateral virtual elements with eight nodes. The number of nodes that describe an element varies in the Voronoi mesh

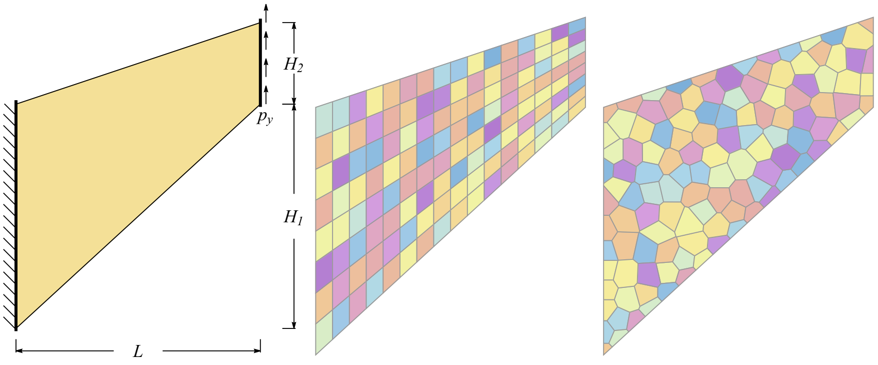
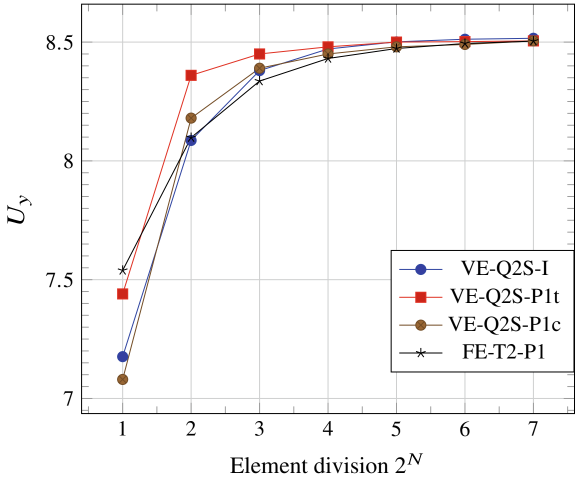


Fig. 6.35 Cook's membrane and different meshes

Fig. 6.36 Incompressible case, convergence study: VE-Q2S-I, VE-Q2S-P1 and FE-T2-P1 elements



from 6 to 16. Due to the incompressible formulation only the shear modulus has to be defined which has the value $\mu = 40$.

Stabilization for the quadratic serendipity mixed virtual elements is based on (6.99) while for the linear element with constant pressure the energy stabilization is employed. The constitutive parameters $\widehat{\Lambda}$ and $\widehat{\mu}$ for the stabilization energy (6.33) were kept as in the examples for the compressible case, see Sect. 6.2.5.1, and computed from (6.34).

The results of the different formulations of mixed virtual elements were compared for different mesh densities with the FE-T2-P1 Taylor-Hood triangular element, see Fig. 6.36. It can be observed that the virtual element VE-Q2S-I can handle the incompressible case almost as well as the FE-T2-P1 element[12] while the VE-Q2S-P1

[12] It shoud be noted that the standard VE-Q2S formulation used in Sect. 6.2.5.1 can be used for nearly incompressible material behaviour with the constitutive parameter $\Lambda \to \infty$. Thus it is locking free.

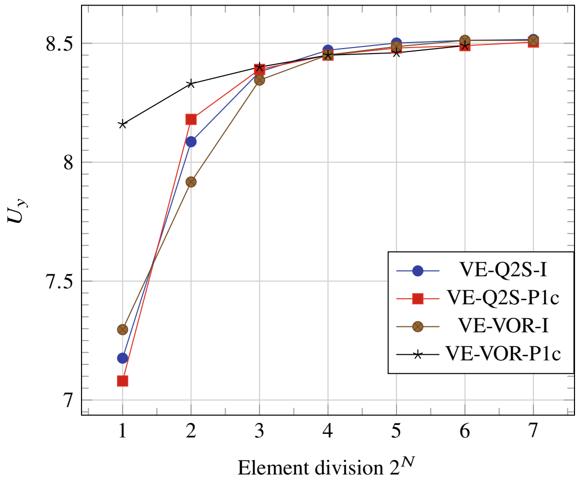


Fig. 6.37 Convergence Study for distorted meshes

elements have a superior coarse mesh accuracy. The FE-T2-P1 element is able to produce the final deformation in just one load step with 5 iterations, independent of the mesh size. Also the VE-Q2S-I and VE-Q2S-P1 elements are able to compute the solution in one load step with 5 iterations, and thus have the same robustness as the FE-T2-P1 element.

Convergence for distorted meshes The effect of irregular Voronoi discretizations has to be investigated in order to see whether the virtual element formulation is robust. For the Cook's membrane problem the Voronoi mesh depicted on the right side of Fig. 6.35 is applied for the analysis using a series of mesh refinements.

This sequence of Voronoi meshes, 2^N with $N \in \{1, 2, 3, 4, 5, 6, 7\}$ is investigated for a incompressible material using the same constitutive parameters as selected above. Figure 6.37 depicts the convergence for different discretizations. The regular elements VE-Q2S-I and VE-Q2S-P1 are compared with the solutions with VE-VOR-I and VE-VOR-P1 based on Voronoi meshes. A very good coarse mesh accuracy can be observed especially for the VE-VOR-P1c element, even in comparison with the solutions based on a regular grid. However for meshes with $N > 4$ all solutions basically lead to the same displacement.

In a final step the load was increased by a factor of 4. A Voronoi mesh with 1017 elements was selected for this application. The load was applied in four steps, leading to a total number of 24 iterations to obtain the deformed state depicted in Fig. 6.38. The displacement related to the total load has the magnitude $u_x = -13.23$ and $u_y =$

However VE-Q2S is in comparison with VE-Q2S-I not as robust as it needs that load steps to obtain the final result. For the Cook's membrane problem this would mean that four load steps, using VE-Q2S, instead of only one load step, using VE-Q2S-I, are needed to obtain the results presented in Fig. 6.36.

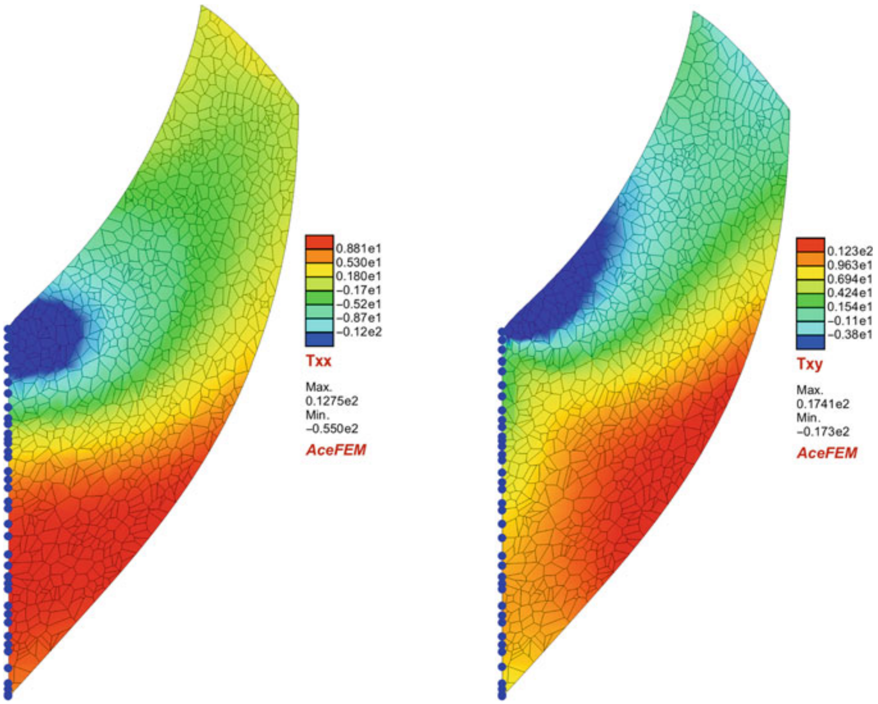


Fig. 6.38 Deformed state and Kirchhoff stresses τ_{xx} and τ_{xy} for a total load of $q_0 = 16$

26.94 at the point (40, 60). Figure 6.38 shows the distribution of the Kirchhoff stresses τ_{xx} and τ_{xy} in the deformed membrane (real scale).

6.3.4.2 Punch Problem, Incompressible

A solid is considered which is loaded by a line load at half of the upper surface, as shown in Fig. 6.39. The applied load leads to severe deformations that are constraint by incompressible behaviour. The finite deformation response can be used to test the robustness of the virtual element formulation.

The geometrical data are $H = L = 1$. The displacements are set to zero along the bottom of the solid, the left and right sides of the solid are fixed in the X-direction, while the top surface is fixed in the X-direction, see also Wulfinghoff et al. (2017). A vertical load of $p_y = 800$ is applied. The Lamé parameter $\mu = 80$ was selected for the linear and quadratic incompressible virtual elements. The punch is discretized by regular (Q2S), see left side of Fig. 6.40, and Voronoi meshes (VOR), see right side of Fig. 6.40.

For a discretization with VE-VOR-I virtual elements the configuration related to the final loading $p_y = 800$ is depicted in Fig. 6.41 for two Voronoi meshes with

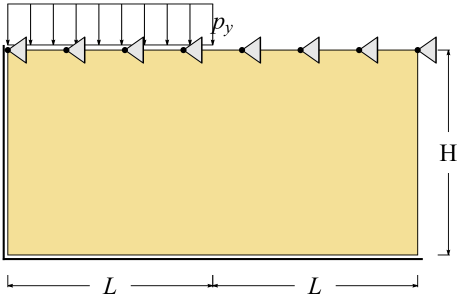


Fig. 6.39 Punch problem—Geometry and boundary conditions

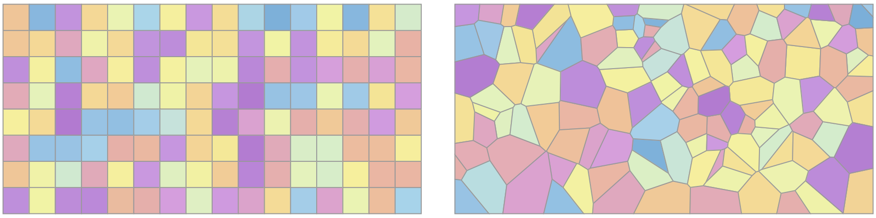


Fig. 6.40 Punch problem—Regular mesh (Q2S) and Voronoi mesh (VOR)

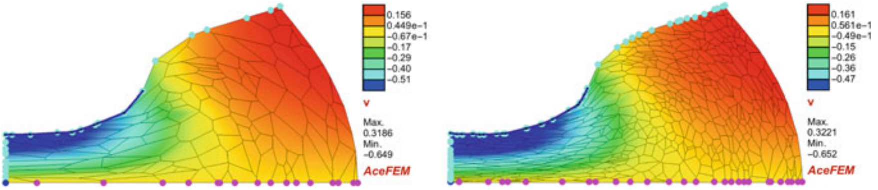


Fig. 6.41 Deformed configuration of the punch problem, incompressible case

different mesh densities $N = 3$ and $N = 5$. The analysis using virtual elements is robust since finer meshes do not influence the overall convergence characteristics and lead to the same number of load steps and iterations.

For the VE-Q2S-I and VE-VOR-I an energy stabilization was utilized with parameter selection based on (6.34). The VE-Q2S-P1 and VE-VOR-P1 elements were stabilized using the bi-linear discrete form according to (6.99).

A convergence study is performed for incompressible deformations of the punch problem using regular, see left side of Fig. 6.40, and Voronoi meshes, see right side of Fig. 6.40. Here the linear and quadratic virtual elements and the classical triangular Taylor-Hood finite element FE-T2-P1 are compared. The regular mesh consists of rectangular 8 node virtual elements here denoted by VE-Q2S-I and VE-Q2S-P1t. The Voronoi mesh is denoted by VE-VOR-I and VE-VOR-P1t.

It can be seen in Fig. 6.42 that the convergence of the virtual element using the Voronoi mesh is not completely smooth for VE-VOR-I. This effect can also

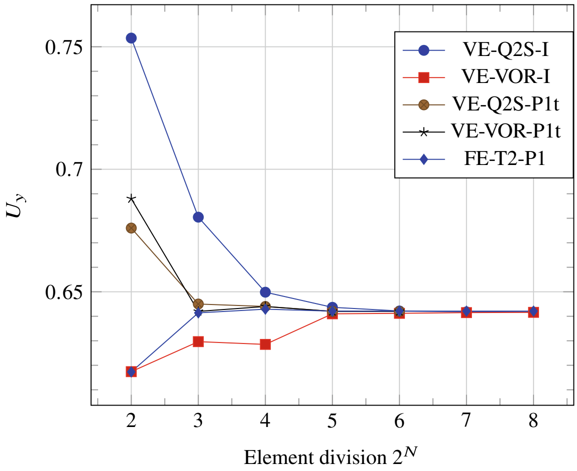


Fig. 6.42 Deformation at the left upper point of the punch, convergence study: VE-Q2S-I, VE-VOR-I, VE-Q2S-P1t, VE-VOR-P1t and FE-T2-P1 elements

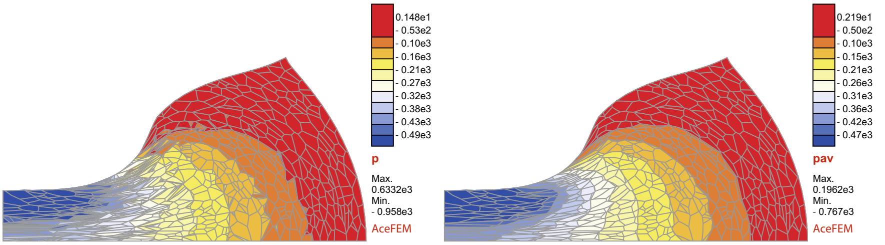


Fig. 6.43 Punch problem: contour plot of the nodal and element averaged pressures for a Voronoi mesh ($N = 5$) using VE-VOR-P1t

be observed for VE-VOR-P1t, however not as pronounced. Such behaviour can be associated with the random nature of the Voronoi meshes. However all results converge to the solution generated with the Taylor-Hood FE-T2-P1 element once the mesh division is $N = 6$. Again the quadratic virtual element demonstrates a very good coarse grid accuracy. For a mesh with ($N = 3$) the solution deviates less than 2% from the converged solution. Thus the VE-P0 and Taylor-Hood type versions of the virtual element can handle incompressible deformations well, while fulfilling the constraint $J = 1$.

For the virtual element VE-VOR-P1t, the pressure distribution is depicted in Fig. 6.43 for a Voronoi mesh of 512 elements. The pressure on the left side of Fig. 6.43 shows the pressure which is plotted directly from the nodal pressure degrees of freedom. The plot exhibits some peaks and irregularities, but no checkerboard modes. The averaged pressure plot on the right side of Fig. 6.43 is smooth. This is also the case for the regular mesh. These results underline that the virtual element method can handle large strain incompressible problems for arbitrary meshes. Further numerical examples that illustrate this behaviour can be found in Wriggers et al. (2021a).

6.3.4.3 Quasi Incompressible Deformation of a Cube

The example discussed in Sect. 6.2.5.3, see Fig. 6.29, is considered for quasi incompressible material with a Poisson ratio of $\nu = 0.4983$. This leads to the Lamé constants $\Lambda = 499.92568$ MPa and $\mu = 1.61148$ Mpa. In addition to the complexity having almost incompressible behaviour, two line singularities arise which are related to the jump in the Neumann boundary conditions at the end of the constant load q, see Fig. 6.29 and the discussion in Schröder et al. (2021).

The convergence of the vertical displacement in Z-direction at the midpoint of the block, $u_Z(50, 50, 50)$, versus the degrees of freedom (DOF) of the discretization was investigated for different finite elements in Schröder et al. (2021). Regular discretizations with $(4 \times 4 \times 4)$, $(8 \times 8 \times 8)$, $(16 \times 16 \times 16)$ and $(32 \times 32 \times 32)$ elements were employed in this benchmark for linear and quadratic displacement as well as mixed elements.

To see how the virtual element based on the mixed principle (6.103) performs in this benchmark, the same mesh structure was used. The results can be found in Table 6.2 where the number of degrees of freedom, n_{dof}, is given for the mesh sequence of H1-elements having 8 nodes. The results for H2 (27 nodes) and H2S (20 nodes) meshes are computed on uniform meshes with the same number of nodes as the corresponding H1-mesh which leads roughly to the same number of unknowns.

The virtual element solutions were compared with the classical hexahedral displacement element FE-H1 with linear ansatz and a hexahedral displacement element FE-H2 with quadratic ansatz. A hexahedral mixed element FE-H1P0 with linear ansatz for the displacement field and discontinuous constant pressure field at element level using the mixed form (6.103) and a sophisticated enhanced element FE-TSCG of mixed type and includes 12 assumed enhanced strain fields and linear displacements, see Korelc et al. (2010), were employed as well.

The results are presented in Table 6.2. Since all the elements were developed using *AceGen*, see Korelc and Wriggers (2016), they exhibit quadratic convergence within the Newton-Raphson solution algorithm. The enhanced strain element FE-TSCG and the FE-H1P0 element are softer than the FE-H2 element. Compared to the finite element responses the virtual elements perform well and converge. It is evident from Table 6.2 that the displacement elements converge from below while this is not always the case for mixed elements, e.g. the FE-TCSG element converges from above. It can be clearly stated that this sophisticated enhanced element has superior performance.

Table 6.2 Mid displacement $u_z(50, 50, 50)$ for different element types

n_{dofs}	VE-H1-I	VE-H2S-I	FE-H1	FE-H2	FE-H1P0	FE-TSCG
260	18.71	23.56	7.78	18.32	19.87	20.14
1800	19.61	21.29	13.17	19.54	20.02	20.10
13328	19.85	20.50	17.54	19.98	20.01	20.03
102432	19.93	20.13	19.52	20.01	20.00	20.01

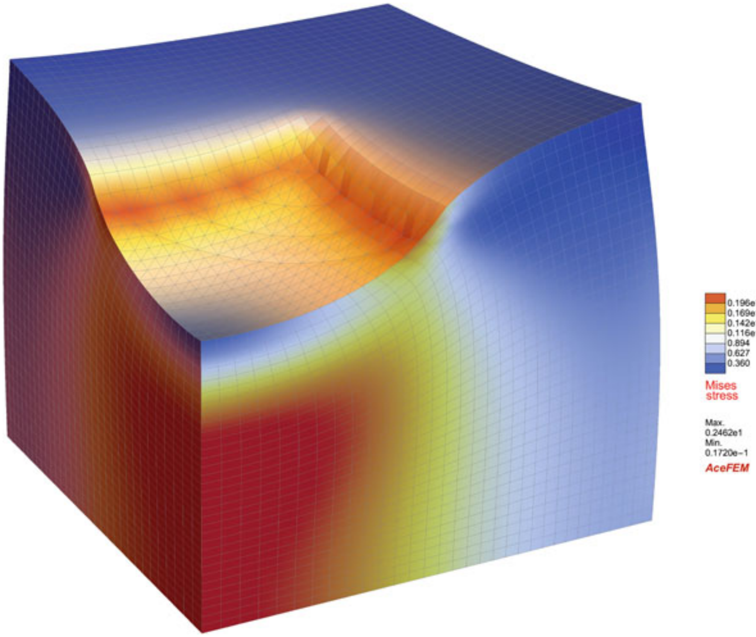


Fig. 6.44 Incompressible punch problem: contour plot of the von Mises stress for a regular H1-mesh (32 × 32) using the virtual element VE-H1-I

However the virtual elements perform as well as the FE-H1P0 element[13] which means that virtual elements using the energy stabilization are a good alternative. This can also be observed for Voronoi meshes.

The von Mises stresses are depicted in Fig. 6.44. Observe that the virtual element VE-H1-I with linear ansatz functions is able, despite only producing constant stresses within the element, to pick up the line singularity at the jump of the applied load from q to zero. This is demonstrated by the high von Mises stress at the end of the load q.

6.4 Anisotropic Elastic Behaviour

Anisotropy has to be considered in many engineering applictions. Anisotropic materials span the range from crystals via composite to biological tissues. Depending on the material class the anisotropic models are different, for an overview, see e.g. Schröder (2009). Here we concentrate on composites that demonstrate transversly isotropic material behaviour. Composites are solids that consists of two different

[13] In many commercial finite element codes the FE-H1P0 is the "work horse", meaning it will be employed for various problems regarding incompressible finite strains and J_2 plasticity.

material phases. One is the matrix phase which is toughened up by fibers that have a specific direction.

The modeling of transversly isotropic materials is based on a split of the strain energy $\Psi = \Psi^{iso} + \Psi^{ti}$ into an isotropic part Ψ^{iso}, see e.g. Sect. 2.2.2.1, and a part Ψ^{ti} that introduces the transversly isotropic constitutive behaviour. Two possibilities are considered in Sect. 2.2.2.3. Transversly isotropic material can be either modeled by a general polyconvex strain energy, see (2.63), or by an inextensibility constraint, see (2.65), which was relaxed in (2.66) by a perturbed Lagrangian form. The latter approach is adequate for numerical simulation of materials with a very high fiber stiffness in comparison with the matrix material, like a rubber tire with steel cords. The two possibilities lead to:

- **Polyconvex strain energy.** The formulation for the virtual element method, based on the transversly isotropic strain energy (2.63), is written for $\gamma = 1$ and for one preferred direction $\boldsymbol{a}$ as

$$\Psi^{ti} = C \left[\frac{1}{\alpha + 1} [\text{tr}(\boldsymbol{C}\, \boldsymbol{M})]^{\alpha+1} + \frac{1}{\beta + 1} [\text{tr}(J^2\, \boldsymbol{C}^{-1} \boldsymbol{M})]^{\beta+1} + J^{-2} \right] \tag{6.104}$$

  with the structure tensor $\boldsymbol{M} = \boldsymbol{a} \otimes \boldsymbol{a}$.
  The essential physical behaviour of a solid is associated with the consistency part of the strain energy within the virtual element method. Due to that Ψ^{ti} is not considered in the stability part. Based on this observation we can write for the potential, in analogy to (6.38),

$$U_c^a(\boldsymbol{u}_\pi) = \int\limits_{\Omega_v} \left[\Psi^{iso}(\nabla \boldsymbol{u}_\pi) + \Psi^{ti}(\nabla \boldsymbol{u}_\pi) - \bar{\boldsymbol{f}} \cdot \boldsymbol{u}_\pi \right] \, d\Omega - \int\limits_{\Gamma_e^\sigma} \bar{\boldsymbol{t}} \cdot \boldsymbol{u}_\pi \, d\Gamma \,. \tag{6.105}$$

  For a linear ansatz the first integral can be evaluated according to (6.41)

$$\begin{aligned} \int\limits_{\Omega_v} \left[\Psi^{iso}(\nabla \boldsymbol{u}_\pi) + \; \Psi^{ti}(\nabla \boldsymbol{u}_\pi) \right] \, d\Omega &= \left[\Psi^{iso}(\nabla \boldsymbol{u}_\pi) + \Psi^{ti}(\nabla \boldsymbol{u}_\pi) \right] \, \Omega_v \\ &= \left[\Psi^{iso}(\mathbb{P}_\nabla^{d,1}\, \mathbf{u}_v) + \Psi^{ti}(\mathbb{P}_\nabla^{d,1}\, \mathbf{u}_v) \right] \, \Omega_v \end{aligned} \tag{6.106}$$

  which yields a constant term that, however, is highly nonlinear with respect to the nodal displacement $\mathbf{u}_v$. Residual and tangent matrix for the consistency part follow according to (6.42).
  For a quadratic element, like the serendipity element, relations (6.46) and (6.50) have to be applied and integration over the interior of an element Ω_v is necessary. The stabilization is exactly the same for linear and quadratc ansatz function, as discussed in Sect. 6.2.2.
- **Inextensibility constraint.** For very stiff fibers in a composite a virtual element formulation can be developed as well. The following perturbed Lagrangian potential describes transversly isotropic responses for stiff fibers, see (2.66)

$$U_c^a(\boldsymbol{u}_\pi, s_a^{ti}) = U_c^{iso}(\boldsymbol{u}_\pi) + \sum_{a=1}^{2} \int\limits_{\Omega} \left(s_a^{ti}\,(\text{tr}[\boldsymbol{C}(\boldsymbol{u}_\pi)\,\boldsymbol{M}_a] - 1) - \frac{1}{2C_{ca}}\,[s_a^{ti}]^2 \right) d\Omega\,. \tag{6.107}$$

where $E_a = \text{tr}[\boldsymbol{C}(\boldsymbol{u}_\pi)\,\boldsymbol{M}_a] - 1)$ is the directional strain, see (2.14) and C_a the fiber stiffness. The value s_a^{ti} relates to the fiber stress and U_c^{iso} represents the isotropic consistency part of the strain energy, see (2.64). The two fiber directions are defined by the structural tensor $\boldsymbol{M}_a = \boldsymbol{a}_a \otimes \boldsymbol{a}_a$, see also (2.14). For a low order ansatz with linear displacements and constant fiber stresses s_a^{ti} at element level $s_a^{ti} = \bar{s}_a^{ti}$ the integration of the last two terms is trivial leading to

$$U_c^a((\boldsymbol{u}_\pi, \bar{s}_a^{ti}) = U_c^{iso}(\boldsymbol{u}_\pi) + \sum_{a=1}^{2} \left(\bar{s}_a^{ti}(\text{tr}[\boldsymbol{C}(\boldsymbol{u}_\pi)\,\boldsymbol{M}_a] - 1) - \frac{1}{2C_{ca}}[\bar{s}_a^{ti}]^2 \right) \Omega_v \tag{6.108}$$

This formulation introduces two additional variables $\bar{s}_a^{ti}$ which can be elimniated at element level. In the above equation the right Cauchy-Green tensor is given by $\boldsymbol{C}(\boldsymbol{u}_\pi) = [\boldsymbol{F}(\boldsymbol{u}_\pi)]^T \boldsymbol{F}(\boldsymbol{u}_\pi)$ with $\boldsymbol{F}(\boldsymbol{u}_\pi) = \mathbf{1} + \nabla \boldsymbol{u}_\pi$. Within the virtual element discretization the deformation gradient can be expressed as

$$\boldsymbol{F}(\boldsymbol{u}_\pi) = \mathbf{1} + \mathbb{P}_\nabla^{d,1}\,\mathbf{u}_v$$

using (3.51) and (3.152) in the two- and three-dimensional case, respectively. Again it is clear that (6.108) is constant over the element but depends in a nonlinear way on the nodal unknowns $\mathbf{u}_v$.

The formulation in (6.108) relates to a perturbed Lagrangian approach which is a variant of the penalty method for enforcing constraints. Thus in case of rigid fibers the stiffness parameter C_{ca} can be viewed as a penalty parameter. With $C_{ca} \to \infty$ the case of inextensible fibers is recovered. This interpretation can be used to model the stiff fibers as a directional material constraint. However an introduction of such constraints can lead to locking behaviour as observed for incompressible materials. Due to the fact that the anisotropic material model imposes constraints, it is advisable to use within a discretization with a linear ansatz of the displacement field a constant approximation of the fiber stress, which is analogous to the VE-ET-I formulation for incompressible materials, see Sect. 6.3.

- **Mixed form.** Another possibility for the formulation of nearly rigid fibers is related to a mixed Hu-Washizu principle. Here the fiber stiffnesses can be included in the form

$$\begin{aligned} U_c^{HW}(\boldsymbol{u}_\pi, s_a^{ti}, \delta_a) &= U_c^c(\boldsymbol{u}_\pi) \\ &+ \sum_{a=1}^{2} \int\limits_{\Omega} \left(s_a^{ti}\frac{1}{2}(\text{tr}[\boldsymbol{C}(\boldsymbol{u}_\pi)\,\boldsymbol{M}_a] - \delta_a) + \frac{K_{ca}}{2}(\delta_a - 1)^2 \right) d\Omega \end{aligned} \tag{6.109}$$

where δ_a corresponds to stretches in fiber direction and K_{ca} is the stiffness of the fiber. Again a constant ansatz $\bar{\delta}_a$ is selected for this new variable which results at element level to four additional variables, $s_a^{ti} = \bar{s}_a^{ti}$ and $\delta_a = \bar{\delta}_a$ for each anisotropic direction $\boldsymbol{a}_a$. These four variables can be eliminated at the level of the virtual element Ω_v. Since all variables are constant for a linear ansatz the integrals in (6.109) can be evaluated directly without using numerical integration. Implementation within the virtual element method is the same as in the formulation above employing for the discretization of $\nabla \boldsymbol{u}_\pi$ (3.51) and (3.152) in the two- and three-dimensional case, respectively.

All derivations with respect to the unknown displacements leading to the residual vector $\mathbf{R}_v^c$ and the tangent matrix $\mathbf{K}_{Tv}^c$ were performed with the symbolic tool *AceGen*, see Korelc and Wriggers (2016). This yields for (6.38)

$$\mathbf{R}_v^c = \frac{\partial U_c^a(\mathbf{u}_v)}{\partial \mathbf{u}_v} \quad \text{and} \quad \mathbf{K}_{Tv}^c = \frac{\partial \mathbf{R}_v^c(\mathbf{u}_v)}{\partial \mathbf{u}_v} \tag{6.110}$$

where $\mathbf{u}_v$ are the nodal displacements of the virtual element Ω_v. The residual and tangent for the incompressible case are evaluated similarly.

6.4.1 Numerical Examples, Anisotropic Case

Two examples are considered in this section to illustrate the performance of the low order virtual element formulation when compared with existing finite elements. The examples are subjected to loads that lead to finite deformation strain states and thus the depicted deformation states are not scaled. Solutions based on (6.105) are labelled VE-ET-A and the ones based on (6.108) are denoted by VE-ET-C.

6.4.1.1 Cook's Membrane with Anisotropy

The Cook's membrane is solved as a three-dimensional solid. The membrane is depicted in Fig. 6.45 together with the geometric data. The thickness of the membrane is $t = 10$. It is subjected to a constant load on the right side of magnitude p_0 and clamped at the left end: $X = 0 \rightarrow u_X = u_Y = u_Z = 0$.

The material behaviour is transversely isotropic with one preferred direction $\boldsymbol{a} = \{\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\}$. For the isotropic strain energy we use in this example (2.59) with

$$c_3\, g(\det \boldsymbol{C}) = \varepsilon_1 \left([\det \boldsymbol{C}]^{\varepsilon_2} + [\det \boldsymbol{C}]^{-\varepsilon_2} - 2 \right) - (6c_1 + 12c_2) \log \sqrt{\det \boldsymbol{C}}$$

while the transversely isotropic part is given by (6.104). The parameters for the isotropic part are selected as $c_1 = 42$, $c_2 = 84$, $\varepsilon_1 = 100$ and $\varepsilon_2 = 10$. The material

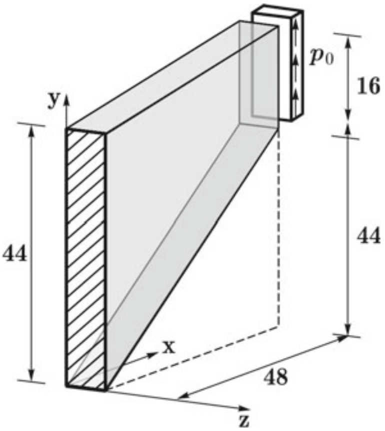


Fig. 6.45 Cook's membrane problem for anisotropic material

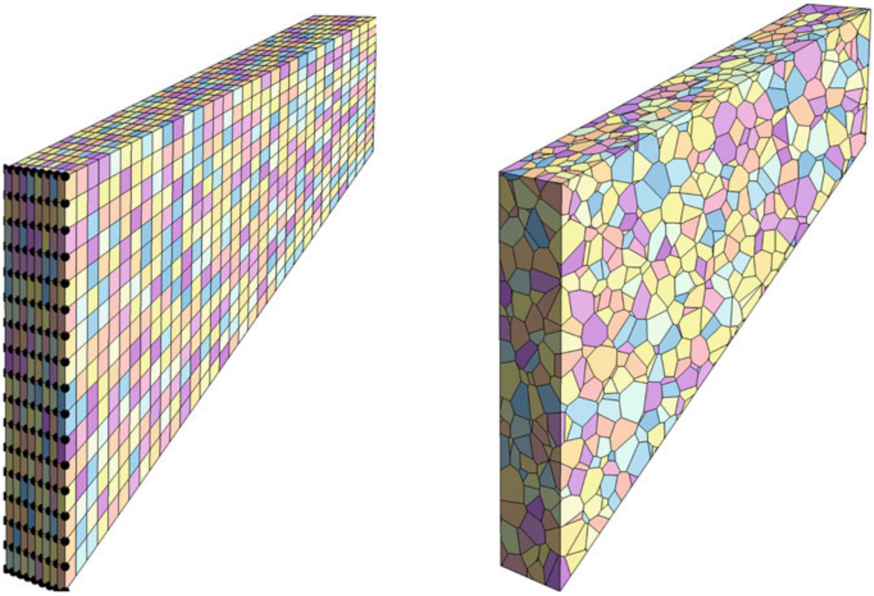


Fig. 6.46 Regular (VE-H1-A) and Voronoi (VE-VOR-A) meshes

parameters for the transversely isotropic strain energy are given by $C = 3000, \alpha = 4$ and $\beta = 8$.

Two different meshes are used to compute the response of Cook's membrane with three-dimensional virtual elements using a linear ansatz. These consist of a regular mesh with eight noded hexahedra (H1) shown on the left side of Fig. 6.46 and a Voronoi mesh made of cells with randomly distributed size on the right hand side.

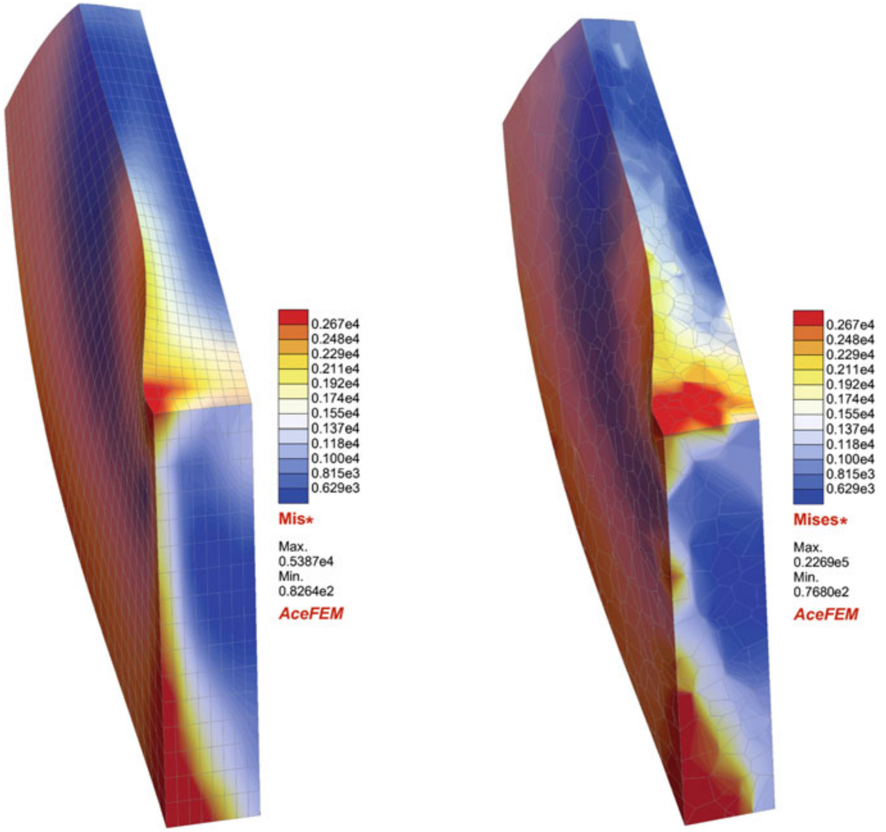


Fig. 6.47 Cook's membrane von Mises stresses for the VE-H1-A and VE-VOR-A elements

The load is applied in 10 steps up to $p_0 = 10^3$ which yields finite deformations and a twisting of the structure can be observed. The deformations are depicted together with the von Mises stress contours in Fig. 6.47. The regular discretization yields smoothed stresses, as expected. Since the stresses are constant within each element the irregular Voronoi mesh demonstrates larger deviations. However the general stress distribution is quite similar as well as the deformed shape of both discretizations. Again the virtual element is robust and allows the prediction of complex deformation states for structures with anisotropic response.

6.4.1.2 Bias-Extension Test, Anisotropic Behaviour

Tension locking related to anisotropic behaviour can occur in structures where stiff fibers are embedded in a very soft matrix. This can be illustrated by considering a rectangular specimen under tension loading, see Fig. 6.48. The fibers are oriented in

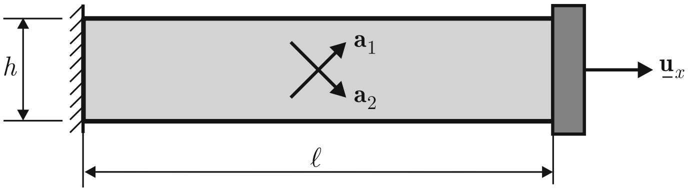


Fig. 6.48 Bias extension test of a woven composite

±45° in the initial configuration. This bias-extension test was used in ten Thjie and Akkerman (2008) and Hamila and Boisse (2013) to investigate behaviour of standard finite element formulations and special interpolation techniques to avoid locking.

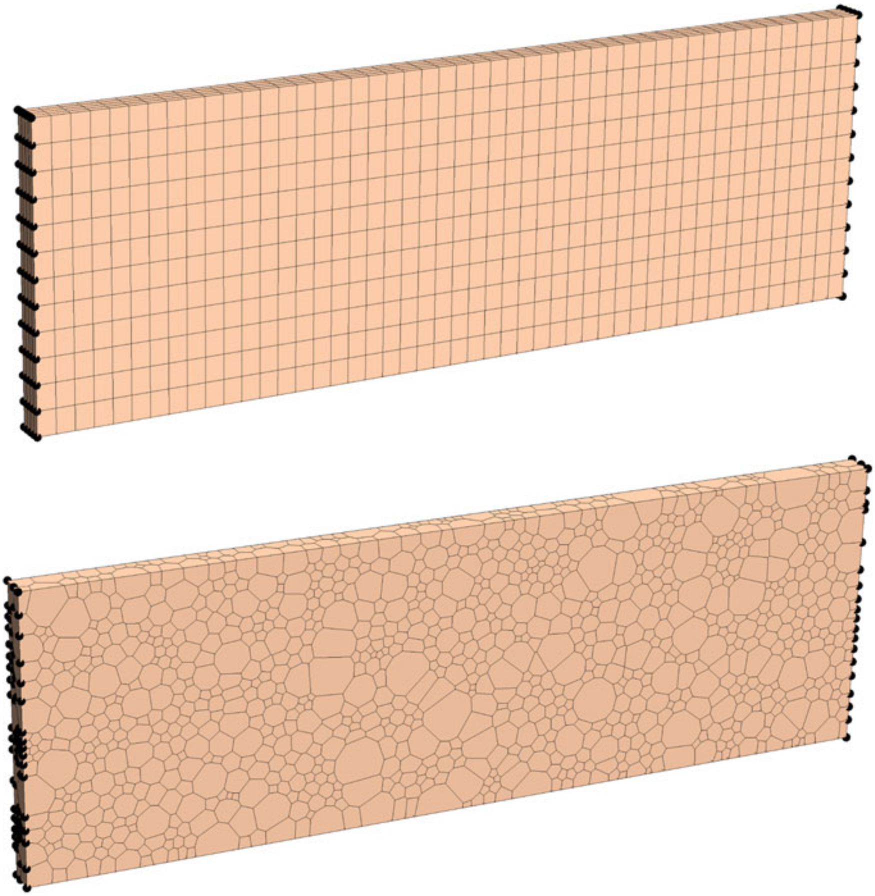


Fig. 6.49 Regular (VE-H1-C) and Voronoi (VE-VOR-C) virtual element mesh

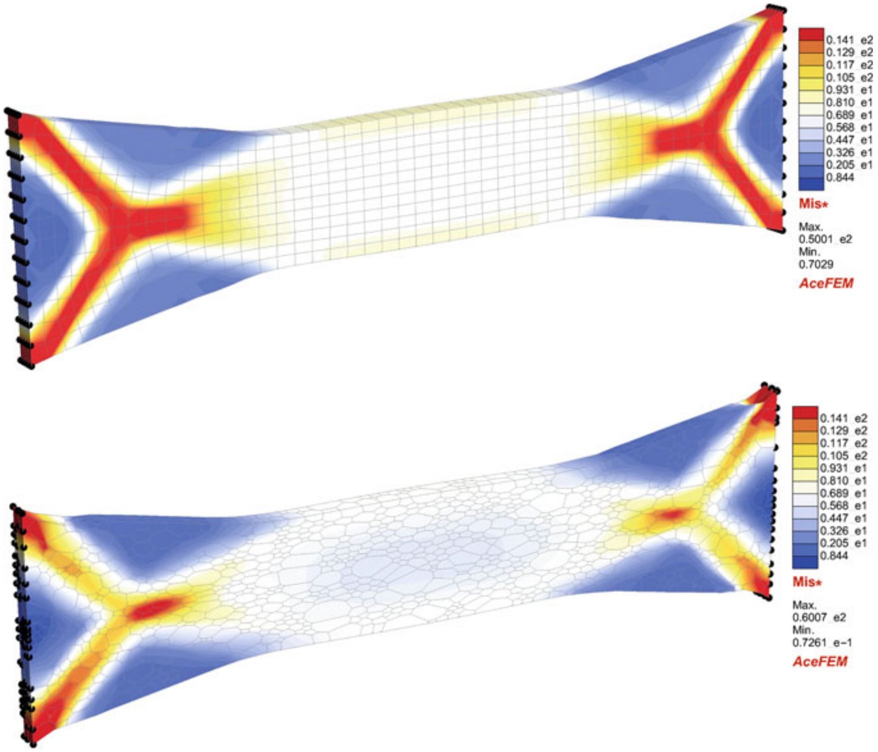


Fig. 6.50 Von Mises stresses for both meshes

The length of the specimen is $L = 300$, its width is $H = 100$ and the thickness of the specimen is $T = 1$. The specimen is clamped at both ends and pulled using a constant displacement $\bar{u}_x = 65$. The material properties of the matrix material are described by the Lamé constants $\Lambda = 1$ and $\mu = 1$. The stiffness for both fibers is $C_{c1} = C_{c2} = 4000$.

The numerical solution of the bias-extension test was obtained with a three-dimensional formulation. The load was applied in one step and the Newton-Raphson method was used to obtain the solution of this nonlinear problem.

The mesh for the three-dimensional numerical simulation can be found in Fig. 6.49. The Voronoi mesh is generated with randomly distributed element sizes.

The deformed state of the specimen after loading can be observed in Fig. 6.50 which cleary shows that the specimen undergoes finite deformations for the applied stretch. The results in Fig. 6.50 depict also the von Mises stresses within the specimen. It can be observed that the stress concentration shown by the red color is less prominent in the VE-VOR-C mesh when compared with the regular VE-H1-C result. Also the results using the VE-VOR-C mesh are nonsymmetric which is due to the randomly distributed Voronoi cells. Hence a refinement is needed to recover a symmetric stress state and a more distinct stress concentration.

Generally it can be concluded that the virtual element formulation for anisotropic materials is locking free and thus can even be applied to problems with a large ratio in the matrix and fiber stiffness for finite deformations.

References

Artioli, E., L. Beirão da Veiga, C. Lovadina, and E. Sacco. 2017a. Arbitrary order 2d virtual elements for polygonal meshes: Part I, elastic problem. *Computational Mechanics* 60: 355–377.

Artioli, E., S. De Miranda, C. Lovadina, and L. Patruno. 2020c. A dual hybrid virtual element method for plane elasticity problems. *ESAIM: Mathematical Modelling and Numerical Analysis* 54(5): 1725–1750.

Artioli, E., S. De Miranda, C. Lovadina, and L. Patruno. 2017. A stress/displacement virtual element method for plane elasticity problems. *Computer Methods in Applied Mechanics and Engineering* 325: 155–174.

Beirão da Veiga, L., C. Lovadina, and G. Vacca. 2018. Virtual elements for the Navier-Stokes problem on polygonal meshes. *SIAM Journal on Numerical Analysis* 56(3):1210–1242.

Beirão da Veiga, L., F. Brezzi, A. Cangiani, G. Manzini, L. Marini, and A. Russo. 2013. Basic principles of virtual element methods. *Mathematical Models and Methods in Applied Sciences* 23(01): 199–214.

Beirão da Veiga, L., C. Lovadina, and D. Mora. 2015. A virtual element method for elastic and inelastic problems on polytope meshes. *Computer Methods in Applied Mechanics and Engineering* 295: 327–346.

Beirão da Veiga L., F. Brezzi, and L. Marini. 2013. Virtual elements for linear elasticity problems. *SIAM, Journal of Numerical Analysis* 51: 794–812.

Berbatov, K., B.S. Lazarov, and A.P. Jivkov. 2021. A guide to the finite and virtual element methods for elasticity. *Applied Numerical Mathematics* 169: 351–395.

Boffi, D., F. Brezzi, and M. Fortin. 2013. *Mixed Finite Element Methods and Applications*, vol. 44. Berlin: Springer.

Botti, M., D.A. Di Pietro, and P. Sochala. 2017. A hybrid high-order method for nonlinear elasticity. *SIAM Journal on Numerical Analysis* 55 (6): 2687–2717.

Brenner, S.C., Q. Guan, and L.Y. Sung. 2017. Some estimates for virtual element methods. *Computational Methods in Applied Mathematics* 17 (4): 553–574.

Brezzi, F., and M. Fortin. 1991. *Mixed and Hybrid Finite Element Methods*. Berlin, Heidelberg, New York: Springer.

Brezzi, F., R.S. Falk, and L.D. Marini. 2014. Basic principles of mixed virtual element methods. *ESAIM: Mathematical Modelling and Numerical Analysis* 48(4): 1227–1240.

Chen, A., and N. Sukumar. 2023a. Stabilization-free serendipity virtual element method for plane elasticity. *Computer Methods in Applied Mechanics and Engineering* 404: 115784.

Chen, A., and N. Sukumar. 2023b. Stabilization-free virtual element method for plane elasticity. *Computer Methods in Applied Mechanics and Engineering* 404: 115784.

Chi, H., L. Beirão da Veiga, and G. Paulino, Some basic formulations of the virtual element method (VEM) for finite deformations. *Computer Methods in Applied Mechanics and Engineering* 318: 148–192 (2017).

Cook, R.D., D.S. Malkus, and M.E. Plesha. 1989. *Concepts and Applications of the Finite Element Analysis*, 3rd ed. Hoboken: Wiley.

D'Altri, A.M., de Miranda S., Patruno L., and Sacco E. 2021. An enhanced VEM formulation for plane elasticity. *Computer Methods in Applied Mechanics and Engineering* 376: 113663.

Dassi, F., C. Lovadina, and M. Visinoni. 2020. A three-dimensional Hellinger-Reissner virtual element method for linear elasticity problems. *Computer Methods in Applied Mechanics and Engineering* 364: 112910.

De Bellis, M., P. Wriggers, and B. Hudobivnik. 2019. Serendipity virtual element formulation for nonlinear elasticity. *Computers & Structures* 223: 106094.

Frerichs, D., and C. Merdon. 2022. Divergence-preserving reconstructions on polygons and a really pressure-robust virtual element method for the stokes problem. *IMA Journal of Numerical Analysis* 42 (1): 597–619.

Gain, A.L., C. Talischi, and G.H. Paulino. 2014. On the virtual element method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes. *Computer Methods in Applied Mechanics and Engineering* 282: 132–160.

Hamila, N., and P. Boisse. 2013. Locking in simulation of composite reinforcement deformations. Analysis and treatment. *Composites: Part A* 109–117.

Hansbo, P., and M.G. Larson. 2003. Discontinuous galerkin and the crouzeix–raviart element: Application to elasticity. *ESAIM: Mathematical Modelling and Numerical Analysis* 37(1): 63–72.

Hudobivnik, B., F. Aldakheel, and P. Wriggers. 2018. Low order 3d virtual element formulation for finite elasto-plastic deformations. *Computational Mechanics* 63: 253–269.

Korelc, J., and P. Wriggers. 2016. *Automation of Finite Element Methods*. Berlin: Springer.

Korelc, J. 2000. Automatic generation of numerical codes with introduction to AceGen 4.0 symbolc code generator. http://www.fgg.uni-lj.si/Symech.

Korelc, J., U. Solinc, and P. Wriggers. 2010. An improved EAS brick element for finite deformation. *Computational Mechanics* 46: 641–659.

Korelc J. Acegen, AceFEM and AceShare. 2023. http://symech.www.fgg.uni-lj.si.

Kouhia, R., and R. Stenberg. 1995. A linear nonconforming finite element method for nearly incompressible elasticity and stokes flow. *Computer Methods in Applied Mechanics and Engineering* 124 (3): 195–212.

Krysl, P. 2015. Mean-strain eight-node hexahedron with optimized energy-sampling stabilization for large-strain deformation. *International Journal for Numerical Methods in Engineering* 103: 650–670.

Krysl, P. 2016. Mean-strain 8-node hexahedron with optimized energy-sampling stabilization. *Finite Elements in Analysis and Design* 108: 41–53.

Kwak, D.Y., and H. Park. 2022. Lowest-order virtual element methods for linear elasticity problems. *Computer Methods in Applied Mechanics and Engineering* 390: 114448.

Lamperti A., M. Cremonesi, U. Perego, A. Russo, and C. Lovadina. 2023. A Hu–Washizu variational approach to self-stabilized virtual elements: 2D linear elastostatics. *Computational Mechanics* 1–21.

Li, L., H. Su, and Y. He. 2022. Penalty virtual element method for the 3d incompressible flow on polyhedron mesh. *Entropy* 24 (8): 1129.

Lovadina, C., and M. Visinoni. 2022. Virtual element methods for three-dimensional hellinger-reissner elastostatic problems. *Communications in Applied and Industrial Mathematics* 13 (1): 57–69.

Mengolini, M., M.F. Benedetto, and A.M. Aragón. 2019. An engineering perspective to the virtual element method and its interplay with the standard finite element method. *Computer Methods in Applied Mechanics and Engineering* 350: 995–1023.

Mirtich, B. 1996. Fast and accurate computation of polyhedral mass properties. *Journal of Graphics Tools* 1 (2): 31–50.

Mora, D., and G. Rivera. 2020. A priori and a posteriori error estimates for a virtual element spectral analysis for the elasticity equations. *IMA Journal of Numerical Analysis* 40 (1): 322–357.

Mueller-Hoeppe, D.S., S. Loehnert, and P. Wriggers. 2009. A finite deformation brick element with inhomogeneous mode enhancement. *International Journal for Numerical Methods in Engineering* 78: 1164–1187.

Nadler, B., and M. Rubin. 2003. A new 3-d finite element for nonlinear elasticity using the theory of a cosserat point. *International Journal of Solids and Structures* 40: 4585–4614.

Pian, T.H.H., and K. Sumihara. 1984. Rational approach for assumed stress finite elements. *International Journal for Numerical Methods in Engineering* 20: 1685–1695.

Schröder, J. 2009. Anisotropic polyconvex energies. In *Polyconvex Analysis*, vol 62, ed. J. Schröder, 1–53. Wien: CISM, Springer.

Schröder, J., T. Wick, S. Reese, P. Wriggers, R. Müller, S. Kollmannsberger, M. Kästner, A. Schwarz, M. Igelbüscher, N. Viebahn, et al. 2021. A selection of benchmark problems in solid mechanics and applied mathematics. *Archives of Computational Methods in Engineering* 28: 713–751.

Simo, J.C., and M.S. Rifai. 1990. A class of assumed strain methods and the method of incompatible modes. *International Journal for Numerical Methods in Engineering* 29: 1595–1638.

Simo, J.C., R.L. Taylor, and K.S. Pister. 1985. Variational and projection methods for the volume constraint in finite deformation elasto-plasticity. *Computer Methods in Applied Mechanics and Engineering* 51: 177–208.

Ten Thjie, R.H.W. and R. Akkerman. Solutions to intra-ply shear locking in finite element analyses of fibre reinforced materials. *Composites: Part A* 1167–1176 (2008).

van Huyssteen, D., and B.D. Reddy. 2020. A virtual element method for isotropic hyperelasticity. *Computer Methods in Applied Mechanics and Engineering* 367: 113134.

Wriggers, P., M. De Bellis, and B. Hudobivnik. 2021. A Taylor-Hood type virtual element formulation for large incompressible strains. *Computer Methods in Applied Mechanics and Engineering* 385: 114021.

Wriggers, P., B. Hudobivnik, and F. Aldakheel. 2021. NURBS-based geometries: A mapping approach for virtual serendipity elements. *Computer Methods in Applied Mechanics and Engineering* 378: 113732.

Wriggers, P. 2008. *Nonlinear Finite Elements*. Berlin, Heidelberg, New York: Springer.

Wriggers, P., B. Reddy, W. Rust, and B. Hudobivnik. 2017. Efficient virtual element formulations for compressible and incompressible finite deformations. *Computational Mechanics* 60: 253–268.

Wulfinghoff, S., H.R. Bayat, A. Alipour, and S. Reese. 2017. A low-order locking-free hybrid discontinuous Galerkin element formulation for large deformations. *Computer Methods in Applied Mechanics and Engineering* 323: 353–372.

Yu, Y. 2022. mVEM: A MATLAB software package for the virtual element methods. arXiv:2204.01339.

Zhang, B., J. Zhao, Y. Yang, and S. Chen. 2019. The nonconforming virtual element method for elasticity problems. *Journal of Computational Physics* 378: 394–410.

Zienkiewicz, O.C., and R.L. Taylor. 2000. *The Finite Element Method,* vol. 2, 5th ed. Oxford, UK: Butterworth-Heinemann.

Chapter 7
Virtual Elements for Problems in Dynamics

Many discretization schemes—like finite differences, finite element and boundary element methods—can be applied to solve problems related to linear and nonlinear dynamics. Here, the major additional challenge, when compared to static analysis, is the modeling of a solution in time which needs related discretizations and algorithms, see e.g. Hughes (1987), Bathe (2006), Wriggers (2008) for finite element analysis. So far the applications of virtual elements in dynamics are limited. The computation of mass matrices was discussed in Beirão da Veiga et al. (2014). Some early work related to engineering can be found in Park et al. (2019) who used explicit integrations schemes. In Mazzia et al. (2020) the virtual element method was applied to solve transient processes in the environment related to the dynamics of transitional landforms. In Adak et al. (2022) the dynamics of nonlocal plate was investigated. Cihan et al. (2021a, b) employed virtual elements for nonlinear dynamic problems including finite strains and plasticity. Eigenvalue problems were treated with the virtual element method in Boffi et al. (2022). Furthermore, virtual elements were applied within the discrete element method to analyse granular media with flexible particles, see Gay Neto et al. (2021).

This chapter discusses details of the extension of the virtual element method to dynamics for small and finite strain response in solid mechanics.

7.1 Continuum Formulation

The theoretical background for applications in elasto-dynamics can be found in Chap. 2 which leads to a set of equations that can be classified as hyperbolic partial differential equations. We recall that all data are functions of time. For a well-posed initial-boundary value problem initial conditions must be specified for the displacement $\boldsymbol{u}(\boldsymbol{X}, t)$ and the velocity $\dot{\boldsymbol{u}}(\boldsymbol{X}, t)$. Together with the balance of momentum (2.19), the Dirichlet and von Neuman boundary conditions (2.20), (2.21) and the initial conditions the strong form of the initial-boundary value problem can be stated with respect to the initial configuration Ω

P. Wriggers et al., *Virtual Element Methods in Engineering Sciences*,
https://doi.org/10.1007/978-3-031-39255-9_7

$$\begin{aligned} \varrho_0\, \ddot{\boldsymbol{u}}(\boldsymbol{X},t) &= \operatorname{Div} \boldsymbol{P}(\boldsymbol{X},t) + \varrho_0\, \bar{\boldsymbol{b}}(\boldsymbol{X},t) && \text{in } \Omega \\ \boldsymbol{u}(\boldsymbol{X},t) &= \bar{\boldsymbol{u}}(\boldsymbol{X},t) && \text{on } \Gamma_D \\ \boldsymbol{P}(\boldsymbol{X},t)\boldsymbol{N}(\boldsymbol{X},t) &= \bar{\boldsymbol{t}}(\boldsymbol{X},t) && \text{on } \Gamma_N \\ \boldsymbol{u}(\boldsymbol{X},0) &= \boldsymbol{u}_0(\boldsymbol{X}) && \text{in } \Omega \\ \dot{\boldsymbol{u}}(\boldsymbol{X},0) &= \dot{\boldsymbol{u}}_0(\boldsymbol{X}) && \text{in } \Omega \end{aligned} \tag{7.1}$$

where the first Piola-Kirchhoff stress tensor is given as a function of the strains via the constitutive relations in Sect. 2.2. For the given data ($\varrho_0\, \bar{\boldsymbol{b}}\,,\, \bar{\boldsymbol{u}}\,,\, \bar{\boldsymbol{t}}\,,\, \boldsymbol{u}_0\,,\, \dot{\boldsymbol{u}}_0$) one has to find the displacement $\boldsymbol{u}(\boldsymbol{X},t)$ as a solution of the above set of equations.

The corresponding weak form of the stationary case, see Sect. 2.3, has to be complemented by the inertia contribution leading to

$$(\varrho_0\, \ddot{\boldsymbol{u}}\,,\, \boldsymbol{v}) + a(\boldsymbol{u}\,,\, \boldsymbol{v}) = f(\boldsymbol{v}) \tag{7.2}$$

where $a(\boldsymbol{u}\,,\, \boldsymbol{v})$ and $f(\boldsymbol{v})$ are defined in (2.85) and the inertia term is given by

$$(\varrho_0\, \ddot{\boldsymbol{u}}\,,\, \boldsymbol{v}) = \int\limits_{\Omega} \varrho_0\, \ddot{\boldsymbol{u}} \cdot \boldsymbol{v}\; \mathrm{d}\Omega\,. \tag{7.3}$$

Equation (7.2) is the basis for the formulation of the virtual element discretization. All virtual element formulations regarding the stationary case $a(\boldsymbol{u}\,,\, \boldsymbol{v}) = f(\boldsymbol{v})$ can be used, see e.g. Sect. 6 for elasticity. The only additional quantity which has to be addressed is the inertia term together with a proper time integration scheme.

When using automatic differentiation to derive the discretization schemes and associated software, see Korelc and Wriggers (2016), it is often more efficient to start from a potential. This of course is not always possible, like in this case or in plasticity. A way out is to use a the *pseudo*-potential formulation which was already discussed in Sect. 2.3.3 for plasticity. Such potential can be constructed for the inertia term

$$U(\boldsymbol{u}, \ddot{\boldsymbol{u}}) = \int\limits_{\Omega} \left[\varrho_0\, \ddot{\boldsymbol{u}} \cdot \boldsymbol{u} + \Psi(\nabla \boldsymbol{u}) - \bar{\boldsymbol{f}} \cdot \boldsymbol{u} \,\right] \mathrm{d}\Omega - \int\limits_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \boldsymbol{u}\; \mathrm{d}\Gamma\,. \tag{7.4}$$

The *pseudo*-potential includes a real potential related to the elastic variables in the strain energy Ψ and the loading terms. The inertia term is designed in a *pseudo*-potential fashion: during the first variation of the potential with respect to the displacement field, the acceleration has to be kept constant

$$\left.\frac{\mathrm{d}U}{\mathrm{d}\boldsymbol{u}}\right|_{\ddot{u}=const.} \cdot \delta \boldsymbol{u} = \int\limits_{\Omega} \left[\varrho_0\, \ddot{\boldsymbol{u}} \cdot \delta \boldsymbol{u} + \boldsymbol{P}(\nabla \boldsymbol{u}) \cdot \delta \boldsymbol{F} - \bar{\boldsymbol{f}} \cdot \delta \boldsymbol{u} \,\right] \mathrm{d}\Omega - \int\limits_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \delta \boldsymbol{u}\; \mathrm{d}\Gamma\,. \tag{7.5}$$

Here the variation of the strain energy function was derived using

$$\delta\Psi = \frac{\partial \Psi}{\partial \boldsymbol{F}} \cdot \delta \boldsymbol{F} = \boldsymbol{P} \cdot \delta \boldsymbol{F}$$

with $\delta \boldsymbol{F} = \nabla \delta \boldsymbol{u}$ and $\boldsymbol{P}$ being the fist Piola-Kirchhoff stress tensor. Due to the equivalence of the variation $\delta \boldsymbol{u}$ and the test function $\boldsymbol{v}$ the result of the first variation in (7.5) is equal to the weak form (7.2).[1]

Since the displacement $\boldsymbol{u}(\boldsymbol{X}, t)$ is a function of space and time a discretization scheme is needed that takes into account as well a spatial discretization as a discretization in time. Classically, first a spatial discretization is performed, which can be derived for virtual elements based on the results of Sect. 5 leading to a linear or nonlinear matrix formulation, still depending on time. The resulting set of ordinary differential equations can be solved by time-stepping algorithms of different order and type, see e.g. Hughes (1987), Bathe (2006), Wriggers (2008).

7.2 Mass Matrix

Since the virtual element formulation, based on the potential Ψ and the loading terms, is already described in Chaps. 5 and 6 for linear and nonlinear elastic problems, the only remaining task is to discretize the inertia term $(\varrho_0\, \ddot{\boldsymbol{u}}\,, \boldsymbol{v})$ in the weak form (7.2) or the *pseudo*-potential $(\varrho_0\, \ddot{\boldsymbol{u}}\,, \boldsymbol{u})$ in (7.5).

The ansatz space that was discussed e.g. in Sect. 3.1 has to be written in a form that separates the variables with respect to space and time. The inertia term does not depend on a gradient in (7.4), but directly on the displacement and the acceleration. Hence the ansatz based on the assumption of the relaxed space from Sect. 3.1.2 should be used, see (3.37) and Beirão da Veiga et al. (2014). With that we can approximate the function $\boldsymbol{u}_h$ by $\boldsymbol{u}_\pi$ and a remainder leading to

$$\boldsymbol{u}_h = \boldsymbol{u}_\pi + (\boldsymbol{u}_h - \boldsymbol{u}_\pi) \quad \text{and } \ddot{\boldsymbol{u}}_h = \ddot{\boldsymbol{u}}_\pi + (\ddot{\boldsymbol{u}}_h - \ddot{\boldsymbol{u}}_\pi)\,. \tag{7.6}$$

Insertion of this ansatz into the inertia term in (7.4) yields for a virtual element Ω_v

$$\int\limits_{\Omega_v} \varrho_0\, \ddot{\boldsymbol{u}} \cdot \boldsymbol{u}\; \mathrm{d}\Omega = \int\limits_{\Omega_v} \varrho_0\ \ [\ddot{\boldsymbol{u}}_\pi + (\ddot{\boldsymbol{u}}_h - \ddot{\boldsymbol{u}}_\pi)] \cdot [\boldsymbol{u}_\pi + (\boldsymbol{u}_h - \boldsymbol{u}_\pi)]\; \mathrm{d}\Omega\,. \tag{7.7}$$

[1] Of course, it is possible to start from the Hamilton principle $\int_{t_1}^{t_2} (T - U_e + W)\, \mathrm{d}t \rightarrow STAT$ to arrive at Eq. (7.5) by its variation. In that case $T = \frac{1}{2} \int_\Omega \varrho_0\, \dot{\boldsymbol{u}} \cdot \dot{\boldsymbol{u}}\, \mathrm{d}t$ is the kinetic energy, $U_e = \int_\Omega \Psi(\boldsymbol{u})\; \mathrm{d}\Omega$ the strain energy and $W = \int_\Omega \bar{\boldsymbol{f}} \cdot \delta \boldsymbol{u}\; \mathrm{d}\Omega + \int_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \delta \boldsymbol{u}\; \mathrm{d}\Gamma$ is the potential of the external forces. However the *pseudo*-potential (7.4) leads to the same weak form and yields from the numerical point of view a more efficient implementation.

By using the definition of the relaxed space it follows from (3.37)

$$\int\limits_{\Omega_v} X_i^n \, (\boldsymbol{u}_\pi - \boldsymbol{u}_h) \, \mathrm{d}\Omega = 0 \tag{7.8}$$

for polynomials X_i^n up to order $n \leq 2$. Since $\boldsymbol{u}_\pi$ has the same ansatz order as X_i^n, see e.g. the derivation of (3.35), and the projected displacement $\boldsymbol{u}_\pi = \mathbf{H}_u^{d,n}(X, Y) \, \mathbf{u}_v(t)$ has same ansatz function as the projected acceleration $\ddot{\boldsymbol{u}}_\pi = \mathbf{H}_u^{d,n}(X, Y) \, \ddot{\mathbf{u}}_v(t)$ the mixed terms in the scalar product in (7.7) vanish which leads to the simplification

$$\int\limits_{\Omega_v} \varrho_0 \, \ddot{\boldsymbol{u}} \cdot \boldsymbol{u} \, \mathrm{d}\Omega = \int\limits_{\Omega_v} \varrho_0 \;\; [\ddot{\boldsymbol{u}}_\pi \cdot \boldsymbol{u}_\pi + (\ddot{\boldsymbol{u}}_h - \ddot{\boldsymbol{u}}_\pi) \cdot (\boldsymbol{u}_h - \boldsymbol{u}_\pi)] \; \mathrm{d}\Omega \, . \tag{7.9}$$

As in the stationary case, the inertia term consists of two integrals. The first part is related to the consistency term and the second part can be viewed as the stabilization term. Beirão da Veiga et al. (2014) noted that the inertia term in (7.9) only needs to be stabilized when the initial-boundary values problem is reaction dominated. Such cases were discussed in e.g. Ahmad et al. (2013) in the context of the virtual element method. However using only the consistency parts leads to rank deficient element mass matrices which e.g. cannot be applied within explicit integration schemes where the mass matrix has to be inverted.

By introducing the ansatz functions, constructed in Sect. 3, in (7.9) it is possible to compute the consistency part of the mass matrix for different ansatz orders n. This is illustrated below for linear and quadratic ansatz functions.

Two-dimensional case. The virtual element ansatz for the two-dimensional case was provided in detail in (3.58) for the linear interpolations and in (3.84) for the quadratic case. The results yield linear and quadratic ansatz functions which depend on the nodal degrees of freedom $\mathbf{u}_v$ of the virtual element

$$\boldsymbol{u}_\pi(\boldsymbol{X}, t) = \mathbf{H}_u^{(2,1)}(X, Y) \, \mathbb{P}_u^{(2,1)} \, \mathbf{u}_v(t) \, , \tag{7.10}$$

$$\boldsymbol{u}_\pi(\boldsymbol{X}, t) = \mathbb{P}_u^{(2,2)}(X, Y) \, \mathbf{u}_v(t) \, . \tag{7.11}$$

Here a separation of variables was introduced which split the ansatz in a spatial part, represented by the ansatz functions $\mathbf{H}_u^{(2,1)}(X, Y)$ and $\mathbb{P}_u^{(2,2)}(X, Y)$, which do not depend on time, and a time depending part that is related to the nodal unknowns $\mathbf{u}_v(t)$.

By inserting these functions into the consistency part of (7.9) we obtain for the linear case the matrix form

$$\int\limits_{\Omega_v} \varrho_0 \; \ddot{\boldsymbol{u}}_\pi \cdot \boldsymbol{u}_\pi \; \mathrm{d}\Omega = \mathbf{u}_v^T \; [\mathbb{P}_u^{(2,1)}]^T \left(\int\limits_{\Omega_v} \varrho_0 \, [\mathbf{H}_u^{(2,1)}(X, Y)]^T \, \mathbf{H}_u^{(2,1)}(X, Y) \; \mathrm{d}\Omega \right) \mathbb{P}_u^{(2,1)} \, \ddot{\mathbf{u}}_v \tag{7.12}$$

leading to the mass matrix

$$\mathbf{M}_{\pi}^{(2,1)} = [\mathbb{P}_u^{(2,1)}]^T \left(\int\limits_{\Omega_v} \varrho_0 \, [\mathbf{H}_u^{(2,1)}(X,Y)]^T \, \mathbf{H}_u^{(2,1)}(X,Y) \, \mathrm{d}\Omega \right) \mathbb{P}_u^{(2,1)} \, . \tag{7.13}$$

In the same way we obtain for the quadratic case

$$\int\limits_{\Omega_v} \varrho_0 \; \ddot{\boldsymbol{u}}_{\pi} \cdot \boldsymbol{u}_{\pi} \, \mathrm{d}\Omega = \mathbf{u}_v^T \left(\int\limits_{\Omega_v} \varrho_0 \, [\mathbb{P}_u^{(2,2)}(X,Y)]^T \mathbb{P}_u^{(2,2)}(X,Y) \, \mathrm{d}\Omega \right) \ddot{\mathbf{u}}_v \tag{7.14}$$

with the mass matrix

$$\mathbf{M}_{\pi}^{(2,2)} = \int\limits_{\Omega_v} \varrho_0 \, [\mathbb{P}_u^{(2,2)}(X,Y)]^T \mathbb{P}_u^{(2,2)}(X,Y) \, \mathrm{d}\Omega \, . \tag{7.15}$$

Note that in the quadratic case the vector of nodal displacements $\mathbf{u}_v$ also contains the internal degrees of freedom related to the moments. The integrals are functions of X and Y and can be evaluated exactly using a transformation to the boundary Γ_v, see Appendix A.

As an example, the explicit form of the integral (7.13), representing the linear ansatz, is presented

$$\int\limits_{\Omega_v} \varrho_0 \, [\mathbf{H}_u^{(2,1)}(X,Y)]^T \, \mathbf{H}_u^{(2,1)}(X,Y) \, \mathrm{d}\Omega = \int\limits_{\Omega_v} \varrho_0 \begin{bmatrix} 1 & 0 & X & 0 & Y & 0 \\ 0 & 1 & 0 & X & 0 & Y \\ X & 0 & X^2 & 0 & XY & 0 \\ 0 & X & 0 & X^2 & 0 & XY \\ Y & 0 & XY & 0 & Y^2 & 0 \\ 0 & Y & 0 & XY & 0 & Y^2 \end{bmatrix} \mathrm{d}\Omega \, . \tag{7.16}$$

The integration can be easily performed for an arbitrary polygon by using the formulae (A.2) to (A.7) in Appendix A. The rank of this matrix is 6 which will not change for the complete element mass matrix in (7.13) where the transformation with the projection matrix $\mathbb{P}_u^{(2,1)}$ corrects the dimensions and relates the mass matrix to the nodal degrees of freedom but does not change the number of non-zero eigenvalues of $\mathbf{M}_{\pi}^{(2,1)}$. As a result the linear mass matrix (7.13) has the size $(2n_V \times 2n_V)$ but the rank 6, where n_V is the number of vertices of the virtual element.

In the same way the quadratic ansatz yields a mass matrix with rank 12 while it has the size of $(4n_V \times 4n_V + 2)$, see also Sect. 3.1.1. Here polynomials up to fourth order have to be integrated. An exact evaluation can be based on (A.24) or by using (A.8).

Three-dimensional case. Ansatz function for linear three-dimensional virtual elements were derived Sect. 3.2 and are summarized in (3.157) as a function of the nodal unknowns

$$\boldsymbol{u}_\pi(\boldsymbol{X},t) = \mathbf{H}_u^{(3,1)}(X,Y,Z)\,\mathbb{P}_u^{(3,1)}\,\mathbf{u}_v(t)\,. \tag{7.17}$$

This yields the matrix formulations

$$\int\limits_{\Omega_v} \varrho_0\;\ddot{\boldsymbol{u}}_\pi\cdot\boldsymbol{u}_\pi\;\mathrm{d}\Omega \tag{7.18}$$

$$= \mathbf{u}_v^T\;[\mathbb{P}_u^{(3,1)}]^T \left(\;\int\limits_{\Omega_v} \varrho_0\,[\mathbf{H}_u^{(3,1)}(X,Y,Z)]^T\;\mathbf{H}_u^{(3,1)}(X,Y,Z)\;\mathrm{d}\Omega\right)\;\mathbb{P}_u^{(3,1)}\;\ddot{\mathbf{u}}_v$$

with the mass matrix

$$\mathbf{M}_\pi^{(3,1)} = [\mathbb{P}_u^{(3,1)}]^T \left(\;\int\limits_{\Omega_v} \varrho_0\,[\mathbf{H}_u^{(3,1)}(X,Y,Z)]^T\;\mathbf{H}_u^{(3,1)}(X,Y,Z)\;\mathrm{d}\Omega\right)\;\mathbb{P}_u^{(3,1)} \tag{7.19}$$

where now the volume integral for polynomials up to second order, see (7.16) has to be evaluated over arbitrary polyhedra. Since the ansatz uses the polynomial $(1\,,X\,,Y\,,Z)$ in three coordinate directions the matrix in the integral has the size 12×12 which leads to a matrix of rank 12. Hence the linear three-dimensional virtual element has a mass matrix with 12 non-zero eigenvalues while its total size is $(3n_V\times 3n_V)$. It is easy to note that the special case of a tetrahedron with four vertices will have a full rank mass matrix.

The evaluation of the integrals for the mass matrix, e.g. (7.19) for the three-dimensional case can be obtained in an exact manner by using an integration over the faces similar to the way presented in Sect. 3.2.3 or by using (A.25). However, also approximations are possible. Cihan et al. (2021a) demonstrated that the evaluation of the integral in (7.19) at the centroid $\boldsymbol{X}_b$ of the virtual element yields sufficiently accurate results. Clearly, such approximation needs less computational effort when compared with other evaluation schemes. Based on this simplification the mass-matrix (7.19) is given by

$$\mathbf{M}_{\pi\,,c}^{(3,1)} = [\mathbb{P}_u^{(3,1)}]^T\;\big(\varrho_0\,[\mathbf{H}_u^{(3,1)}(\boldsymbol{X}_b)]^T\;\mathbf{H}_u^{(3,1)}(\boldsymbol{X}_b)\big)\;\mathbb{P}_u^{(3,1)}\;\Omega_v\,. \tag{7.20}$$

Stabilization of the mass matrix: The stabilization term of the mass matrix is the second term in (7.9). The integral can be approximated in the same manner as in e.g. Sect. 6.1.3, see also Park et al. (2019),

$$\int\limits_{\Omega_v} \varrho_0(\ddot{\boldsymbol{u}}_h - \ddot{\boldsymbol{u}}_\pi) \cdot (\boldsymbol{u}_h - \boldsymbol{u}_\pi) \mathrm{d}\Omega = \varrho_0 \, \Omega_v \, \sum_{i=1}^{n_V} [\mathbf{u}_i - \boldsymbol{u}_\pi(\mathbf{X}_i)]^T \, [\ddot{\mathbf{u}}_i - \ddot{\boldsymbol{u}}_\pi(\mathbf{X}_i)] \tag{7.21}$$

where $\mathbf{u}_i$ is the nodal displacement that is contained in the vector $\mathbf{u}_v$ at element level and the projection $\boldsymbol{u}_\pi$ has to be evaluated at the node i of the virtual element. Furthermore, Ω_v is the element area or volume in the two- or three-dimensional case, respectively. By writing

$$\int\limits_{\Omega_v} \varrho_0(\ddot{\boldsymbol{u}}_h - \ddot{\boldsymbol{u}}_\pi) \cdot (\boldsymbol{u}_h - \boldsymbol{u}_\pi) \, \mathrm{d}\Omega = \mathbf{u}_v^T \, \mathbf{M}_s^{(d,n)} \, \ddot{\mathbf{u}}_v \tag{7.22}$$

where d stands for dimension and n for polynomial order, we can define the stabilization matrix for different ansatz functions. Linear and quadratic ansatz functions lead to:

- **Two-dimensional linear ansatz**:

$$\mathbf{M}_s^{(2,1)} = \varrho_0 \, \Omega_v \, \sum_{i=1}^{n_V} \left[\mathbf{I} - \mathbf{H}_u^{(2,1)}(\mathbf{X}_i) \, \mathbb{P}_u^{(2,1)}\right]^T \left[\mathbf{I} - \mathbf{H}_u^{(2,1)}(\mathbf{X}_i) \, \mathbb{P}_u^{(2,1)}\right] \tag{7.23}$$

  where $\mathbf{I}$ is the identity matrix and (7.10) was used.
- **Two-dimensional quadratic ansatz**:

$$\mathbf{M}_s^{(2,2)} = \varrho_0 \, \Omega_v \, \sum_{i=1}^{2n_V} \left[\mathbf{I} - \mathbb{P}_u^{(2,2)}(\mathbf{X}_i)\right]^T \left[\mathbf{I} - \mathbb{P}_u^{(2,2)}(\mathbf{X}_i)\right] \tag{7.24}$$

  where (7.11) was applied.
- **Three-dimensional linear ansatz**:

$$\mathbf{M}_s^{(3,1)} = \varrho_0 \, \Omega_v \, \sum_{i=1}^{n_V} \left[\mathbf{I} - \mathbf{H}_u^{(3,1)}(\mathbf{X}_i) \, \mathbb{P}_u^{(3,1)}\right]^T \left[\mathbf{I} - \mathbf{H}_u^{(3,1)}(\mathbf{X}_i) \, \mathbb{P}_u^{(3,1)}\right] \tag{7.25}$$

  where (7.17) was employed.

Lumped Mass Matrix. In some cases it is more efficient to use a lumped mass matrix instead of the consistent mass matrix which can be found in (7.13), (7.15) and (7.19) for the consistency part of the virtual element formulation. Different lumping schemes are available, for an overview see Hughes (1987). Here we apply the row sampling technique that provides a diagonal matrix

$$\mathbf{M}_{diag} = \begin{bmatrix} m_1 & & & & \\ & m_2 & & & \\ & & m_3 & & \\ & & & \dots & \\ & & & & m_n \end{bmatrix} \quad \text{with} \;\; m_i = \sum_{k=1}^{n} M_{ik} \tag{7.26}$$

where M_{ik} are the components of the mass matrix $\mathbf{M}_\pi^{(d,n)}$ which depend on the chosen ansatz n. It is interesting to note, that, due to construction, the rank deficient consistent mass matrix stemming from the consistency part, see e.g. (7.11), produces a lumped mass matrix which is easily invertible.

7.3 Solution Algorithms for Small Strains

In the previous chapter we have discussed the application of virtual elements for elastic solids. Now this formulation will be enlarged to elasto-dynamics. After discretization using the virtual element formulations in Chap. 6 and Sect. 7.2 a system of ordinary differential equations appears in matrix form, contrary to the static case, where a linear system of equations has to be solved. In this section, first the matrix formulation of elasto-dynamics is provided and second solution algorithms will be discussed.

7.3.1 Matrix Formulation

In case of dynamics the system of Eqs. (7.1) which can be cast in the form of a pseudo potential (7.4). For small strains the strain energy $\Psi(\boldsymbol{u})$ is provided by (2.43). By inserting this form in the pseudo potential (7.4) we obtain

$$U(\boldsymbol{u}, \ddot{\boldsymbol{u}}) = \int_\Omega \left[\varrho_0 \, \ddot{\boldsymbol{u}} \cdot \boldsymbol{u} + \frac{1}{2} \hat{\boldsymbol{\varepsilon}}^T \mathbb{C} \, \hat{\boldsymbol{\varepsilon}} - \bar{\boldsymbol{f}} \cdot \boldsymbol{u} \right] \mathrm{d}\Omega - \int_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \boldsymbol{u} \, \mathrm{d}\Gamma \tag{7.27}$$

with the infinitesimal strains $\hat{\boldsymbol{\varepsilon}}$, see (2.9).

A matrix formulation for d-dimensional virtual elements with ansatz order n yields with Voigt notation, see Sect. 6.1.1 and (6.14),

$$\frac{1}{2} \mathbf{u}_v^T \, [\mathbb{P}_\nabla^{(d,n)}]_v^T \int_{\Omega_v} [\mathbf{B}_{u\,\pi}^{(d,n)}(X, Y)]^T \mathbb{C} \, \mathbf{B}_{u\,\pi}^{(d,n)}(X, Y) \, \mathrm{d}\Omega \; [\mathbb{P}_\nabla^{(d,n)}]_v \mathbf{u}_v = \frac{1}{2} \mathbf{u}_v^T \, \mathbf{K}_v^{(d,n)} \mathbf{u}_v$$

for the strain energy in (7.27) within an element Ω_v. By adding the mass contribution (consistency and stabilization term) and the loading terms, $\mathbf{f}_v^{(d,n)}$ for the body force and $\mathbf{P}_e^{(d,n)}$ for surface tractions, the matrix expression

$$\mathbf{u}_v^T \left[\mathbf{M}_\pi^{(d,n)} + \mathbf{M}_s^{(d,n)}\right] \ddot{\mathbf{u}}_v + \frac{1}{2}\mathbf{u}_v^T \mathbf{K}_v^{(d,n)}\, \mathbf{u}_v - \mathbf{u}_v^T \left(\mathbf{f}_v^{(d,n)} + \mathbf{P}_e^{(d,n)}\right) \tag{7.28}$$

can be written for an element Ω_v. The first variation of this pseudo potential with respect to the nodal displacements, with fixed accelerations, and assembly over all elements leads to, see Sect. 6.1.1 and (6.10),

$$\mathbf{M}\, \ddot{\mathbf{u}}(t) + \mathbf{K}\, \mathbf{u}(t) = \mathbf{F}(t)\,. \tag{7.29}$$

In this equation **M** is the mass matrix, see Sect. 7.2, and **K** the stiffness matrix of the virtual element discretization. The accelerations $\ddot{\mathbf{u}}$ and displacements **u** refer to the global unknowns of the discretized solid. Both matrices, **M** and **K**, are independent of time while the load vector **F**, containing body and traction forces, as well as the acceleration $\ddot{\mathbf{u}}$ and the displacement **u** depend on time. Thus the discretization using virtual elements leads to a set of second order linear ordinary differential equations in time which can be solved in different ways.[2]

7.3.2 *Numerical Integration in Time, Time Stepping Schemes*

Different options exist to solve the second order linear ordinary differential equation system in time, provided by (7.29). Especially in the linear case very efficient time stepping schemes can be developed. In the following we will neglect the damping part to simplify the presentation.

Possible solution schemes are presented next.

- **Modal analysis** is based on the computation of eigenvalues and -vectors of the dynamical system (7.29). The displacement can then be written as $\mathbf{u}(\boldsymbol{X},t) = \boldsymbol{\varphi}(\boldsymbol{X})\ \sin \omega t$. The eigenvalue system

$$(\mathbf{K} - \boldsymbol{\omega}^2\, \mathbf{M})\, \boldsymbol{\varphi} = \mathbf{0} \tag{7.30}$$

yields the eigenvalues ω_i and eigenvectors $\boldsymbol{\varphi}_i$. By defining a matrix containing all eigenvectors $\boldsymbol{\Psi} = \left[\boldsymbol{\varphi}_1 | \boldsymbol{\varphi}_2 | \dots | \boldsymbol{\varphi}_i | \dots | \boldsymbol{\varphi}_N\right]$ it is possible to transform the matrix equation (7.29) to a diagonal form. With the orthogonality conditions for the eigenvectors (note $\boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_k = 0$, for $i \neq k$) it yields

$$\boldsymbol{\Psi}^T \mathbf{K}\, \boldsymbol{\Psi} = \boldsymbol{\omega}^{\mathbf{2}}{}_d\,, \quad \boldsymbol{\Psi}^T \mathbf{M}\, \boldsymbol{\Psi} = \mathbf{I} \tag{7.31}$$

[2] For dynamical applications in engineering often damping due to material, frictional forces in connections and other structural components has to be considered. By adding the term $\mathbf{C}\,\dot{\mathbf{u}}$ to the left hand side of (7.29) damping can be introduced where **C** is the damping matrix. Often the damping matrix is formulated as so called modal or Rayleigh damping by a combination of mass and stiffness matrix: $\mathbf{C} = \alpha_1\, \mathbf{M} + \alpha_2\, \mathbf{K}$, for more details see e.g. Bathe (1996), Hughes (1987), Zienkiewicz and Taylor (2000a).

where $\boldsymbol{\omega^2}_d$ is a diagonal matrix that contains all eigenvalues ω_i, $(1 \le i \le N)$. Furthermore, the eigenvectors are scaled such that the transformation with the mass matrix yields the identity matrix $\mathbf{I}$.
By using the orthogonality of the eigenvectors, the displacement $\mathbf{u}$ can be written as

$$\mathbf{u}(\boldsymbol{X}, t) = \boldsymbol{\Psi}(\boldsymbol{X})\, \mathbf{q}(t) \tag{7.32}$$

where the components q_i in $\mathbf{q}(t)$ state the influence of each eigenvector $\boldsymbol{\varphi}_i$. This modal amplitude q_i is a function of time. Note that no approximation is introduced when all eigenvectors N of the system (7.30) are included. The transformation (7.32) can now be inserted into the differential system of Eqs. (7.29) and by premultiplication with $\boldsymbol{\Psi}$ a diagonal form follows

$$\ddot{\mathbf{q}}(t) + \boldsymbol{\omega}_d^2\, \mathbf{q}(t) = \hat{\mathbf{F}}(t) \quad \text{with} \quad \boldsymbol{\Psi}^T \mathbf{F}(t) = \hat{\mathbf{F}}(t) \tag{7.33}$$

where the orthogonality relations (7.31) were employed. Equivalently we can write

$$\ddot{q}_i(t) + \omega_i^2 q_i(t) = \hat{F}_i \quad \text{with} \quad \hat{F}_i = \boldsymbol{\varphi}_i^T\, \hat{\mathbf{F}}(t)\,. \tag{7.34}$$

There are as many ordinary differential equations (7.34) as degrees of freedom. The initial values of the system (7.1) for the displacements and the velocities ($\mathbf{u}(0) = \mathbf{u}_0$) and ($\dot{\mathbf{u}}(0) = \mathbf{v}_0$) can be transformed as well by using (7.32) and (7.31). It follows

$$\mathbf{q}(0) = \boldsymbol{\Psi}^T \mathbf{M}\, \mathbf{u}_0 \quad \text{and} \quad \dot{\mathbf{q}}(0) = \boldsymbol{\Psi}^T \mathbf{M}\, \mathbf{v}_0\,. \tag{7.35}$$

Solving all N ordinary equations is not efficient. However, it is possible to reduce the work considerably by only using as many eigenvalues and -vectors as needed to obtain a good approximation of the solution of (7.33). In this case, the solution $\mathbf{u}$ is approximated by using only a subset of the eigenvectors

$$\mathbf{u} = \bar{\boldsymbol{\Psi}}\, \mathbf{q} \tag{7.36}$$

where $\bar{\boldsymbol{\Psi}} = \left[\boldsymbol{\varphi}_1 | \boldsymbol{\varphi}_2 | \ldots | \boldsymbol{\varphi}_M\right]$ with $M < N$, see e.g. Cook et al. (1989), Bathe (1996). The needed eigenvalues and -vectors can be selected such that the subspace of the chosen eigenvectors reproduces a desired amount of the total energy. A measure for the error can be defined as, see e.g. Cook et al. (1989),

$$\epsilon(t) = \frac{\|\mathbf{F}(t) - \mathbf{M}\, \bar{\boldsymbol{\Psi}}\, \ddot{\mathbf{q}} - \mathbf{K}\, \bar{\boldsymbol{\Psi}}\, \mathbf{q}\|}{\|\mathbf{F}(t)\|} \tag{7.37}$$

where it is assumed that $\mathbf{F} \neq \mathbf{0}$ when $\epsilon(t)$ is computed for a specific time. The error has to be monitored during the analysis and to be kept within a specified tolerance, e.g. 1% .

- **Explicit time integration**. For a time stepping method that is explicit the dynamic equation (7.29) has to be evaluated at time t_n

$$\mathbf{M}\,\ddot{\mathbf{u}}_n + \mathbf{K}\,\mathbf{u}_n = \mathbf{F}_n \tag{7.38}$$

where the subscript n denoted the value of the displacement and acceleration at time step t_n. Many different time stepping schemes can be applied to solve (7.38), for an overview see e.g. Wood (1990). Here we use a very common method which is based on the central difference quotient for derivatives at time t_n. This difference scheme leads to the approximation

$$\ddot{\mathbf{u}}_n = \frac{1}{(\Delta t)^2}\,[\,\mathbf{u}_{n+1} - 2\,\mathbf{u}_n + \mathbf{u}_{n-1}\,] \tag{7.39}$$

for the acceleration which can be inserted into (7.38) leading to

$$\mathbf{M}\,\mathbf{u}_{n+1} = (\Delta t)^2\,\mathbf{F}_n + \left[2\,\mathbf{M} - (\Delta t)^2\,\mathbf{K}\right]\,\mathbf{u}_n - \mathbf{M}\,\mathbf{u}_{n-1}\,. \tag{7.40}$$

For the computation of $\mathbf{u}_{n+1}$ the mass matrix has to be inverted. While in a finite element analysis the mass matrix has full rank, this is not the case for virtual elements when only the consistency part of the mass matrix is used, see Sect. 7.2. Hence stabilization has to be introduced, see (7.21), or a lumped mass matrix needs to be employed, see (7.26). In any case, explicit time stepping is only efficient when a lumped mass matrix is employed. Then no equation system needs to be solved and the evaluation of (7.40) is trivial and fast. The accuracy of the central difference scheme is of second order.

In (7.40) the displacement $\mathbf{u}_{n-1}$ at time t_{n-1} is needed. This poses a problem at the start of the time stepping ($t_0 = 0$). By using a backward Taylor series expansion of second order at $t_0 = 0$ the displacement $\mathbf{u}_{0-1}$ can be computed from initial conditions for displacements and velocities $\mathbf{u}_0$ und $\dot{\mathbf{u}}_0 = \mathbf{v}_0$

$$\mathbf{u}_{0-1} = \mathbf{u}_0 - \Delta t\,\mathbf{v}_0 + \frac{1}{2}(\Delta t)^2\ddot{\mathbf{u}}_0 \quad \text{with} \;\; \ddot{\mathbf{u}}_0 = \mathbf{M}^{-1}\,[\,\mathbf{F}_0 - \mathbf{K}\,\mathbf{u}_0]\;. \tag{7.41}$$

The central difference method is unstable for large time steps. The stability limit of the method is given by the Courant number $\Delta t \le 2\,/\,\omega_{max}$ where ω_{max} is the largest eigenvalue. This value depends on the element with smallest size in the discretization. The critical time step can be estimated by $\Delta t \approx h\,/\,c_L$ where h is a characteristic (smallest) element length and c_L the velocity of a longitudinal wave, depending on the constitutive parameters. Since this limitation yields often very small time steps, the method is best suited for problems with short term loading like e.g. shocks, impact or wave propagation.

- **Implicit time integration**. Contrary to the explicit scheme, implicit integration methods are based on the discretized dynamic equation at time t_{n+1}

$$\mathbf{M}\,\ddot{\mathbf{u}}_{n+1} + \mathbf{K}\,\mathbf{u}_{n+1} = \mathbf{F}_{n+1} \tag{7.42}$$

Several time stepping schemes exist for the implicit integration of the equations of motion, see e.g. Wood (1990), Bathe (1996). An often used variant is the Newmark scheme which leads to the following difference equations for displacements and velocities at time t_{n+1}

$$\begin{aligned} \mathbf{u}_{n+1} &= \mathbf{u}_n + \Delta t\,\dot{\mathbf{u}}_n + \frac{(\Delta t)^2}{2}\,[\,(1-2\beta)\,\ddot{\mathbf{u}}_n + 2\beta\,\ddot{\mathbf{u}}_{n+1}\,] \\ \dot{\mathbf{u}}_{n+1} &= \dot{\mathbf{u}}_n + \Delta t\,[\,(1-\gamma)\,\ddot{\mathbf{u}}_n + \gamma\,\ddot{\mathbf{u}}_{n+1}\,] \end{aligned} \tag{7.43}$$

with two parameters γ and β. By inserting these approximations into the discretized dynamic equation (7.42) we obtain an equation system for the accelerations

$$\hat{\mathbf{M}}\,\ddot{\mathbf{u}}_{n+1} = \mathbf{F}_{n+1} - \mathbf{K}\,\hat{\mathbf{u}}_n \tag{7.44}$$

with

$$\begin{aligned} \hat{\mathbf{M}} &= \mathbf{M} + \beta\,(\Delta t)^2\,\mathbf{K} \qquad \text{and} \\ \hat{\mathbf{u}}_n &= \mathbf{u}_n + \Delta t\,\dot{\mathbf{u}}_n + (\frac{1}{2} - \beta)(\Delta t)^2\,\ddot{\mathbf{u}}_n\,. \end{aligned}$$

The method is implicit since the equation system (7.44) has to be solved to obtain the accelerations.[3] Once the accelerations are known Eqs. (7.43) yield displacements and velocities at time t_{n+1}.

The selection of the parameters γ und β determines the accuracy and stability of the method. The Newmark method is unconditionally stable for $\frac{1}{2} \le \gamma \le 2\,\beta$, meaning an arbitrary large time step Δt can be employed. The method is thus preferable for responses with periodic loading, like vibration problems of machines and buildings in structural dynamics. For $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$ the method is equivalent to the trapezoidal rule and hence of second order accuracy. For $\beta = 0$ and $\gamma = \frac{1}{2}$ one recovers the explicit central difference scheme.

7.4 Solution Algorithms for Finite Strains

For large deformations the system of Eqs. (7.1) can be formulated in terms of a pseudo potential (7.4) with the strain energy $\Psi(\nabla \boldsymbol{u})$. By inserting this form in the pseudo potential (7.4) we obtain

[3] It is straightforward to reformulate the above equation system in such a way that an equivalent equation to (7.44) yields the displacements at time t_{n+1}. Such formulation of Newmark's method can be found in many papers, but of course, both formulations are equivalent.

$$U(\boldsymbol{u}, \ddot{\boldsymbol{u}}) = \int_{\Omega} \left[\varrho_0 \, \ddot{\boldsymbol{u}} \cdot \boldsymbol{u} + \Psi(\nabla \boldsymbol{u}) - \bar{\boldsymbol{f}} \cdot \boldsymbol{u} \, \right] \, \mathrm{d}\Omega - \int_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \boldsymbol{u} \, \mathrm{d}\Gamma \, . \tag{7.45}$$

Except for the term of the nonlinear strain energy all other integrals regarding loading and inertia terms are the same as in linear case, see (7.28). It remains to formulate the term associated with the strain energy. This has to be performed for the consistency and the stabilization part. For the latter we choose the energy stabilization discussed in Sect. 6.2.2.

The discretization of the consistency term can be based on a generalization of (3.94) for d dimensions and polynomial ansatz order of n leading to the discretization of the displacement gradient for the consistency part, see e.g. Sect. 6.2.1,

$$\nabla \mathbf{u}_\pi \, = \mathbf{B}_{u\,\pi}^{(d,n)}(\mathbf{X}) \, \mathbb{P}_{\nabla}^{(d,n)} \, \mathbf{u}_v \, . \tag{7.46}$$

This approximation can be applied in $\int_{\Omega_v} \Psi(\nabla \boldsymbol{u}) \mathrm{d}\Omega$. Since $\Psi(\nabla \boldsymbol{u})$ is a nonlinear function with respect to $\nabla \boldsymbol{u}$ the integration cannot be computed via boundary integrals. Thus the area of the virtual element Ω_v has to be subdivided into n_{int} triangles Ω_m^i with $\cup_{m=1}^{n_{int}} = \Omega_v$ in the two-dimensional case, as discussed in Sect. 6.2.1, or in n_{int} tetrahedra in the three-dimensional case. Now the integral including the strain energy can reformulated as

$$\int_{\Omega_v} \Psi(\nabla \boldsymbol{u}_\pi) \mathrm{d}\Omega = \sum_{m=1}^{n_{int}} \int_{\Omega_m^i} \Psi(\nabla \boldsymbol{u}_\pi) \mathrm{d}\Omega \, . \tag{7.47}$$

The integration is performed by applying a Gauss point scheme with weighting points w_g and integration points $\boldsymbol{\xi}_g$ for each Ω_m^i. For an efficient implementation the numerical integration is based on a coordinate transformation using an isoparametric map $\mathbf{X} = \sum_{i=1}^{n_m} N_I(\boldsymbol{\xi}) \mathbf{X}_I$ where $\mathbf{X}_I$ are the vertices of the virtual element. This yields for the general case

$$\begin{aligned} \int_{\Omega_m^i} \Psi[\nabla \mathbf{u}_\pi(\mathbf{X})] \mathrm{d}\Omega &= \int_{\boldsymbol{\xi}} \Psi[\mathbf{B}_{u\,\pi}^{(d,n)}(\boldsymbol{\xi}) \mathbb{P}_{\nabla}^{(d,n)} \, \mathbf{u}_v] \det \mathbf{J}_v(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi} \\ &= \sum_{g=1}^{n_g} w_g \, \Psi[\mathbf{B}_{u\,\pi}^{(d,n)}(\boldsymbol{\xi}_g) \mathbb{P}_{\nabla}^{(d,n)} \, \mathbf{u}_v] \det \mathbf{J}_v(\boldsymbol{\xi}_g) \, . \end{aligned} \tag{7.48}$$

where $\boldsymbol{\xi} = \{\xi \, , \eta\}$ for the two-dimensional and $\boldsymbol{\xi} = \{\xi \, , \eta \, , \zeta\}$ for the three-dimensional case. In short we can write for the sum over all subtriangles or tetrahedra related to the potential energy of the consistency term

$$U_c^{(d,n)}(\mathbf{u}_v) = \sum_{m=1}^{n_{int}} \left[\sum_{g=1}^{n_g} w_g \, \Psi[\mathbf{B}_{u\,\pi}^{(d,n)}(\boldsymbol{\xi}_g) \mathbb{P}_{\nabla}^{(d,n)} \, \mathbf{u}_v] \det \mathbf{J}_v(\boldsymbol{\xi}_g) \right]_m \, . \tag{7.49}$$

Note that for the linear ansatz this expression is a lot simpler

$$U_c^{(d,1)}(\mathbf{u}_v) = \int\limits_{\Omega_v} \Psi(\nabla \boldsymbol{u}_\pi) \mathrm{d}\Omega = \Psi(\mathbb{P}_\nabla^{(d,1)}\, \mathbf{u}_v)\, \Omega_v \tag{7.50}$$

which still is a nonlinear relation with respect to the nodal displacements $\mathbf{u}_v$.

Now the matrix form of the pseudo-potential follows from (7.45) with respect to an element Ω_v as in (7.28)

$$\mathbf{u}_v^T \left[\mathbf{M}_\pi^{(d,n)} + \mathbf{M}_s^{(d,n)}\right] \ddot{\mathbf{u}}_v + \left[U_c^{(d,n)}(\mathbf{u}_v) + U_s^{(d,n)}(\mathbf{u}_v)\right] - \mathbf{u}_v^T\, (\mathbf{f}_v^{d,n} + \mathbf{P}_e^{d,n}). \tag{7.51}$$

where $U_s^{(d,n)}(\mathbf{u}_v)$ is the energy related to the stabilization. The residual for the consistency term of one virtual element follows from

$$\mathbf{R}_{c\,v}^{(d,n)}(\mathbf{u}_v) = \frac{\partial U_c^{(d,n)}(\mathbf{u}_v)}{\partial \mathbf{u}_v} \tag{7.52}$$

The residual, using the energy formulation, can be obtained for the two-dimensional case by (6.54) and for the three-dimensional case by (6.64) leading to $\mathbf{R}_v^s(\mathbf{u}_v)$ for an element Ω_v.

Differentiation of (7.51) with respect to the nodal displacements $\mathbf{u}_v$, while keeping $\ddot{\mathbf{u}}_v$ fixed, yields

$$\left[\mathbf{M}_\pi^{(d,n)} + \mathbf{M}_s^{(d,n)}\right] \ddot{\mathbf{u}}_v + \left[\mathbf{R}_{c\,v}^{(d,n)}(\mathbf{u}_v) + \mathbf{R}_v^s(\mathbf{u}_v)\right] - (\mathbf{f}_v^{d,n} + \mathbf{P}_e^{d,n})\,. \tag{7.53}$$

Assembly over all elements leads to, see Sect. 6.1.1 and (6.10),

$$\mathbf{M}\, \ddot{\mathbf{u}}(t) + \mathbf{R}[\mathbf{u}(t)] = \mathbf{F}(t)\,. \tag{7.54}$$

For the solution of the time dependent nonlinear problem the implicit Newmark scheme is applied. By rewriting $(7.43)_1$ as

$$\ddot{\mathbf{u}}_{n+1} = \frac{1}{\beta(\Delta t)^2}(\mathbf{u}_{n+1} - \mathbf{u}_n) - \frac{1}{\beta(\Delta t)}\dot{\mathbf{u}}_n - \frac{1-2\beta}{2\beta}\, \ddot{\mathbf{u}}_n \tag{7.55}$$

and inserting into the (7.54) at time t_{n+1} a nonlinear equation system follows

$$\mathbf{G}(\mathbf{u}_{n+1}) = \frac{1}{\beta(\Delta t)^2}\mathbf{M}\, \mathbf{u}_{n+1} + \mathbf{R}(\mathbf{u}_{n+1}) - \mathbf{F}_{n+1} - \mathbf{M}\, \widehat{\ddot{\mathbf{u}}}_n = \mathbf{0} \tag{7.56}$$

with

$$\widehat{\ddot{\mathbf{u}}}_n = \frac{1}{\beta(\Delta t)^2}\mathbf{u}_n + \frac{1}{\beta(\Delta t)}\dot{\mathbf{u}}_n + \frac{1-2\beta}{2\beta}\, \ddot{\mathbf{u}}_n\,. \tag{7.57}$$

Algorithm 1: Newmark algorithm for finite strain dynamics

```
Given: $\mathbf{u}_0 = \mathbf{u}(t_0)\,,\, \dot{\mathbf{u}}_0 = \dot{\mathbf{u}}(t_0)\,,\, \ddot{\mathbf{u}}_0 = \ddot{\mathbf{u}}(t_0) = \mathbf{0}\,,\, \mathbf{F}_{n+1}$    Find: $\mathbf{u}_{n+1}\,,\, \dot{\mathbf{u}}_{n+1}\,,\, \ddot{\mathbf{u}}_{n+1}$ ;
for $n = 0$ to $n_{steps}$ do
    $\mathbf{u}^0_{n+1} = \mathbf{u}_n$ ;
    for $i = 0$ to $n_{iter}$ do
        $\mathbf{K}^{eff}_T(\mathbf{u}^i_{n+1})\Delta\mathbf{u}^{i+1}_{n+1} = -\mathbf{G}(\mathbf{u}^i_{n+1})$;
        $\mathbf{u}^{i+1}_{n+1} = \mathbf{u}^i_{n+1} + \Delta\mathbf{u}^{i+1}_{n+1}$;
        $\|\mathbf{G}(\mathbf{u}^{i+1}_{n+1})\| \leq \varepsilon \longrightarrow$ STOP
    end
    $\mathbf{u}_{n+1} = \mathbf{u}^{i+1}_{n+1}$;
    $\ddot{\mathbf{u}}_{n+1} = \frac{1}{\beta(\Delta t)^2}(\mathbf{u}_{n+1} - \mathbf{u}_n) - \frac{1}{\beta(\Delta t)}\dot{\mathbf{u}}_n - \frac{1-2\beta}{2\beta}\,\ddot{\mathbf{u}}_n$;
    $\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \Delta t\,[\,(1-\gamma)\,\ddot{\mathbf{u}}_n + \gamma\,\ddot{\mathbf{u}}_{n+1}]$;
end
```

The equation system (7.56) is usually solved within a time step by the Newton-Raphson scheme. With

$$\mathbf{K}^{eff}_T(\mathbf{u}_{n+1}) = \frac{\partial \mathbf{G}(\mathbf{u}_{n+1})}{\partial \mathbf{u}_{n+1}} = \frac{1}{\beta(\Delta t)^2}\mathbf{M} + \mathbf{K}_T(\mathbf{u}_{n+1}) \tag{7.58}$$

the Algorithm 1 can be formulated, where $\mathbf{K}_T(\mathbf{u}_{n+1})$ is the tangent stiffness matrix resulting from the residual $\mathbf{R}(\mathbf{u}_{n+1})$.

The algorithm provides displacements, velocities and accelerations at the new time $t_{n+1} = t_n + \Delta t$ where Δt has to be selected by the user with respect to the characteristics of the physical problem. Within the algorithm ε is a tolerance that is usually set to $\varepsilon = 10^{-10}$, n_{iter} is the maximum number of steps in the algorithm which is set to $n_{iter} = 20$. Since the Newton algorithm has quadratic convergence properties, 4 to 6 iterations are usually sufficient to find the solution within a time step. If the problem is highly nonlinear it can happen that no convergence is achieved within the n_{iter} steps. In such cases the time step has to be reduced and the Newton iteration has to start again with the initial values at t_n. Classically the reduction is given by $\Delta t_{new} = \Delta t_{old}\,/\,2$. Once the displacements, velocities and accelerations are determined they will be applied as initial data for the next time step and so forth until the maximum number of time steps n_{steps} is reached.

7.5 Numerical Examples

In this section the performance of different virtual element formulations will be investigated. For comparison purposes results of the standard finite element method (FEM) are also included. The material parameters used in this book are the same for all examples and are provided in Table 7.1, unless it is otherwise specified.

Table 7.1 Material parameters used for the numerical examples

No.	Parameter	Label	Value	Unit
1	Elastic modulus	E	210	kN/mm^2
2	Poisson ratio	ν	{0.3, 0.499999}	–
3	Density	ρ	0.0027	g/mm^3

The virtual element discretization is based on a first order pure displacement formulation, see (6.41) with the stabilization (6.61) using the standard mesh types VEM-H1 and VEM-VO types, see Appendix B. Additionally, the formulations and mesh types are employed:

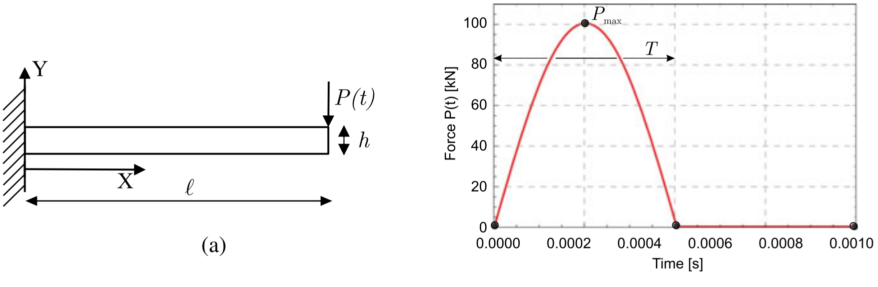
- VEM H1-JP0: A regular shaped 3D virtual element with 8 nodes. This element is using a Hu-Washizu formulation with a linear ansatz for the displacement, constant pressure p and constant dilatation θ as additional degrees of freedom, see (6.103) with the stabilization (6.61).
- VEM VO-JP0: A 3D voronoi shaped virtual element with arbitrary number of nodes. This element is using a Hu-Washizu formulation with a linear ansatz for the displacement, constant pressure p and constant dilatation θ as additional degrees of freedom (6.103) with the stabilization (6.61).

For a representative comparison, the standard finite element formulations FE-H1 with linear ansatz and FE-H2 with quadratic ansatz and 27 nodes are selected. Additionally a regular shaped 3D finite element FE-H1-JP0 with 8 nodes is used. This element is based on a Hu-Washizu formulation with a linear ansatz for the displacement, constant pressure p and constant dilatation θ as additional degrees of freedom, see Simo et al. (1985a).

The stabilization parameter of the static part β^{stat} is chosen in all the simulations using (6.63), unless it is otherwise specified. For the dynamic part the mass-matrix is computed according to (7.20) without any stabilization.

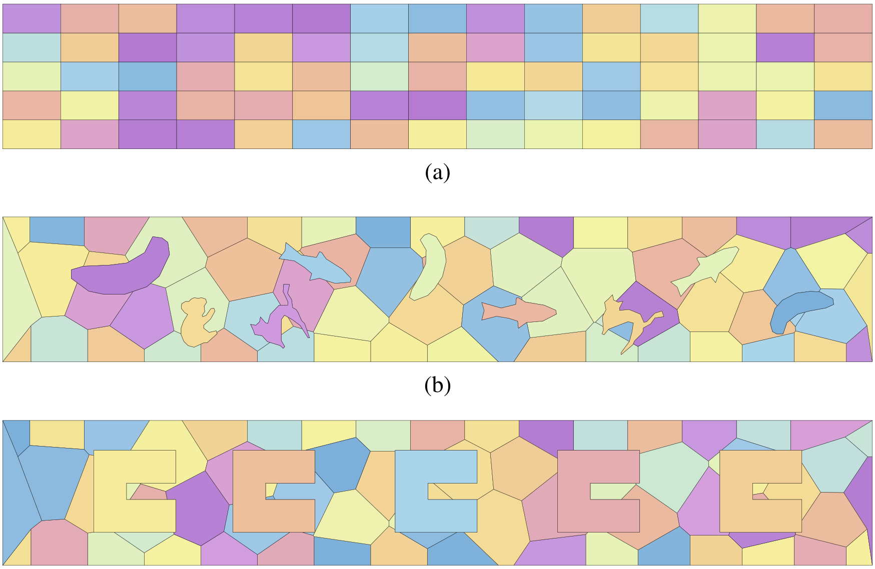
7.5.1 Transversal Beam Vibration

Transversal vibrations in beams are a classical application of dynamics and will be used to compare virtual element to finite element solutions. A cantilever beam with an end load is considered. Its geometry and the time dependent load are depicted in Fig. 7.1a. The material parameters are the Young's modulus $E = 210\,\text{kN/mm}^2$, Poisson ratio $\nu = 0.3$ and density $\varrho = 0.0027\,\text{g/mm}^3$. The beam length is $\ell = 30$ mm, the height is $h = 5$ mm and the depth $b = 1$ mm. A point load $P(t)$ acts in transversal direction at the upper corner at $\text{X} = \ell$ as shown in Fig. 7.1a. The load depends on the time, see Fig. 7.1b, with a peak of the load $P_{\max} = 100$ kN. The period T of the applied load is adjusted to the bending stiffness of the beam and defined by the analytical solution related to the first eigenfrequency of a clamped beam



(b)

Fig. 7.1 2D Example—Transversal beam vibration. Boundary value problem in (**a**) and applied force in (**b**)


(a)

(b)

(c)

Fig. 7.2 2D Example—Transversal Beam Vibration. **a** VE-Q2S Mesh, **b** VEM Animal-Mesh and **c** C-Mesh

$$T = \frac{3.5156}{2\pi\ell^2}\sqrt{\frac{12\rho}{Ebh^3}} \tag{7.59}$$

In order to analyze the position effect of the element centroid on evaluating the integral of the dynamic part, we used different types of meshes as illustrated in Fig. 7.2. The “animal” mesh (Fig. 7.2b) includes concave elements. To investigate the effect of using concave elements where the centroid of the element is outside of

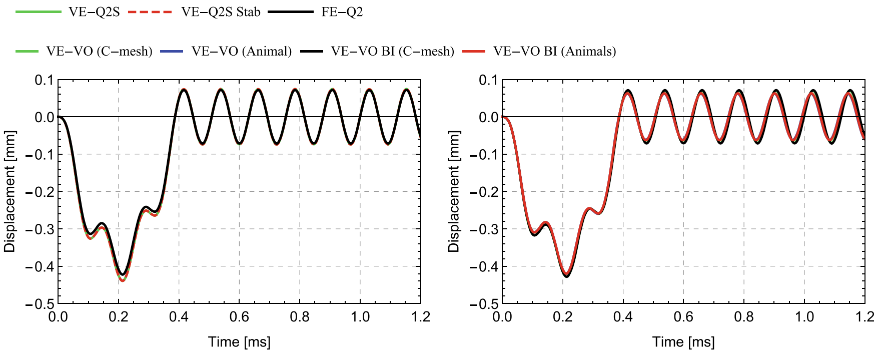


Fig. 7.3 Displacement over time response at X $= \ell$ for 2D Example—Transversal Beam Vibration

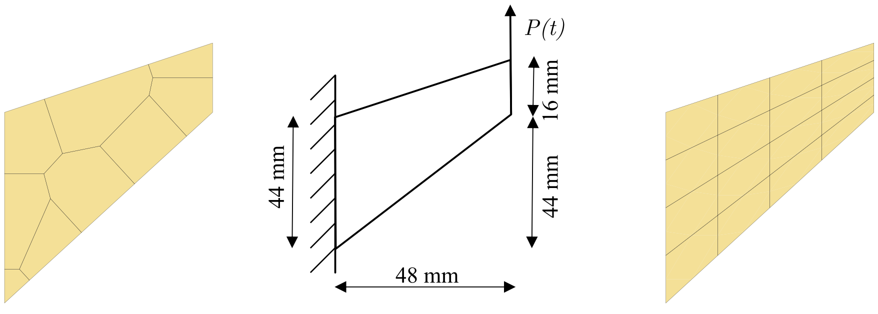
the element domain, we introduce a special mesh with elements like C's, where the centroid of the element is outside of the element domain (Fig. 7.2c).

Figure 7.3 depicts the displacement over time response at the end of the beam (X $= \ell$). The finite element solution is computed for 1000 FE-Q2 elements, whereas VEM results are obtained with the meshes depicted in Fig. 7.2. A comparison of response of the virtual elements VO and VO-BI illustrates that different evaluations of the integral (mass matrix) in Eq. (6.38) yield the same displacement results either using the centroid (VO) of the element or an exact evaluation (VO-BI). Furthermore, we observe that the displacements in the center of the beam are slightly larger than the finite element results. The period fits very well compared with FEM results. In general the virtual element results are in a good agreement even, as in this case, when compared with finite element FE-Q2 results which approximate the displacement field with a quadratic ansatz. The linear virtual elements VE-Q2S and VE-Q2S-Stab reproduce nearly the same response. Thus a very good approximation of the dynamical behaviour of the beam is provided by the virtual element method, even without stabilizing the mass matrix.

The influence of the different meshes shows, that the C-mesh yields a larger deflection when compared with the results of the other mesh types. Nevertheless qualitatively the shape of the displacement over time response fits very well the finite element FE-Q2 results and the virtual element VE-Q2S results. Hence we can conclude that the evaluation of the integral for the mass matrix at the centroid of the element does not affect the results.

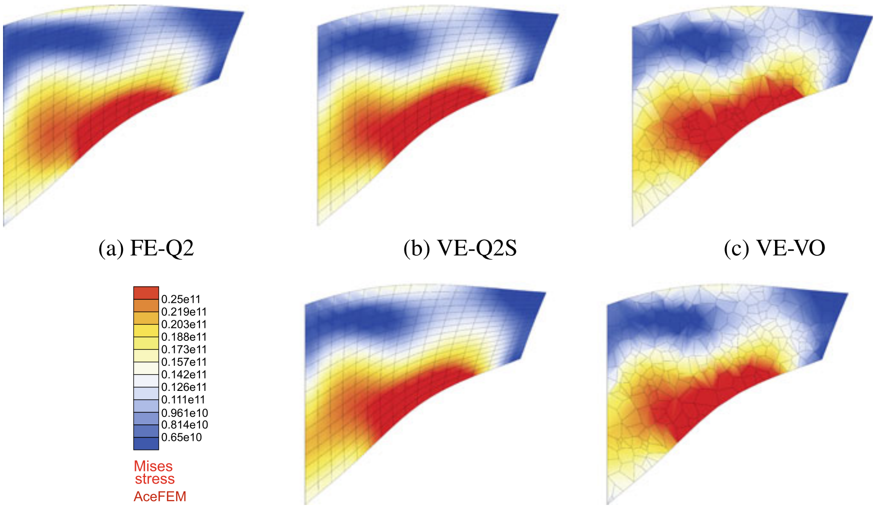
7.5.2 Cook's Membrane Problem

In the two-dimensional Cook's membrane problem the virtual element performance will be compared with finite element results. The geometrical setup and boundary conditions are provided in Fig. 7.4b. In this test a force of $P_{max} = 10000$ kN/mm is applied at the right edge as a line load, as depicted in Fig. 7.4b. The VE-VO mesh



(a) (b) (c)

Fig. 7.4 2D Example—Cook's membrane. **a** VE-VO Mesh, **b** boundary value problem and **c** VE-Q2S Mesh



(a) FE-Q2 (b) VE-Q2S (c) VE-VO

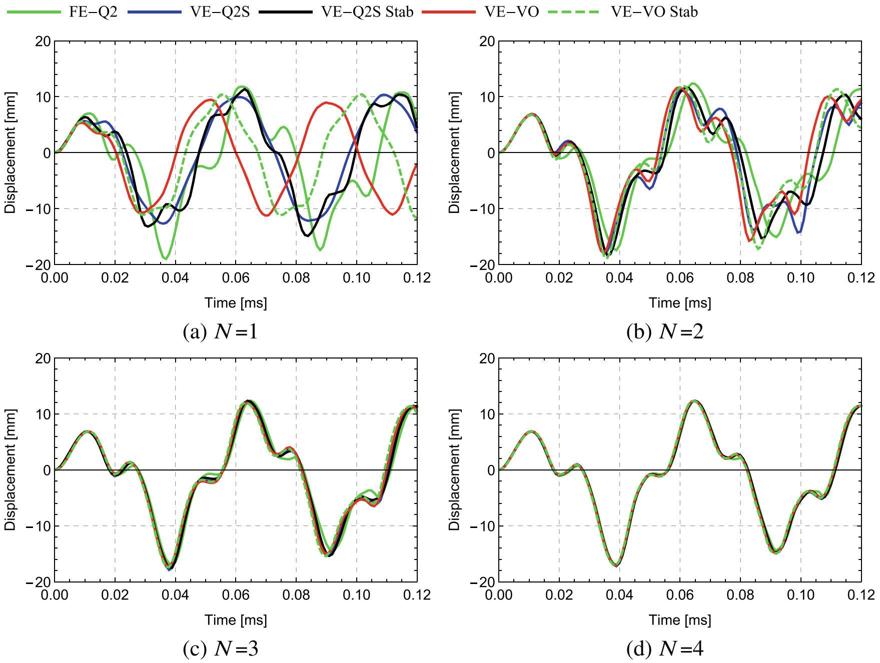
(d) VE-Q2S Stab (e) VE-VO Stab

Fig. 7.5 2D Example—Cook's membrane. Von Mises stress distribution at time $t = 0.035$ ms for different elements at same scale

and regular VE-Q2S mesh are also depicted in Fig. 7.4a, c, respectively. The material data are provided in Table 7.1.

The contour plots of the von Mises stress distribution for different elements at the time $t = 0.035$ ms are sketched in Fig. 7.5. Both elements VE-Q2S and VE-Q2S Stab, which use the non stabilized and the stabilized mass matrix, produce nearly the same von Mises stress distribution. The nonlinear behaviour is clearly observed by large deformations interacting with dynamic effects at finite strains.

Figure 7.6 shows a mesh refinement study for different mesh types with the element division of 2^N with $N \in \{1, 2, 3, 4\}$. For $N = 3$ and higher the solutions of



(a) N=1

(b) N=2

(c) N=3

(d) N=4

Fig. 7.6 Convergence Study—Displacement over time response for 2D Example—Cook's membrane. Element division 2^N, where N increases from (**a**) to (**d**)

all mesh types converge. Furthermore, a comparison with the finite element method shows a very good agreement.

This study underlines again, that the evaluation of the mass matrix at the element centroid, using (7.20), is sufficient for a successful analysis of a dynamical problem with VEM. Interestingly, different evaluations of the integral related to the mass matrix can yield a rank deficient mass matrix or a mass matrix with full rank, but both approaches reproduce almost identical responses.

7.5.3 3D Beam

A clamped three-dimensional beam is dynamically loaded by a surface load $p(t)$ at the free end, as illustrated in Fig. 7.7. The load is applied as a half sine function, see Fig. 7.1b, with the time period $T_0 = 0.0008$ and an amplitude of 45 N/mm^2. Thereafter, the force is released and the beam is oscillating freely. The time increment for the numerical time stepping procedure is set to $\Delta t = 1\mu$s.

The essential aspect of this investigation is to demonstrate the performance of the Hu-Washizu formulation for compressible and nearly incompressible material

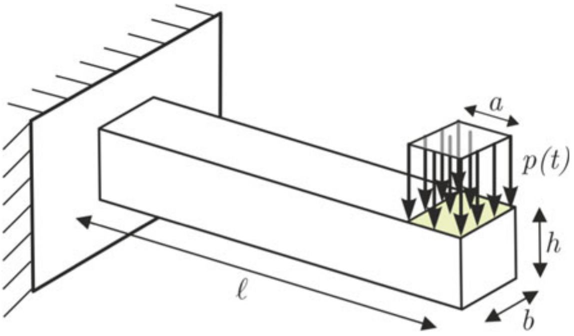


Fig. 7.7 3D Beam—Geometry and boundary conditions

behaviour in dynamics. Different Poisson's ratio are chosen:

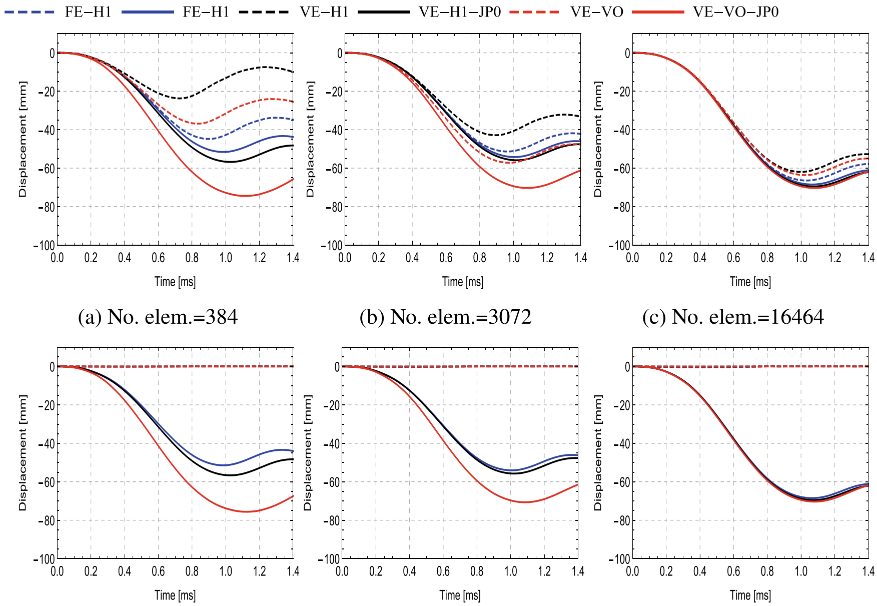
$$\nu \in \{0.3,\ 0.45,\ 0.49,\ 0.499,\ 0.4999,\ 0.49999,\ 0.499999\}\,.$$

The material parameters used in the simulations are listed in Table 7.1.

Figure 7.8 shows the time history of the displacement at the tip of the beam. The response for a compressible material with $\nu = 0.3$ is outlined in Fig. 7.8a—c. The maximal vertical displacement of the beam converges with increasing node number to nearly $u = 70$ mm, see Fig. 7.8c.

Nevertheless, the finite and virtual elements, which are based on a pure displacement formulation FE-H1 and VE-H1/VO, tend to provide stiffer responses. Such an observation is in line with the artificial stiffening due to shear locking which can occur in bending dominated situations, like in this example. The response for in increased value of Poisson's ratio, which is set nearly up to the incompressible limit (i.e. $\nu = 0.499999$), is outlined in Fig. 7.8d–f. A strongly stiffer response is observed for the pure displacement elements FE-H1 and VE-H1/VO when compared with the stable and robust mixed finite and virtual element formulations FE-H1-JP0 and VE-H1/VO-JP0. Thus the Hu-Washizu based finite and virtual elements produce a much softer response and can handle incompressible material and bending behaviour well.

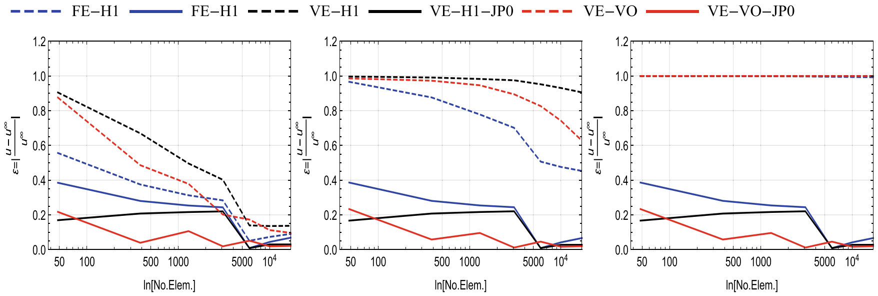
For a representative comparison between all elements, the relative error of the maximum displacement, shown in Fig. 7.8, is plotted in Fig. 7.9 for different element types and Poisson ratios. The error is computed with respect to an overkill solution, that is obtained from the mixed finite element FE-H1-JP0 using 100000 elements. The results in Fig. 7.9a depict for the compressible case ($\nu = 0.3$) that the error is remarkably reduced by increasing the number of elements for all types. The pure displacement elements FE-H1 and VE-H1/VO demonstrate in this example a high error in comparison with mixed FEM and VEM formulations, especially in the case of coarse meshes. When increasing the Poisson's ratio, the error of the pure displacement elements is further increased, reaching its maximum for $\nu = 0.499999$. The mixed finite and virtual elements stay nearly constant and are not effected by any kind of



(a) No. elem.=384 (b) No. elem.=3072 (c) No. elem.=16464

(d) No. elem.=384 (e) No. elem.=3072 (f) No. elem.=16464

Fig. 7.8 3D Beam—Displacement over time response for different element types and mesh discretization in (**a**) to (**c**) with ν=0.3 and (**d**) to (**f**) with ν=0.499999



(a) ν=0.3 (b) ν=0.499 (c) ν=0.499999

Fig. 7.9 3D Beam—error of the maximum displacement over time for different element types and various Poisson's ratio in (**a**) to (**c**)

locking phenomena. This illustrates the importance of using a mixed formulation for virtual element modeling incompressible materials.

References

Adak, D., D. Mora, and S. Natarajan. 2022. Convergence analysis of virtual element method for nonlinear nonlocal dynamic plate equation. *Journal of Scientific Computing* 91: 23.

Ahmad, B., A. Alsaedi, F. Brezzi, L. Marini, and A. Russo. 2013. Equivalent projectors for virtual element methods. *Computers and Mathematics with Applications* 66: 376–391.

Bathe, K.J. 1996. *Finite element procedures*. Englewood Cliffs, NJ: Prentice-Hall.

Bathe, K.J. 2006. *Finite element procedures*. Prentice Hall.

Beirão da Veiga, B.L., F. Brezzi, L.D. Marini, and A. Russo. 2014. The hitchhiker's guide to the virtual element method. *Mathematical Models and Methods in Applied Sciences* 24 (8): 1541–1573.

Boffi, D., F. Gardini, and L. Gastaldi. 2022. The virtual element approximation of eigenvalue problems. In *The virtual element method and its applications, volume 31 of SEMA SIMAI*, ed. P.L. Antonietti, L.B. da Veiga and G. Manzini. Springer Nature.

Cihan, M., F. Aldakheel, B. Hudobivnik, and P. Wriggers. 2021. Virtual element formulation for finite strain elastodynamics. *Computer Modeling in Engineering & Sciences* 129: 1151–1180.

Cihan, M., B. Hudobivnik, F. Aldakheel, and P. Wriggers. 2021. 3d mixed virtual element formulation for dynamic elasto-plastic analysis. *Computational Mechanics* 68: 1–18.

Cook, R.D., D.S. Malkus and M.E. Plesha. 1989. *Concepts and applications of the finite element analysis*. Hoboken, 3rd ed. Wiley.

Gay, N.A., B. Hudobivnik, T.F. Moherdaui, and P. Wriggers. 2021. Flexible polyhedra modeled by the virtual element method in a discrete element context. *Computer Methods in Applied Mechanics and Engineering* 387: 114163.

Hughes, T.J.R. 1987. *The finite element method*. Englewood Cliffs, NJ: Prentice Hall.

Korelc, J., and P. Wriggers. 2016. *Automation of finite element methods*. Berlin: Springer.

Mazzia, A., M. Ferronato, P. Teatini, and C. Zoccarato. 2020. Virtual element method for the numerical simulation of long-term dynamics of transitional environments. *Journal of Computational Physics* 407: 109235.

Park, K., H. Chi, and G. Paulino. 2019. On nonconvex meshes for elastodynamics using virtual element methods with explicit time integration. *Computer Methods in Applied Mechanics and Engineering* 356: 669–684.

Simo, J.C., R.L. Taylor, and K.S. Pister. 1985. Variational and projection methods for the volume constraint in finite deformation elasto-plasticity. *Computer Methods in Applied Mechanics and Engineering* 51: 177–208.

Wood, W.L. 1990. *Practical time-stepping schemes*. Oxford: Clarendon Press.

Wriggers, P. 2008. *Nonlinear finite elements*. Berlin, Heidelberg, New York: Springer.

Zienkiewicz, O.C., and R.L. Taylor. 2000. *The finite element method,* vol. 1, 5th ed. Oxford, UK: Butterworth-Heinemann.

Chapter 8
Virtual Element Formulation for Finite Plasticity

The application of virtual elements covers a wide range in solid mechanics. Besides pure elastic behaviour, material models can be introduced that include path dependency and thus history variables. The treatment of such constitutive models needs a special approach with respect to algorithms and the formulation of the virtual element method. The details will be discussed in the following for the case of three-dimensional von Mises plasticity at finite strains. This chapter follows the approach provided in Wriggers and Hudobivnik (2017) and Hudobivnik et al. (2018). Application of virtual element discretizations for rolling processes involving plastic behaviour can be found in Böhm et al. (2020). Extensions to applications involving anisotropic crystal-plasticity are discussed in Böhm et al. (2023).

8.1 Formulation of the Virtual Element

The formulation of a finite deformation virtual element for elasto-plasticity is based on the theoretical background provided in Sects. 2.2.3 and 2.3.3. For finite strains the deformation gradient $\boldsymbol{F}$ is split in an elastic, $\boldsymbol{F}_e$, and a plastic part, $\boldsymbol{F}_p$, in a multiplicative way: $\boldsymbol{F} = \boldsymbol{F}_e\,\boldsymbol{F}_p$, see (2.72). The kinematical relation (2.74),

$$\boldsymbol{b}_e = \boldsymbol{F}\,\boldsymbol{C}_p^{-1}\,\boldsymbol{F}^T \quad \text{with} \;\; \boldsymbol{C}_p = \boldsymbol{F}_p^T\,\boldsymbol{F}_p\,,$$

the flow rule (2.76)

$$f(\boldsymbol{s}, \alpha) = \sigma_{VM} - [\,Y_0 + (Y_\infty - Y_0)e^{-\delta\,\alpha} + H\,\alpha\,] \leq 0$$

and the evolution of the plastic strain (2.80)

P. Wriggers et al., *Virtual Element Methods in Engineering Sciences*,
https://doi.org/10.1007/978-3-031-39255-9_8

$$\dot{\boldsymbol{C}}_p^{-1} = -2\,\dot{\gamma}\,\boldsymbol{F}^{-1}\,\boldsymbol{n}\,\boldsymbol{F}\,\boldsymbol{C}_p^{-1} \quad \text{with } \boldsymbol{n} = \frac{\partial f}{\partial \boldsymbol{s}}$$

describe the plastic flow where $\boldsymbol{s}$ is the deviatoric Kirchhoff stress, see (2.77). These equations are employed together with the elastic strain energy Ψ, see e.g. (2.56) to locally characterize the constitutive behaviour.

Additionally, the pseudo potential (2.91), describing the global deformation and stress state,

$$U(\boldsymbol{u},\mathbf{h}) = \int\limits_{\Omega} \left[\Psi(\boldsymbol{u},\mathbf{h}) - \bar{\boldsymbol{f}}\cdot\boldsymbol{u}\,\right]\,\mathrm{d}\Omega - \int\limits_{\Gamma_N} \bar{\boldsymbol{t}}\cdot\boldsymbol{u}\,\mathrm{d}\Gamma$$

is used as starting point for the discretizations with virtual elements. Alternatively, the weak form (2.83) can be employed.

The virtual element is based on a split into the consistency part, which for a linear ansatz results in a constant part of the deformation gradient, and an associated stabilization term, see Sect. 5. As in the hyperelastic case we split the potential in a consistency part U_c and a stabilization part U_{stab}. By summing up all element contributions for n_v virtual elements we obtain

$$U(\boldsymbol{u}\,,\mathbf{h}) = \mathop{\mathsf{A}}_{v=1}^{n_v} \left[U_c(\boldsymbol{u}_\pi,\mathbf{h}_v) + U_{stab}(\boldsymbol{u}_h - \boldsymbol{u}_\pi\,,\mathbf{h}_v)\,\right], \tag{8.1}$$

where v denotes a single virtual element within the discretization. The associated element area/volume is then Ω_v. In the presented formulation the plastic variables $\mathbf{h}_v$ are computed from the consistency term of the virtual element Ω_v for finite strain elasto-plasticity. These plastic variables are kept constant when included as history variables in the stabilization procedure.

8.1.1 Consistency Part Due to Projection

The first part in (8.1) is the consistency part. It can be computed for each virtual element

$$U_c(\boldsymbol{u}_\pi\,,\mathbf{h}_v) = \int\limits_{\Omega_v} \left[\Psi(\nabla\boldsymbol{u}_\pi,\mathbf{h}_v) - \bar{\boldsymbol{f}}\cdot\boldsymbol{u}_\pi\right]\mathrm{d}\Omega - \int\limits_{\Gamma_v^\sigma} \bar{\boldsymbol{t}}\cdot\boldsymbol{u}_\pi\;\mathrm{d}\Gamma \tag{8.2}$$

and includes the volume and traction loading.

With the projection for the displacement $\boldsymbol{u}_\pi = \mathbf{H}_{u\,\pi}^{(d,n)}(\mathbf{X})\,\mathbf{u}_v$ and the gradient $\nabla\boldsymbol{u}_\pi = \mathbf{B}_{u\,\pi}^{(d,n)}(\mathbf{X})\mathbb{P}_\nabla^{(d,n)}\mathbf{u}_v$ the above equation leads to

$$
\begin{aligned}
U_c(\mathbf{u}_v\,, \mathbf{h}_v) = &\int\limits_{\Omega_v} \Psi(\mathbf{B}_{u\,\pi}^{(d,n)}(\mathbf{X})\mathbb{P}_{\nabla}^{(d,n)}\mathbf{u}_v, \mathbf{h}_v)\,\mathrm{d}\Omega \\
&- \mathbf{u}_v^T \int\limits_{\Omega_v} [\mathbf{H}_{u\,\pi}^{(d,n)}(\mathbf{X})]^T\,\bar{\boldsymbol{f}}\mathrm{d}\Omega - \mathbf{u}_v^T \int\limits_{\Gamma_v^\sigma} [\mathbf{H}_{u\,\pi}^{(d,n)}(\mathbf{X})]^T\bar{\boldsymbol{t}}\,\mathrm{d}\Gamma .
\end{aligned}
\tag{8.3}
$$

The discretization of the deformation gradient $\boldsymbol{F}$ allows to compute the kinematical relation (2.74) for the virtual element Ω_v

$$
\boldsymbol{b}_e = \boldsymbol{F}\,\boldsymbol{C}_p^{-1}\,\boldsymbol{F}^T \quad \text{with} \quad \boldsymbol{F} = \boldsymbol{1} + \nabla\boldsymbol{u}_\pi = \mathbf{1} + \mathbf{B}_{u\,\pi}^{(d,n)}(\mathbf{X})\mathbb{P}_{\nabla}^{(d,n)}\mathbf{u}_v\,. \tag{8.4}
$$

As in the elastic case, $\nabla\,\boldsymbol{u}_\pi$ is constant for the linear ansatz $\nabla\boldsymbol{u}_\pi = \mathbf{B}_{u\,\pi}^{(d,1)}\mathbb{P}_{\nabla}^{(d,1)}\mathbf{u}_v$. Hence, all kinematical quantities $\boldsymbol{F}$, $\boldsymbol{b}_e$ and the history variables $\boldsymbol{C}_p^{-1}$ and γ are constant as well. Thus the integration of the potential function can be simplified, leading to the efficient form

$$
\int\limits_{\Omega_v} \Psi(\boldsymbol{b}_e, \mathbf{h}_v)\,\mathrm{d}\Omega = \Psi(\boldsymbol{b}_e, \mathbf{h}_v)\,\Omega_v \tag{8.5}
$$

where Ω_v is the area/volume of the virtual element. Note that the pseudo strain energy $\Psi(\boldsymbol{b}_e, \mathbf{h}_v)$ is still a nonlinear function with respect to the displacement nodal degrees of freedom and the plastic history variables $\mathbf{h}_v$.

8.1.2 Algorithmic Treatment of Finite Strain Elasto-plasticity

The evolution equation in (2.80) can be integrated by using an exponential map, see e.g. Korelc and Stupkiewicz (2014), which is the most accurate and efficient way. This leads with the evolution of the hardening variable $\Delta\alpha = \Delta\gamma$, see (2.79), to the incremental form of (2.80)

$$
\boldsymbol{C}_p^{-1} = \boldsymbol{F}^{-1}\exp[2\Delta\alpha\,\boldsymbol{n}]\,\boldsymbol{F}\,\boldsymbol{C}_{p\,n}^{-1} \tag{8.6}
$$

where the time integration is performed within the time step $\Delta t = t - t_n$. In the following, quantities without an index represent the current time ($t = t_{n+1}$) and quantities with index n are related to the last time step t_n. The relation (8.6) returns the plastic variables exactly due to the use of the exponential map. Consequently the plastic incompressibility of the J_2 plasticity model is exactly satisfied.

The equations in Algorithm 1 have now to be solved locally at element level. Since all kinematical variables are constant at element level for the low order discretization the following set of equations has to be solved at the current time $t = t_{n+1}$

Algorithm 1: Return map algorithm for finite strain elasto-plasticity

```
Given: $\boldsymbol{F}$, $\boldsymbol{C}_{pn}^{-1}$, $\alpha_n$      Find: $\boldsymbol{C}_p^{-1}$, $\alpha$ ;
$\boldsymbol{b}_e = \boldsymbol{F}\,\boldsymbol{C}_{pn}^{-1}\,\boldsymbol{F}^T$;
$I_1 = \operatorname{tr}\boldsymbol{b}_e\,,\ \ I_3 = \det\boldsymbol{b}_e$;
$W = \frac{\mu}{2}(I_1 - 3 - \ln I_3) + \frac{\lambda}{4}(I_3 - 1 - \ln I_3)$;
$\boldsymbol{\tau}_e = 2\,\boldsymbol{b}_e\,\frac{\partial W}{\partial \boldsymbol{b}_e}$ ;
$\boldsymbol{s}_e = \boldsymbol{\tau}_e - \frac{1}{3}\operatorname{tr}\boldsymbol{\tau}_e$;
$f(\boldsymbol{s}_e,\alpha) = \sqrt{\frac{3}{2}}\,\|\,\boldsymbol{s}_e\,\| - [\,Y_0 + (Y_\infty - Y_0)\,e^{-\delta\,\alpha} + H\alpha\,]$;
$\boldsymbol{n} = \frac{\partial f(\boldsymbol{s}_e,\alpha)}{\partial \boldsymbol{s}_e}$ ;
$\hat{\mathbf{Q}}_v = \boldsymbol{F}\,\boldsymbol{C}_p^{-1} - \exp[2(\alpha - \alpha_n)\,\boldsymbol{n}]\,\boldsymbol{F}\,\boldsymbol{C}_{pn}^{-1}$ ;
Result: Evolution equation, flow rule and history variables
$\mathbf{Q}_v = \left\{\hat{\mathbf{Q}}_v\,,\,f\right\} = \{Q_{11},\ Q_{22},\ Q_{33},\ Q_{12},\ Q_{13},\ Q_{23},\ f\}^T$ ;
$\mathbf{h}_v = \{\boldsymbol{C}_p^{-1},\alpha\} = \{C_{p11}^{-1} - 1, C_{p22}^{-1} - 1, C_{p33}^{-1} - 1, C_{p12}^{-1}, C_{p13}^{-1}, C_{p23}^{-1}, \alpha\}^T$
```

$$
\begin{aligned}
\mathbf{R}(\mathbf{u},\mathbf{h},\mathbf{h}_n) &= \mathbf{0} \\
\mathbf{Q}_v(\boldsymbol{F}_v\,,\mathbf{h}_v,\mathbf{h}_{vn}) &= \mathbf{0} \quad v = 1, 2, \ldots, n_v\,.
\end{aligned}
\tag{8.7}
$$

Here $\mathbf{R}$ is the residual stemming from the first variation of the pseudo-potential which is equivalent to the weak form. It contains the total number of unknowns $\mathbf{u}$ of the problem, and with $\mathbf{h}$ all sets of history variables related to the n_v virtual elements of a discretization. The equation set $\mathbf{Q}_v$ includes n_h local equation systems, as defined in Algorithm 3. These have to be solved for the update of the plastic variables. Classically the equations $\mathbf{Q}_v$ have to be evaluated at Gauss point level within an element. Since the discretization of the three-dimensional virtual element is based on a linear polynomial ansatz leading to a constant gradient in Ω_v only one evaluation per virtual element v is necessary. The n_h equations of $\mathbf{Q}_v$ return the history variables $\mathbf{h}_v$. The system of nonlinear equations is classically solved using a nested algorithm where $\mathbf{u}$ and $\mathbf{h}$ are the unknown variables. In this algorithm the local Eq. (8.7)$_2$ are solved for fixed $\mathbf{u}$ displacements using a Newton-Raphson scheme as described in Algorithm 1. The Newton-Raphson method is again applied to solve the set of global Eq. (8.7)$_1$ freezing now the history variables, for details see Simo and Hughes (1998), Korelc and Wriggers (2016).

The element residual $\mathbf{R}_v$ of the consistency part of a virtual element follows from (8.5)

$$
\mathbf{R}_v^c = \Omega_v \; \frac{\partial\Psi(\mathbf{u}_v\,,\mathbf{h}_v)}{\partial\mathbf{u}_v}\bigg|_{\frac{D\mathbf{h}_v}{D\mathbf{F}_v}=\mathbf{0}}
\tag{8.8}
$$

where the history variables are treated as fixed variables and Ψ depends on $\mathbf{u}_v$, see (8.3) and (8.4). The explicit form of the residual was derived with the symbolic tool

AceGen, details can be found in Korelc and Wriggers (2016). The nested algorithm requires the solution of the local Algorithm 1 which results in the nonlinear equation system $(8.7)_2$ and has to be solved using a Newton iteration. For this task the local tangent matrix $\mathbf{A}_v$, related to the constitutive equation, has to be computed

$$\mathbf{A}_v = \frac{\partial \mathbf{Q}_v(\boldsymbol{b}, \mathbf{h}_v, \mathbf{h}_{vn})}{\partial \mathbf{h}_v} \,. \tag{8.9}$$

Based on this result, the global tangent matrix for the virtual element can be derived from $(8.7)_1$ and (8.8)

$$\mathbf{K}^c_{Tv} = \left. \frac{\partial \mathbf{R}^c_v(\mathbf{u}_v, \mathbf{h}_v, \mathbf{h}_{vn})}{\partial \mathbf{u}_v} \right|_{\frac{D\mathbf{h}_v}{D\boldsymbol{F}_v} = -\mathbf{A}_v^{-1} \frac{\partial \mathbf{Q}_v}{\partial \boldsymbol{F}_v}} \tag{8.10}$$

where $\mathbf{u}_v$ are the nodal displacements of the virtual element Ω_v, see (3.58) and (3.157). The introduction of the special constraint $\frac{D\mathbf{h}_v}{D\boldsymbol{F}_v} = -\mathbf{A}_v^{-1} \frac{\partial \mathbf{Q}_v}{\partial \boldsymbol{F}_v}$ in (8.10) is necessary within the tool *AceGen* to ensure a consistent linearization for the global algorithm that takes into account the local iteration, for details see Simo and Hughes (1998), de Souza Neto et al. (2008), Korelc and Stupkiewicz (2014), Korelc and Wriggers (2016).

8.1.3 Energy Stabilization of the Virtual Element for Finite Plasticity

A difference between two potential energies can be introduced to formulate the stabilizations, like in the case of hyperelastic materials. The pseudo potential energy related to stabilization is given for a virtual element v by

$$U_{stab}(\boldsymbol{u}_h \,, \mathbf{h}_v) = \hat{U}(\boldsymbol{u}_h \,, \mathbf{h}_v) - \hat{U}(\boldsymbol{u}_\pi \,, \mathbf{h}_v) \,. \tag{8.11}$$

As for the computation of (6.61) an internal mesh can be created for the evaluation of $\hat{U}(\boldsymbol{u}_h \,, \mathbf{h}_v)$, see Fig. 6.16. Based on this internal mesh, the displacement gradient Grad $\boldsymbol{u}_m$ is computed in each internal tetrahedral element Ω^i_m. It is constant, as well as the deformation gradient $\boldsymbol{F}^i_m$, due to the linear ansatz.

In case of plastic deformations the left Cauchy Green tensor $\boldsymbol{b}_e|_m$ has to be introduced in the stabilization part of the pseudo energy (8.11). This is achieved in an approximative way using the constant plastic strains $\boldsymbol{C}_p^{-1}(\mathbf{h}_v)$ from the consistency part, see Algorithm 3. The kinematic Eq. (2.74) yields

$$\boldsymbol{b}_e|_m = \boldsymbol{F}_m \, \boldsymbol{C}_p^{-1}(\mathbf{h}_v) \, \boldsymbol{F}_m^T \tag{8.12}$$

which then can be utilized together with the strain energy (2.56) to compute the stabilization part.

With the above approximations the expression $\hat{U}(\boldsymbol{u}_h\,,\mathbf{h}_v)$, defining part of the stabilization potential in (8.11), is obtained for the virtual element v by including all n_m internal tetrahedral elements

$$\hat{U}(\boldsymbol{u}_h\,,\mathbf{h}_v) = \mathop{\mathbf{A}}_{v=1}^{n_v} \int\limits_{\Omega_m^i} \hat{\Psi}(\mathbf{u}_h|_m\;,\mathbf{h}_v)\,d\Omega\,. \tag{8.13}$$

Here $\mathbf{u}_h|_m$ are the nodal displacements that are related to the specific tetrahedral element Ω_m^i. For more details see Sect. 6.2.3.

Once the potential is evaluated the residual vector $\mathbf{R}_e^s$ and the tangent matrix $\mathbf{K}_{Tv}^s$ of the stabilization part can be obtained

$$\mathbf{R}_v^s = \left.\frac{\partial \hat{U}_{stab}(\mathbf{u}_v\,,\mathbf{h}_v)}{\partial \mathbf{u}_v}\right|_{\frac{D\mathbf{h}_v}{D\mathbf{F}_v}=\mathbf{0}} \quad \text{and} \quad \mathbf{K}_{Tv}^s = \frac{\partial \mathbf{R}_v^s(\mathbf{u}_v,\mathbf{h}_v)}{\partial \mathbf{u}_v}\,. \tag{8.14}$$

The total residual and tangent matrix of the virtual element for finite plasticity are given by the sum of expressions (8.8), (8.10) and (8.14)

$$\mathbf{R}_v = \mathbf{R}_v^c + \mathbf{R}_v^s \quad \text{and } \mathbf{K}_{Tv} = \mathbf{K}_{Tv}^c + \mathbf{K}_{Tv}^s\,.$$

Note that the approximation in (8.12) yields a non-symmetric tangent matrix for the stabilization part, since the plastic strains $\boldsymbol{C}_p^{-1}$ from the consistency part are used.

The values of the Lamé parameters in the pseudo potential (8.11) have to be different from the ones that describe the physical problem. A procedure for a proper selection of these constitutive parameters was described in Sect. 6.2.3. This procedure can be applied for plasticity in the same way. One only has to take care that, in case of a plastic deformation in a virtual element, the modulus of elasticity $\hat{E}$ has to be reduced to account for the decreased stiffness due to hardening. A possible reduction is based on the von Mises stress σ_{VM}, see (2.76), and the equivalent plastic strain α

$$\hat{E} = \min\left\{ E\,\frac{\beta}{1+\beta}\,,\max\left(\frac{\sigma_{VM}}{\alpha}\,,\eta_2\,E\right)\right\}\,, \tag{8.15}$$

providing an approximation for the tangent of the hardening curve. The parameter $\eta_2 = 10^{-3}$ is a safeguard ensuring that the influence of the stability term does not disappear for very large plastic deformations where the hardening modulus approaches zero. The value of η_2 has to be sufficiently low in order not to influence the results of the numerical simulation, see Böhm et al. (2023).

8.2 Numerical Examples

The application of virtual elements for inelastic problems undergoing finite deformations is demonstrated in this section. To account for the plastic incompressible behaviour in J_2-theory the Hu-Washizu mixed element formulation from Sect. 6.3.3 is employed, leading to VE-P0 elements. However, it will be shown that pure virtual elements also perform well when the incompressiblity constraint is present, see e.g. the first example in Sect. 6.3.4. Two classical tests related to the statical and the dynamical behaviour are discussed. First, a tension test is considered which leads to necking of a bar. This is followed by the Taylor-Anvil test which is commonly used to measure dynamical material constants.

8.2.1 Necking of Cylindrical Bar

Necking is generally observed when a bar is subjected to tension loading. Its physical model has to include finite elasto-plasticity. In this numerical example necking in a cylindrical bar is considered, see e.g. Hudobivnik et al. (2018), which is classically observed in a tension test of steel bars, see Fig. 8.1.

Experimental observations show that such bars tend to neck when large plastic deformations occur. It is a standard benchmark problem of finite plasticity and has been analyzed by many authors, see e.g. Simo et al. (1993). In order to compute the localization of plastic strains in the necking area one needs a robust and reliable discretization method. Hence this example can be used to test the robustness of the virtual element formulation. The geometrical setup and the boundary conditions of the cylindrical bar with diameter $d = 1$ mm and length $L = 10$ mm are depicted in Fig. 8.1.

To trigger localization and necking in the center of the bar, a geometrical imperfection was introduced in the central zone. In detail, the specimen's net cross section is reduced by choosing the diameter at the center to be $d_c = 0.99\, d$. However, also other imperfections can be applied as well as a pure stability analysis, see e.g. Wriggers et al. (1992). At the left cross section of the bar a Dirichlet boundary condition of $\bar{\boldsymbol{u}} = \mathbf{0}$ is applied.

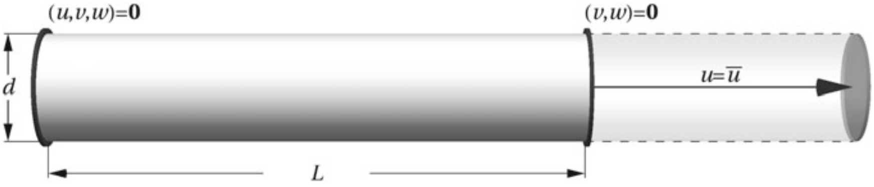


Fig. 8.1 Necking of cylindrical bar. Geometry and boundary conditions

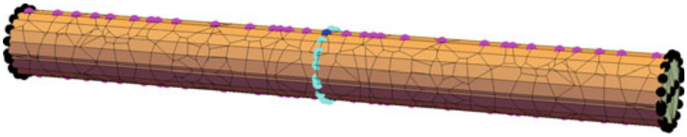


Fig. 8.2 Voronoi mesh for $N = 1$ (8×80 elements)

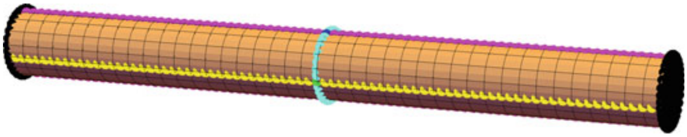


Fig. 8.3 Regular H2 and H1 mesh for $N = 1$ ($8 \times 20 \times 2 \times 2$ elements)

The loading is provided by a prescribed displacement $\bar{u}$ in axial direction at the right side. It has in total the magnitude of 50 % of the bar length L leading to $\bar{u} = 0.5L$.

Different mesh types are employed within the necking analysis. These are regular meshes using as basis 8 noded hexahedra (H1) and 26 noded hexahedra (H2) for the linear VEM element. Additionally three-dimensional meshes consisting of Voronoi cells were generated. The following elements were employed in this study: the linear three-dimensional virtual element, discussed in Sect. 3.2.3, here denoted VE-H2 with 26 nodes (4 faces per each block side) and a quadratic finite element (FE-H2) with 27 nodes, both use the same mesh, although the VE-H2 has no nodal point at the centroid. Additionally linear FE-H1 and FE-H1-P0 finite elements were used as a reference involving regular and distorted meshes. Here the FE-H1-P0 is formulated with as the mixed Hu-Washizu formulation (H1-P0), see e.g. Korelc and Wriggers (2016). The Voronoi mesh was modeled with the virtual element formulation denoted by (VE-VO). Voronoi, H2 and H1 meshes are depicted in Figs. 8.2, 8.3 and 8.4, respectively. The regular meshes used were automatically generated by the meshing tools provided in *AceFEM*. With this tool the local coordinates ξ and η are mapped into a circle by using the transformation function:

$$y = d\,\frac{\xi}{2}\sqrt{1-\frac{\eta^2}{2}}, \quad z = d\,\frac{\eta}{2}\sqrt{1-\frac{\xi^2}{2}}, \quad \xi \in [-1,1]\,, \quad \eta \in [-1,1]\,. \tag{8.16}$$

The Voronoi meshes for this example were generated by the open-source software package Neper, see Quey et al. (2011). Figure 8.2 shows a Voronoi mesh generated from random seeds which produce arbitrary number of nodes and element sizes.

Due to the symmetry of the boundary value problem, only one eights of the cylinder was discretized and the boundary conditions were modified accordingly. Different mesh densities were employed to compute the solution for all generated mesh types. The mesh refinement is uniform in the sense that finer meshes are included in the coarser meshes. This enables convergence studies that will depict

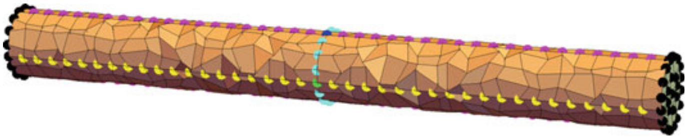


Fig. 8.4 Distorted H1 mesh for $N = 1$ ($8 \times 20 \times 2 \times 2$ elements)

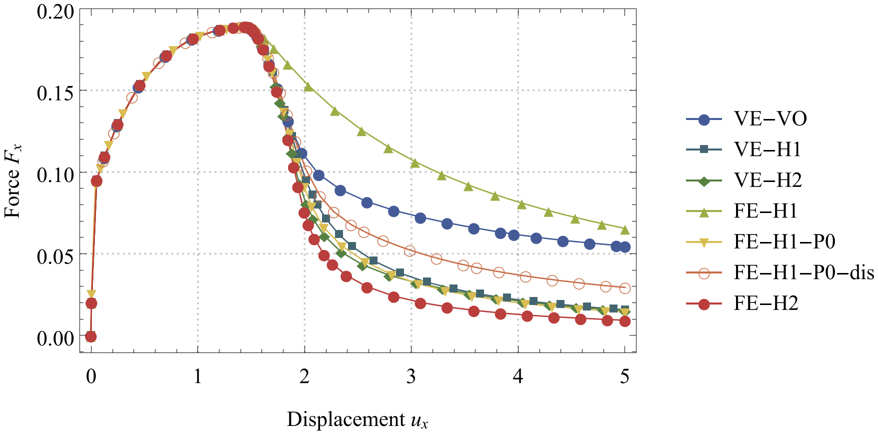


Fig. 8.5 Load deflection curve of the necking problem for different types of elements at $N = 3$

differences of the formulations. The mesh was defined through a parameter N with division $10 \cdot 2^N \times 2^N \times 2^N$.

The load deflection curve for a mesh with $80 \times 8 \times 8$ elements, $N = 3$, is depicted in Fig. 8.5, where different elements and mesh types are compared. Figure 8.5 shows a typical necking response with initial hardening until 1.5 mm displacement is reached. All elements and meshes recover the constant strain part before the necking occurs. They also, besides the FE-H1 element, recover the necking and associated localization up to a displacement of $0.18\,L$ properly. Note, that this is actually the point when a real specimen will break apart in an experimental test. This means that the virtual element method either using a regular H2 mesh or a Voronoi mesh provides good solutions for engineering problems. Only for displacements $\bar{u} > 0.18\,L$ the results are dependent on the discretization, which is most apparent when the FE-H1 element is applied.

The virtual elements with regular meshes and the finite elements FE-H1-P0 and FE-H2 almost yield the same result, showing the very good response for very large strains. The irregular meshes however lead to a stiffer response, see VE-VO and H1 in Fig. 8.4, and thus provide a solution which does not capture the localization for severe strains properly. Nevertheless the Voronoi mesh leads to a less stiff response than FE-H1. A similar behaviour for Voronoi cells is also reported for two-dimensional virtual element formulations for finite elasto-plastic strain in Wriggers and Hudobivnik

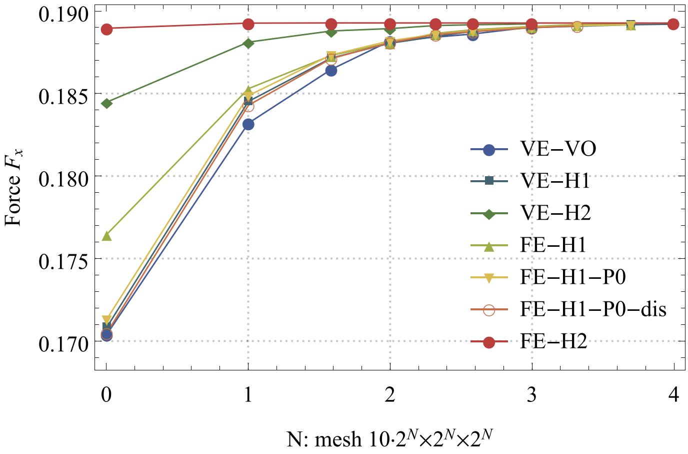


Fig. 8.6 Convergence study of the maximum load values of the necking problem at around $0.14L$ for different mesh densities N

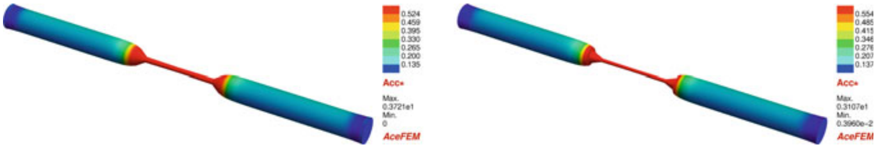


Fig. 8.7 Necking of cylindrical bar: FE-H2 and VE-H2

(2017), while the regular VE-H2 and VE-H1 mesh produced comparable results to FE-H2 and FE-H1-P0 formulations in the two-dimensional case.

Mesh convergence of the proposed 3D virtual element is investigated in Fig. 8.6 using the maximum horizontal force at the right end of the cylinder (at around $0.14L$). Herein, all formulations converge for higher mesh density parameters N. Again we note the very good convergence for the VE-H2 mesh.

Figures 8.7, 8.8 and 8.9 show the distribution of the equivalent plastic strain α at the final deformation state for all elements with mesh division $N = 3$. These figures illustrate the good agreement between all discretizations in the necking area and the capability of the virtual element to solve finite elasto-plastic strain problems. Most meshes (except VE-VO and FE-H1) lead to a comparable distribution of the equivalent plastic strains. The meshes with FE-H2 and VE-H2 produce more localized shapes than the other meshes. The Voronoi (VE-VO) and especially the FE-H1 discretizations exhibit a stiffer response, which results in a larger necking diameter.

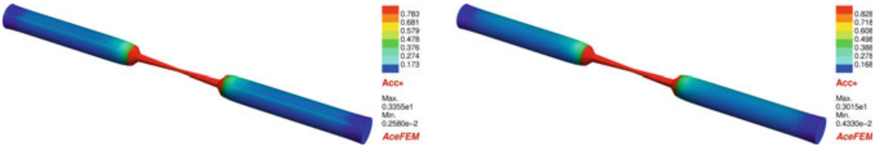


Fig. 8.8 Necking of cylindrical bar: FE-H1-P0 and VE-H1

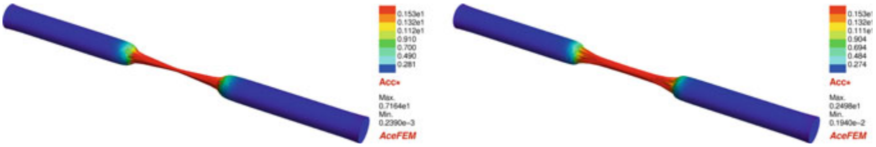


Fig. 8.9 Necking of cylindrical bar: FE-H1 and VE-VO

8.2.2 Taylor Anvil Test

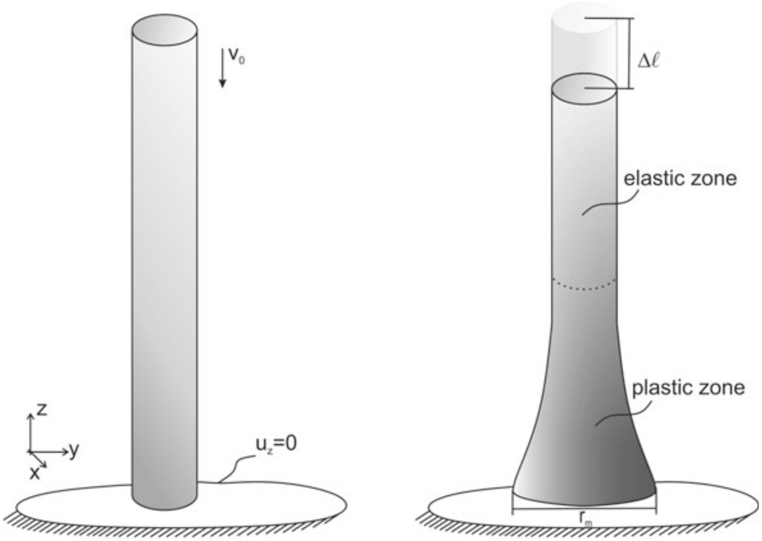
The Taylor-Anvil test is widely employed to characterize the dynamical behaviour of metals. Furthermore, it is used to validate discretization schemes that simulate elasto-plastic finite strain states undergoing dynamic loading, see Simo (1992), Taylor (1948). In the test setup, a rod impacts at high velocity a rigid plate. This is modelled by fixing in longitudinal direction one side of the rod and by prescribing an initial velocity to all other parts of the body, as depicted in Fig. 8.10a.

The material model described in Sect. 2.2.3.2 is applied in the associated finite strain computations together with the flow rule (2.76). Material parameters for the simulations are taken from the literature, see Kamoulakos (1990), Zhu and Cescotto (1995), Camacho and Ortiz (1997), Li et al. (2010), Kumar et al. (2019), in order to be able to compare the results of different discretization schemes with the virtual element method. The elastic behaviour is governed by the two constants: the Youngs modulus $E = 117$ kN/mm^2 and the Poisson ratio of $\nu = 0.35$. The density of the rod is $\rho = 8930$ g/mm^3. The yield stress is $Y_0 = 0.400$ kN/mm^2 and the linear hardening coefficient has the value $H = 0.100$ kN/mm^2. The saturation parameter is set to zero ($\delta = 0$) in the flow rule (2.76). Hence the exponential term disappears and the model is reduced to linear hardening. The initial velocity is set to $v_0 = 227$ m/s. The time increment for the dynamic simulation is $\Delta t = 0.01 \mu$s. During the impact a plastic front develops and moves upwards leading to a deformed state as shown in Fig. 8.10b.

The accumulated equivalent plastic strain is depicted for all element formulations in Fig. 8.11 at the final deformation state. The numerical solution is obtained with 10.000 elements. As expected, large plastic deformations are observed at the end of the rod where the rod strikes the wall. This is well documented in the literature, see Taylor (1948), Kumar et al. (2019), Taylor and Papadopoulos (1993) and clearly demonstrates the influence of the kinetic energy, resulting in high stresses at the front of the rod and a plastic wave traveling in the direction of the free end of the rod. The wave stops travelling when a certain amount of energy is dissipated by the plastic process. Hence, the stresses in the upper part of the rod are not affected by the plastic wave and stay in the elastic range and elastic energy is still stored in the upper part of the rod, as depicted in Fig. 8.11 by zero accumulated plastic strain. Due to this elastic oscillations are visible in Fig. 8.12, however, having a very small amplitude. The oscillation starts at a time around 80 μs.

The contour plots show that all element yield similar results except the FE-H1 discretization. Figure 8.12 demonstrates the length change over time for different element formulations and two discretization with 640 and 10.000 elements. All element types show nearly the same displacement curves over time. Locking effects for this impact test occur only for the FE-H1 discretization. A small oscillation with high frequency can be observed for all formulations. It is related to the elastic response of the rod and more pronounced for the finer mesh. A good coarse mesh accuracy can be attested to the virtual element formulations.

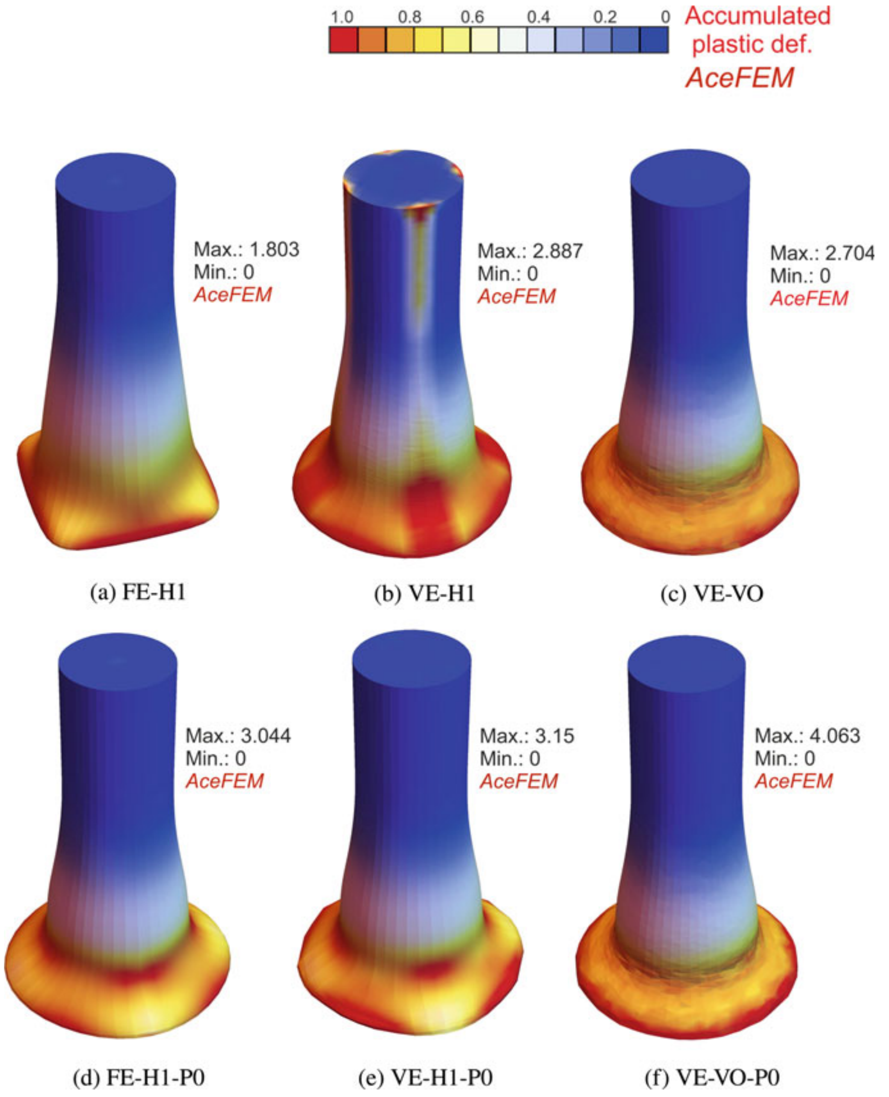
Figure 8.13 depicts the development of the mushroom radius at the lower part ($z = 0$). Good agreement between all element formulations—besides the locking FE-H1—is observed. Again the mixed formulation converges for finite and virtual elements (FE/VE-H1-P0) closest to the results which are available in the literature for this problem, see Tables 8.1 and 8.2. We note that the results using the OTM method in Li et al. (2010) and Li et al. (2010) seems to depict minor locking as well as the VE-H1 discretization. Figure 8.13a demonstrates again the good coarse mesh accuracy of the mixed FE/VE-H1-P0 formulation.



(a) Geometry and boundary value problem

(b) Deformed state (schematic illustration)

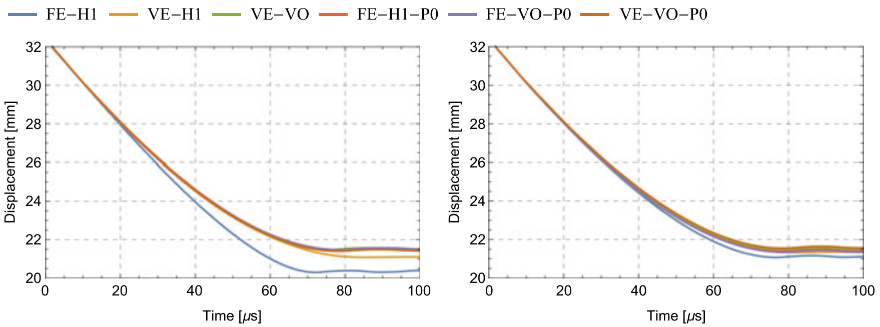
Fig. 8.10 Taylor Anvil test



(a) FE-H1 (b) VE-H1 (c) VE-VO

(d) FE-H1-P0 (e) VE-H1-P0 (f) VE-VO-P0

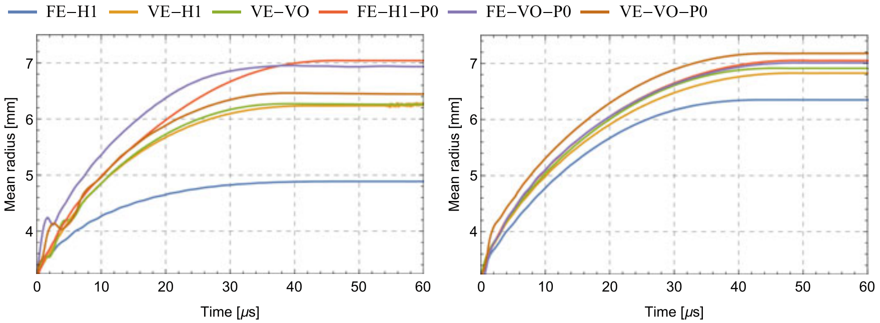
Fig. 8.11 Taylor Anvil test—deformation state for different elements, depicting the accumulated plastic strain $\varepsilon^p_{acc} = \sqrt{\frac{2}{3}}\int_0^t \dot{\gamma}\,\mathrm{d}t$



(a) No. elem.=640

(b) No. elem.=10000

Fig. 8.12 Taylor Anvil test—length change over time for different element numbers and formulations



(a) No. elem.=640

(b) No. elem.=10000

Fig. 8.13 Taylor Anvil test—evolution of the mushroom radius r_m for different element numbers and formulations

Table 8.1 Taylor Anvil test—comparison of different results obtained in the literature

Source	Method	Max. acc. plastic strain	Final height (mm)	Final mushroom radius (mm)
Kamoulakos (1990)	FEM	2.47–3.24	21.47–21.66	7.02–7.12
Zhu and Cescotto (1995)	FEM	2.75–3.03	21.26–21.49	6.89–7.18
Camacho and Ortiz (1997)	FEM	2.97–3.25	21.42–21.44	7.21–7.24
Li et al. (2010)	OTM	3.0	21.43	6.8
Li et al. (2010)	OTM	2.69	21.45	6.84

Table 8.2 Taylor Anvil test—comparison of results for different element types

Element	Max. acc. plastic strain	Final height (mm)	Final mushroom radius (mm)
FE-H1	1.803	21.09–21.17	6.34–6.35
VE-H1	2.887	21.32–21.36	6.81–6.83
VE-VO	2.704	21.51–21.59	6.89–6.91
FE-H1-P0	3.04	21.41–21.5	7.04–7.05
VE-H1-P0	3.15	21.36–21.45	6.99–7.01
VE-VO-P0	4.063	21.56–21.65	7.17–7.18

It is obvious that the underlying discretization schemes produce slightly different results, see Table 8.1. When these results are compared with the values obtained with the virtual element method, as depicted in Table 8.2, it can be observed that good agreement is achieved for all proposed virtual element formulations. But especially the mixed VE-H1-P0 discretization yields superior results and it can be concluded that the mixed formulation should be used for problems involving elasto-plastic response.

References

Böhm, Ch., J. Kruse, M. Stonis, F. Aldakheel, and P. Wriggers. 2020. Virtual element method for cross-wedge rolling during tailored forming processes. *Procedia Manufacturing* 47: 713–718.

Böhm, C., L. Munk, B. Hudobivnik, F. Aldakheel, J. Korelc, and P. Wriggers. 2023. Virtual elements for computational anisotropic crystal plasticity. *Computer Methods in Applied Mechanics and Engineering* 405: 115835.

Camacho, G., and M. Ortiz. 1997. Adaptive Lagrangian modelling of ballistic penetration of metallic targets. *Computer Methods in Applied Mechanics and Engineering* 142: 269–301.

de Souza Neto, E.A., D. Peric, and D.R.J. Owen. 2008. *Computational methods for plasticity, theory and applications*. Wiley, Chichester.

Hudobivnik, B., F. Aldakheel, and P. Wriggers. 2018. Low order 3D virtual element formulation for finite elasto-plastic deformations. *Computational Mechanics* 63: 253–269.

Kamoulakos, A. 1990. A simple benchmark for impact. *Bench Mark* 31–35.

Korelc, J., and S. Stupkiewicz. 2014. Closed-form matrix exponential and its application in finite-strain plasticity. *International Journal for Numerical Methods in Engineering* 98: 960–987.

Korelc, J., and P. Wriggers. 2016. *Automation of finite element methods*. Berlin: Springer.

Kumar, S., K. Danas, and D.M. Kochmann. 2019. Enhanced local maximum-entropy approximation for stable meshfree simulations. *Computer Methods in Applied Mechanics and Engineering* 344: 858–886.

Li, B., F. Habbal, and M. Ortiz. 2010. Optimal transportation meshfree approximation schemes for fluid and plastic flows. *International Journal for Numerical Methods in Engineering* 83 (12): 1541–1579.

Quey, R., P. Dawson, and F. Barbe. 2011. Large-scale 3D random polycrystals for the finite element method: Generation, meshing and remeshing. *Computer Methods in Applied Mechanics and Engineering* 200 (17–20): 1729–1745.

Simo, J.C. 1992. Algorithms for static and dynamic multiplicative plasticity that preserve the classical return mapping schemes of the infinitesimal theory. *Computer Methods in Applied Mechanics and Engineering* 99: 61–112.

Simo, J.C., and T.J.R. Hughes. 1998. *Computational inelasticity*. New York, Berlin: Springer.

Simo, J.C., F. Armero, and R.L. Taylor. 1993. Improved versions of assumed enhanced strain trilinear elements for 3D finite deformation problems. *Computer Methods in Applied Mechanics and Engineering* 110: 359–386.

Taylor, G.I. 1948. The use of flat-ended projectiles for determining dynamic yield stress I: Theoretical considerations. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 194(1038):289–299.

Taylor, R.L., and P. Papadopoulos. 1993. On a finite element method for dynamic contact/impact problems. *International Journal for Numerical Methods in Engineering* 36 (12): 2123–2140.

Wriggers, P., and B. Hudobivnik. 2017. A low order virtual element formulation for finite elasto-plastic deformations. *Computer Methods in Applied Mechanics and Engineering* 327: 459–477.

Wriggers, P., C. Miehe, M. Kleiber, and J. Simo. 1992. A thermomechanical approach to the necking problem. *International Journal for Numerical Methods in Engineering* 33: 869–883.

Zhu, Y., and S. Cescotto. 1995. Unified and mixed formulation of the 4 node quadrilateral elements by assumed strain method: Application to thermomechanical problems. *International Journal for Numerical Methods in Engineering* 38: 685–716.

Chapter 9
Virtual Elements for Thermo-mechanical Problems

Many engineering problems require an analysis that takes into account more than one field. These are known as coupled problems which can be split in two distinct categories. The first category relates to problems in which the physical domains overlap and coupling occurs via the differential equations of the different physical phenomena. In the second category problems are investigated in which coupling occurs only at the domain interfaces, like in fluid-structure interaction. Here we will concentrate on the first category. Examples are thermo-mechanical, phase field and electro-mechanical problems, among others. The single fields are governed by different physical models. In general, the behaviour of each field is influenced by the other fields that are present in the model. Additional complexity arises for the proper setup of numerical simulation procedures when the partial differential equations, describing the single fields, are of different type. Hence, coupled multi-field and multi-physics problems span on one side a vast area of applications and on the other side they are demanding with respect to discretizations and algorithms.

Some applications using the virtual element method in the area of coupled problems can be found in Beirão da Veiga et al. (2021) and Berrone and Busetto (2022) for flow in porous media, in Liu et al. (2019) for a Stokes–Darcy problem, in Böhm et al. (2021) for electro-magneto-mechanical polycristalline materials and in Dhanush and Natarajan (2019), Aldakheel et al. (2019) and Antonietti et al. (2022) for thermo-mechanical applications.

Here we discuss as a specific modeling example thermo-mechanical coupling for problems undergoing finite strains.

9.1 Introduction

Heat treatment processes relate to a field of research with various engineering applications which cover forming, machining and cutting of different components but also heating of car tires due to rolling and additive manufacturing processes. For instance,

P. Wriggers et al., *Virtual Element Methods in Engineering Sciences*,
https://doi.org/10.1007/978-3-031-39255-9_9

heating of a steel bolt and cross wedge rolling are processes under thermo-mechanical loading conditions. These processes exhibit complex coupling phenomena:

1. heat effects the mechanical response by thermal expansion and temperature dependent mechanical properties;
2. mechanical loading changes the geometry of the solid which influences thermal fields, furthermore, inelastic processes—like plasticity and frictional contact—lead to heat dissipation.

All applications significantly benefit from precise predictive computational tools to model coupling phenomena during deformation and heating processes which are based on models allowing for a complete coupling with regard to the two points stated above.

Over the last 40 years research has been pursued in this area with relation to numerical methods, especially finite element technologies. This research is based as well on purely phenomenological as on micro-mechanically motivated continuum formulations. In Argyris and Doltsinis (1981) several numerical concepts for the coupled processes were developed that based on the so-called natural formulation of finite elements. A more general formulation and algorithmic approach can be found in Zienkiewicz and Chan (1989), but still in the region of small mechanical strains. Later Simo and Miehe (1992), Wriggers et al. (1992) investigated coupled associative thermo-plasticity at finite strains, addressing in detail the numerical analysis aspects. A micro-mechanical approach for modeling coupled thermo-crystal-plasticity can been found in Stainier et al. (2002), Chapuis and Driver (2011) and Cereceda et al. (2016). A physical approach based on nonlinear rheological models was introduced in Lion (2000) to describe finite thermoviscoplasticity. Anand et al. (2009) and Miehe et al. (2011) outlined constitutive models for finite thermo-visco-plastic behaviour of amorphous glassy polymers and considered details of the associated numerical implementation. A variational formulation for the coupled multi-field problem was introduced in Čanađija and Mosler (2011) and Bartels et al. (2015) based on the works Yang et al. (2006) and Stainier and Ortiz (2010).

Solution methods for thermo-mechanical problems, using the finite element method as a discretization scheme, can be found in textbooks, see e.g. Zienkiewicz and Taylor (2000) and Wriggers (2008). In recent years different methods were introduced which added new features to the numerical analysis of coupled problems in solid mechanics. Here one can mention isogeometric analysis which is outlined in Hughes et al. (2005) and Cottrell et al. (2009).

In this chapter the virtual element method will be presented as an alternative approach for finite strain thermo-plasticity problems. Low-order formulations with linear ansatz will be considered for problems in two and three dimensions, see e.g. Wriggers and Hudobivnik (2017) and Hudobivnik et al. (2018). The continuum formulation and the associated discretization scheme are based on a minimization of the pseudo energy which was already applied in Chap. 8. Von Mises J_2-plasticity, see Sect. 2.2.3, is employed in the mechanical part including incompressibility of the plastic deformation. Nonlinear isotropic hardening is introduced as in Sect. 8.1.2, see also Simo (1988a, b). The thermal part is based on the work of

Wriggers et al. (1989) and Simo and Miehe (1992) which includes the effect of temperature on the mechanical fields and results in a thermal expansion and the influence of a change of geometry on heat conduction.

The virtual element formulation in this chapter is based on Wriggers and Hudobivnik (2017), Hudobivnik et al. (2018) and Aldakheel et al. (2019).

9.2 Governing Equations

The basic equations for finite strain problems can be found in Sects. 2.1.1, 2.1.2 and 2.2.3. They describe the deformation and boundary conditions for the mechanical phase of the coupled problem, see Fig. 9.1a.

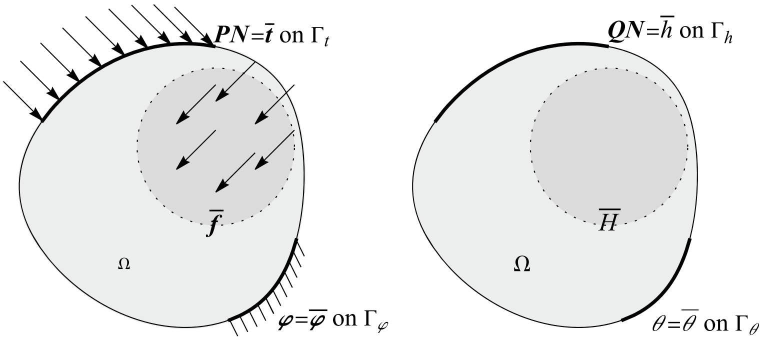
For the thermal problem, we define the absolute temperature field $\theta(\boldsymbol{X}, t) > 0$ in the reference configuration Ω of the solid along with the boundary conditions

$$\theta = \overline{\theta} \text{ on } \Gamma_T \quad \text{and } \boldsymbol{Q}\boldsymbol{N} = \bar{h} \text{ on } \Gamma_h \tag{9.1}$$

with the prescribed temperature field $\overline{\theta}$ and the heat flux $\bar{h}$, as shown in Fig. 9.1b. The Lagrangian heat flux vector $\boldsymbol{Q}$ is assumed to be governed by an isotropic Fourier-type law

$$\boldsymbol{Q} = -J\, K_\theta\, \boldsymbol{F}^{-1}\boldsymbol{F}^{-T}\nabla\theta \tag{9.2}$$

where $\nabla\theta$ is the material temperature gradient and K_θ is the thermal conductivity which must be positive ($K_\theta > 0$) in order to achieve thermodynamical consistency. In this section, the gradient operator is computed with respect to the coordinates in the initial configuration, i.e. $\nabla \equiv \nabla_{\boldsymbol{X}}$.


$\boldsymbol{PN}=\bar{\boldsymbol{t}}$ on Γ_t
$\bar{\boldsymbol{f}}$
Ω
$\boldsymbol{\varphi}=\overline{\boldsymbol{\varphi}}$ on Γ_φ
$\boldsymbol{QN}=\bar{h}$ on Γ_h
$\overline{H}$
Ω
$\theta=\overline{\theta}$ on Γ_θ

(a) deformation map (b) temperature field

Fig. 9.1 Solid with boundary conditions for the coupled problem

The continuum framework for the mechanical part uses as basic variables the displacement vector $\boldsymbol{u}$ while the thermal fields are described by the temperature θ. A multi-field setting of thermo-mechanical finite strain plasticity is then characterized by these two global primary fields which are combined in

$$\mathfrak{U} = \{\boldsymbol{u}, \theta\}\,. \tag{9.3}$$

The subsequent constitutive approach that models the thermo-mechanical coupled problem uses the set

$$\mathfrak{C} = \{\boldsymbol{F}(\boldsymbol{u}), \theta, \nabla\theta, \mathbf{h}\} \tag{9.4}$$

consisting of the deformation gradient $\boldsymbol{F}$, the temperature θ and its gradient $\nabla\theta$ and includes the history variables

$$\mathbf{h} = \{\boldsymbol{C}_p^{-1}, \alpha\}, \tag{9.5}$$

with the plastic strain $\boldsymbol{C}_p$ and the hardening variable α, see Sects. 2.2.3 and 8.1.

9.2.1 Energetic and Dissipative Response Functions

Following the contributions of Simo and Miehe (1992), Wriggers et al. (1992), Aldakheel and Miehe (2017) and Aldakheel (2017), the free energy function for the coupled thermo-plasticity problem at finite strains takes for a multiplicative split of the deformation gradient $\boldsymbol{F} = \boldsymbol{F}_e\,\boldsymbol{F}_p$ in the elastic (e) and plastic part (p) the form

$$\widehat{\Psi}(\mathfrak{C}) = \psi_{e,vol}(J_e) + \psi_{e,iso}(\bar{\boldsymbol{b}}_e) + \psi_{e,th}(J_e, \theta) + \psi_{th}(\theta) + \psi_p(\alpha, \theta) \tag{9.6}$$

in terms of the state variables introduced in (9.4). The volumetric part of the isotropic energetic response function is assume to have the form

$$\psi_{e,vol}(J_e) = \frac{\kappa}{4}(J_e^2 - 1 - 2\ln J_e) \tag{9.7}$$

with the elastic bulk modulus κ and the elastic part of the Jacobian, computed as $J_e = \sqrt{\det \boldsymbol{b}_e}$ where the left Cauchy Green tensor is defined as $\boldsymbol{b}_e = \boldsymbol{F}_e\,\boldsymbol{F}_e^T = \boldsymbol{F}\,\boldsymbol{C}_p^{-1}\,\boldsymbol{F}^T$. The isochoric elastic part of the free energy function is given by

$$\psi_{e,iso}(\bar{\boldsymbol{b}}_e) = \frac{\mu}{2}(\mathrm{tr}\,\bar{\boldsymbol{b}}_e - 3) \tag{9.8}$$

where $\bar{\boldsymbol{b}}_e = J_e^{-\frac{2}{3}}\boldsymbol{b}_e$ is the isochoric part of the elastic part of the left Cauchy–Green tensor, see also (2.74), and μ is the shear modulus.

Following the Coleman–Noll procedure, the Kirchhoff stress tensor $\boldsymbol{\tau}$ and the first Piola–Kirchhoff stress tensor $\boldsymbol{P}$ are obtained from the volumetric-isochoric elastic parts of the free energy function $\widehat{\Psi}(\mathfrak{C})$ for isotropic material behaviour

$$\boldsymbol{\tau} = 2\boldsymbol{b}_e \frac{\partial \widehat{\Psi}}{\partial \boldsymbol{b}_e} \qquad \text{and} \qquad \boldsymbol{P} = \boldsymbol{\tau} \boldsymbol{F}^{-T} \,. \tag{9.9}$$

The coupled thermoelastic part is defined as

$$\psi_{e,th}(J_e, \theta) = -3\alpha_\theta(\theta - \theta_0)\, \frac{\partial \psi_{e,vol}}{\partial J_e} \quad \text{with} \quad \frac{\partial \psi_{e,vol}}{\partial J_e} = \frac{\kappa}{2}(J_e - \frac{1}{J_e}) \tag{9.10}$$

where θ_0 is the reference temperature and α_θ is the thermal expansion coefficient. The pure thermal contribution of the free energy follows as

$$\psi_{th}(\theta) = c\left(\theta - \theta_0 - \theta \log \frac{\theta}{\theta_0}\right) \tag{9.11}$$

with the heat capacity c. The absolute temperature field is strictly positive ($\theta > 0$).

A formulation for the plastic potential can be found in Simo and Miehe (1992). It has the form

$$\psi_p(\alpha, \theta) = Y_0(\theta)\, \alpha + \frac{H(\theta)}{2}\alpha^2 + [Y_\infty(\theta) - Y_0(\theta)]\big[\alpha + \exp(-\delta\alpha)/\delta\big] \tag{9.12}$$

in terms of the temperature dependent material parameters $Y_0(\theta) > 0$, $Y_{\infty(\theta)} \geq Y_0(\theta)$ and $H(\theta) \geq 0$. These can be defined as

$$\begin{aligned} Y_0(\theta) \; &= \; Y_0\big[1 - w_0(\theta - \theta_0)\big]\,, \\ Y_\infty(\theta) &= Y_\infty\big[1 - w_h(\theta - \theta_0)\big]\,, \\ H(\theta) \; &= \; H\big[1 - w_h(\theta - \theta_0)\big] \end{aligned} \tag{9.13}$$

where w_h is the hardening/softening parameter, w_0 is the softening parameter of the flow stress and δ is the saturation parameter.

The yield function of von Mises-type finite thermo-plasticity $f(\boldsymbol{s}\,, \alpha\,, \theta)$ restricts the elastic region and has the form

$$f(\boldsymbol{s}\,, \alpha\,, \theta) = \sqrt{\frac{3}{2}}\; \|\boldsymbol{s}\| - \sigma_y(\alpha, \theta) \tag{9.14}$$

with the deviator of the Kirchhoff stress $\boldsymbol{s}$ and the yield limit σ_y

$$\boldsymbol{s} = \text{dev}\boldsymbol{\tau} = \boldsymbol{\tau} - \frac{1}{3}\text{tr}\, \boldsymbol{\tau}\, \mathbf{1} \quad \text{and} \quad \sigma_y(\alpha, \theta)\,. \tag{9.15}$$

Rate dependent plastic deformations can be described by a Perzyna-type model with the dual dissipation function for visco-plasticity

$$\Phi^*(\boldsymbol{s}, \sigma_y) = \frac{1}{2\eta_p}\Big\langle f \Big\rangle^2 \tag{9.16}$$

where the yield function $f(\boldsymbol{s}\,, \alpha\,, \theta)$ is used. Note that $\langle f \rangle = \frac{1}{2}(f + |\, f\,|)$ is the Macauley bracket which only yields positive values of f. This formulations needs an additional parameter, the viscosity η_p.

For completeness we recall the evolution equations for the plastic variables, see (2.80),

$$\dot{\boldsymbol{C}}_p^{-1} = -2\,\dot{\gamma}\,\boldsymbol{F}^{-1}\,\boldsymbol{n}\,\boldsymbol{F}\,\boldsymbol{C}_p^{-1} \quad \text{and } \dot{\alpha} = \dot{\gamma} = \frac{1}{\eta_p}\langle\, f\,\rangle\,, \tag{9.17}$$

with the flow direction

$$\boldsymbol{n} = \frac{\partial f}{\partial \boldsymbol{s}}\,. \tag{9.18}$$

This set of equations will be used later for the algorithmic treatment of thermo-plasticity.

The discretized form of the evolution Eq. (9.17) follows from Simo and Miehe (1992) and Korelc and Stupkiewicz (2014). Together with the Karush–Kuhn–Tucker conditions (2.81) the evolution (9.17) forms the local residual of the problem within the time stepping scheme, see Algorithm 1,

$$\boldsymbol{F}\boldsymbol{C}_p^{-1} - \exp[-2(\alpha - \alpha_n)\,\boldsymbol{n}]\,\boldsymbol{F}\,\boldsymbol{C}_{p,n}^{-1} = 0 \quad \text{and} \quad f(\boldsymbol{s}\,, \alpha\,, \theta) = 0\,, \tag{9.19}$$

where $\boldsymbol{C}_{p,n}^{-1}$ and α_n are the converged history values at the previous step. The system of Eq. (9.19) has to be solved locally at the element level only when $f > 0$, see Algorithm 1 in Sect. 8.1.2. For further details on the algorithmic treatment of (9.19), which includes the plastic history field array $\mathbf{h}$, we refer to Simo and Miehe (1992) and Wriggers (2008).

9.2.2 Global Constitutive Equations

The global governing equations describing the coupled problem are the balance of linear momentum, see (2.19), and the absolute temperature evolution, see for the stationary part (2.41),

$$\begin{aligned} \text{Div}[\boldsymbol{P}] + \bar{\boldsymbol{f}} &= \boldsymbol{0} \\ c\,\dot{\theta} + \text{Div}[\boldsymbol{Q}] - \mathcal{D}_{loc} + \mathcal{H} - R &= 0 \end{aligned} \tag{9.20}$$

along with the Neumann-type boundary conditions in (2.20), (2.21) and (9.1), the prescribed body force $\bar{\boldsymbol{f}}$ and heat source R. The reduced local dissipation density function is given as

$$\mathcal{D}_{loc} = \zeta \ \sigma_y(\alpha, \theta) \ \dot{\alpha} \tag{9.21}$$

with the constant dissipation factor $\zeta \in [0, 1]$, see e.g. Zdebel and Lehmann (1987). The heat capacity c and the latent heating $\mathcal{H}$ in (9.20) are defined as

$$c = -\theta \partial^2_{\theta\theta} \widehat{\Psi}(\mathfrak{C}) \qquad \text{and} \quad \mathcal{H} = -\theta \partial_\theta \Big[\boldsymbol{P} \cdot \dot{\boldsymbol{F}} - \mathcal{D}_{loc} \Big] . \tag{9.22}$$

9.2.3 Weak form and Pseudo-Potential Energy Function

Based on the set of constitutive equations, the coupled thermo-mechanical problem can be formulated. Classically, the weak form of Eq. (9.20) is the starting point for the development of a discretization method. In a thermo-mechanical analysis the weak form is decomposed into mechanical M and thermal θ parts

$$G(\boldsymbol{u}, \theta) = G_M(\boldsymbol{u}, \theta) + G_\theta(\boldsymbol{u}, \theta) \tag{9.23}$$

where the first part represents the weak form of the linear momentum, first equation in (9.20),

$$G_M(\boldsymbol{u}, \theta) = \int\limits_\Omega \Big[\boldsymbol{P}(\boldsymbol{u}, \mathbf{h}) \cdot \boldsymbol{F}(\boldsymbol{v}) - \bar{\boldsymbol{f}} \cdot \boldsymbol{v} \Big] \, d\Omega - \int\limits_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \boldsymbol{v} \, d\Gamma \tag{9.24}$$

and the second part is the weak form of the temperature evolution, second equation in (9.20),

$$G_\theta(\theta, \boldsymbol{u}) = \int\limits_\Omega \vartheta \left[\frac{c}{\Delta t}(\theta - \theta_n) + \mathcal{H} - \mathcal{D}_{loc} - R \right] d\Omega - \int\limits_\Omega \nabla\vartheta \cdot \boldsymbol{Q} \, d\Omega - \int\limits_{\Gamma_h} \bar{h}\vartheta \, d\Gamma \tag{9.25}$$

where the time discretization $\dot{\alpha} = \frac{\alpha - \alpha_n}{\Delta t}$ is used in $\mathcal{D}_{loc}$ defined in (9.21).

In these weak forms the test functions for the displacement and the temperature fields are given by $\boldsymbol{v}, \vartheta$, respectively. Note that weak form (9.24) depends on the temperature through the temperature depending material parameters, see (9.12). The weak form (9.25) depends on the contrary on the displacement field through the dissipation, see (9.21), and the latent heat, see (9.22).

When using automatic differentiation tools, like the software tool *AceGen,* see Korelc and Wriggers (2016), it can be computationally more efficient to base the development of the thermo-elasto-plastic formulations on a pseudo-potential energy function instead of using the weak form (9.23). The pseudo-potential energy depends only on the thermo-elastic parts and keeps the plastic history field array $\mathbf{h}$ constant during the first variation, additionally certain contributions in the mechanical and thermal parts related to the temperature have to be kept constant as well. The pseudo potential can be formulated as

$$U(\boldsymbol{u},\theta;\mathbf{h}) = \int\limits_{\Omega} \Big[\Psi(\mathfrak{C};\mathbf{h}) - \bar{\boldsymbol{f}}\cdot\boldsymbol{u} \; - R\,\theta\Big]\, \mathrm{d}\Omega - \int\limits_{\Gamma_N} \bar{\boldsymbol{t}}\cdot\boldsymbol{u}\,\mathrm{d}\Gamma - \int\limits_{\Gamma_h} \bar{h}\,\theta\,\mathrm{d}\Gamma \tag{9.26}$$

with the thermo-mechanical energy function

$$\Psi(\mathfrak{C};\mathbf{h}) = \Psi_M + \Psi_\theta\,. \tag{9.27}$$

In detail, the two parts are given by

$$\begin{aligned} \Psi_M &= \;\; \psi_{e,vol} + \psi_{e,iso} - \big[3\alpha_\theta(\theta-\theta_0)\big]_{ct}\,\partial_{J_e}\psi_{e,vol} \\ \Psi_\theta &= \tfrac{c}{2\Delta t}(\theta-\theta_n)^2 + \theta\,\big[\mathcal{H}-\mathcal{D}_{loc}\big]_{ct} - \nabla\theta\cdot\big[\boldsymbol{Q}\big]_{ct}\,. \end{aligned} \tag{9.28}$$

The components inside the square bracket $[-]_{ct}$ have to be kept constant during the variation of potential in the automatic differentiation procedure. For more information about the construction of such a pseudo-potential, we refer the interested reader to Korelc and Wriggers (2016). The variation of the pseudo potential (9.26) with respect to the displacement and temperature field, while keeping the mentioned terms $[-]_{ct}$ constant, leads to the weak forms (9.24) and (9.25).

The resulting global equation set depends on the displacement and the absolute temperature. It can be solved either by a monolithic or by a staggered algorithm. Classically, due to the complexity of the linearization for a coupled formulation the staggered scheme was employed, see e.g. Simo and Miehe (1992), Cervera et al. (1999) and Martins et al. (2017). By using the software tool *AceGen,* see Korelc and Wriggers (2016), a robust and efficient monolithic scheme can be developed and implemented. The latter computational approach is used for the numerical simulations discussed in the chapter.

9.3 Virtual Element Discretization

The low order formulation, used in Sects. 8.1.1 and 8.1.3 for finite-strain plasticity, has now to be amended such that coupled problems can be treated. For that the ansatz space is enlarged by the temperature field. Again the virtual element method relies on the split of the ansatz space into a part $\mathfrak{U}_\pi$ representing the projected primary fields defined in (9.3) and a remainder

$$\mathfrak{U}_h = \mathfrak{U}_\pi + (\mathfrak{U}_h - \mathfrak{U}_\pi) \quad \text{with } \mathfrak{U}_\pi = \{\boldsymbol{u}_\pi, \theta_\pi\}\,. \tag{9.29}$$

The simplest form of a low order virtual element is provided by a linear ansatz, see (3.130) for the pure mechanical case,

$$\mathfrak{U}_\pi = \begin{Bmatrix} u_{\pi\, x} \\ u_{\pi\, y} \\ u_{\pi\, z} \\ \theta_\pi \end{Bmatrix} = \mathbf{a} \cdot \mathbf{N}_\pi = \begin{bmatrix} a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \\ a_4 & a_8 & a_{12} & a_{16} \end{bmatrix} \begin{Bmatrix} 1 \\ X \\ Y \\ Z \end{Bmatrix} \tag{9.30}$$

with the sixteen unknowns $\mathbf{a}$ which have to be determined in terms of the nodal degrees of freedom. Contrary to the form of the ansatz used in (3.130) equation (9.30) has a very compact form, avoiding unnecessary zeros.

Here the same procedures as in Sect. 3.2.3, see *Remark 3.5*, and Sect. 4.2 will be applied jointly to define the ansatz functions $\mathfrak{U}_\pi$ for the coupled virtual element. In case of a linear polynomial ansatz function, the simplest and most efficient way is to use (3.137) and (3.140) to compute the projection. This procedure is based on tensor notation and determines the projection $\mathfrak{U}_\pi$ for a linear ansatz function for a virtual element Ω_v, see Sect. 3.1.5,

$$\nabla \mathfrak{U}_\pi = \frac{1}{\Omega_v} \int\limits_{\Omega_v} \nabla\, \mathfrak{U}_h \; d\Omega = \frac{1}{\Omega_v} \int\limits_{\Gamma_v} \mathfrak{U}_h \otimes \boldsymbol{N} \;\; \Gamma = \mathbb{D}_\nabla^{(3,1)}\, \mathfrak{U}_v \, . \tag{9.31}$$

Here $\boldsymbol{N}$ is the normal at the face Γ_f of the domain Ω_v. Furthermore, $\mathbb{D}_\nabla^{(3,1)}$, see e.g. (3.152), is an operator that links the projected gradient to the nodal degrees of freedom $\mathfrak{U}_v$.[1] The set of nodal degrees of freedom of a virtual element Ω_v are defined as

$$\mathfrak{U}_v = \left\langle \mathfrak{U}_1 \, \mathfrak{U}_2 \, \ldots \mathfrak{U}_{n_V} \right\rangle^T \tag{9.32}$$

with the nodal degrees of freedom at node K given by $\mathfrak{U}_K = \left\{ u_{Kx} \, , u_{Ky} \, , u_{Kz} \, , \theta_K \right\}^T$.

The projection in (9.31) does not determine the ansatz $\mathfrak{U}_\pi$ in (9.30) completely. Thus a further condition has to be employed to obtain all constants $\mathbf{a}$. For this purpose we adopt the condition that equalizes the sum of the nodal values of $\mathfrak{U}_h$ and the nodal values of its projection $\mathfrak{U}_\pi$. This condition yields for each element Ω_v

$$\sum_{K=1}^{n_V} \mathfrak{U}_\pi(\mathbf{X}_K) = \sum_{K=1}^{n_V} \mathfrak{U}_h(\mathbf{X}_K) \, , \tag{9.33}$$

where $\mathbf{X}_K$ are the coordinates of the nodal point K and the sum includes all n_V boundary nodes. Using (3.140) it follows

[1] The operator $\mathbb{D}_\nabla^{(3,1)}$ does not have to be computed explicitly when using automatic software generation as in Sect. 3.1.5.

$$\begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} = \sum_{K=1}^{n_V} [\mathfrak{U}_K - \nabla \mathfrak{U}_\pi \, \mathbf{X}_K] \tag{9.34}$$

The two conditions (9.31) and (9.34) yield all unknowns $\mathbf{a}$. Thus the projection $\mathfrak{U}_\pi$ can be formulated in terms of the nodal values

$$\mathfrak{U}_\pi = \mathbf{H}^{(3,1)}(X, Y) \, \mathbb{D}_{\mathfrak{U}}^{(3,1)} \, \mathfrak{U}_v \, . \tag{9.35}$$

Based on the split in (9.29), the pseudo-potential function defined in (9.26) can be rewritten by summing up all element contributions for the n_v virtual elements as

$$U(\mathfrak{U}_h, \mathbf{h}) = \mathop{\mathsf{A}}_{v=1}^{n_v} U(\mathfrak{U}_h, \mathbf{h}_v) \tag{9.36}$$

with

$$U(\mathfrak{U}_h, \mathbf{h}_v) = U_c(\mathfrak{U}_\pi, \mathbf{h}_v) + U_{stab}(\mathfrak{U}_h - \mathfrak{U}_\pi, \mathbf{h}_v) \, , \tag{9.37}$$

with the consistency part U_c and an associated stabilization term U_{stab}. In this discretization scheme, the plastic history variables are determined based on the consistency term, see e.g. Wriggers and Hudobivnik (2017) and Hudobivnik et al. (2018). These variables are then used in the stabilization procedure and are frozen during the evaluation of the stabilization term.

The pseudo-energy part related to Ψ in (9.26) can be computed for the coupled problem by neglecting the contributions of the body forces and tractions (these were already provided in Chap. 6). It yields for a virtual element Ω_v

$$U_c(\mathfrak{U}_\pi, \mathbf{h}) = \int_{\Omega_v} \Psi(\mathfrak{C}_\pi) \, d\Omega = \Psi(\mathfrak{C}_\pi) \, \Omega_v \quad \text{with } \mathfrak{C}_\pi = \{\nabla \boldsymbol{u}_\pi, \theta_\pi, \nabla \theta_\pi, \mathbf{h}_v\} \tag{9.38}$$

where the primary fields $\mathfrak{U}_\pi$ are linear functions and their gradient $\nabla \mathfrak{U}_\pi$ is constant over the domain of the virtual element Ω_v. As a consequence, the pseudo-energy Ψ is integrated by evaluating the function at the barycenter of the element $\mathbf{X}_c$, see Fig. 9.2, and by multiplying it with the element volume Ω_v. This procedure is analogous to a Gauss integration with one point. The history field array $\mathbf{h}_v$ in (9.38) contains local variables. These variables are updated locally at element level within the nested Newton–Raphson procedure using the Algorithm 1 provided in Sect. 8.1.2.

The energy stabilization of the virtual element formulation is also introduced for coupled problems like it was done for finite-strain plasticity in Sect. 8.1.3. Here a stabilization potential has to be constructed for the coupled problem. First the virtual element domain Ω_v is subdivided into an internal triangle (2D) / tetrahedral (3D) mesh as depicted in Fig. 9.2. Then the potential is evaluated at the centroid $\boldsymbol{X}_c^i$ of triangle i using a linear ansatz for the primary fields $\mathfrak{U}_h$, see Wriggers et al.

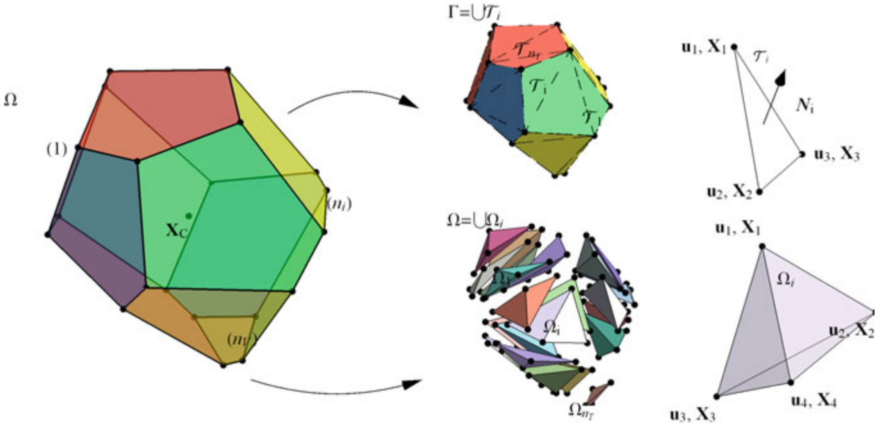


Fig. 9.2 Virtual element ansatz. Internal tetrahedral mesh and virtual element faces split into multiple triangles

(2017) and Hudobivnik et al. (2018). To obtain the triangularization of the virtual element, specific triangulation algorithms are used that are provided by the commands `ToElementMesh` or `TriangulateMesh` in *Mathematica*. These are very robust, furthermore the quality of the created triangulation can be controlled by the user.

Within the stabilization potential the contributions of the mechanical and thermal parts are summed up for one virtual element Ω_v. The assembly of all elements yields the discretized form of the stabilization potential, based on the pseudo energy functional (9.27),

$$U_{stab}(\mathfrak{U}_h - \mathfrak{U}_\pi, \mathbf{h}_v) = \sum_{s\in\{M,\theta\}} \beta_s \left(\sum_{i=1}^{n_T} \Omega_v^i \; \Psi_s(\mathfrak{C})|_c - \Psi_s(\mathfrak{C}_\pi)|_c \; \Omega_v \right) \tag{9.39}$$

with Ω_v^i being the area of the i^{th} triangle/tetrahedron in the element e for the two- respectively three-dimensional case. The number of internal triangles/tetrahedra is n_T. The potential $\Psi_s(\mathfrak{C})|_c$ is evaluated by applying a one point integration for each internal triangle.

For the coupled problem, two stabilization parameters are introduced

$$\boldsymbol{\beta} = \left\{ \beta_M, \beta_\theta \right\}, \tag{9.40}$$

one for the mechanical (elastic-plastic) part (M) and the other for thermal part (θ). For the thermal problem, we propose a constant value for β_θ which has to be chosen from the interval

$$0 < \beta_\theta \leq 1 \,. \tag{9.41}$$

As in the pure mechanical case, the thermal part of the potential $U(\mathfrak{U}, \mathbf{h})$ in (9.36) will depend only on the projection part $U_c(\mathfrak{U}_\pi, \mathbf{h})$ for $\beta_\theta \to 0$, leading to rank deficiency. When $\beta_\theta \to 1$ a pure FEM solution related to the internal mesh will be reproduced. An optimal ratio extrapolated from various numerical examples lies in the range $\beta_\theta \in [0.2, 0.6]$ see e.g. Hudobivnik et al. (2018) and Aldakheel et al. (2019). In the numerical example $\beta_\theta = 0.4$ is selected for all simulations. In case of a pure elastic-plastic state, the stabilization parameter β_M derives from the same procedure introduced in Wriggers and Hudobivnik (2017) leading to

$$\beta_M = \min\left\{0.4\,,\, \frac{\sigma_{VM}}{E\,\alpha}\right\}\;, \tag{9.42}$$

where $\sigma_{VM} = \sqrt{3/2}\,\|\boldsymbol{s}\|$ is the von Mises stress, E is the Young's modulus and α is the equivalent plastic strain providing an approximation for the tangent of the hardening curve.

All further derivations were performed with the software tool *AceGen*. This yields the residual vector $\mathbf{R}_v$ and the consistent tangent matrix $\mathbf{K}_{Tv}$ of the virtual element Ω_v. Based on (9.36) along with the pseudo potentials (9.38) and (9.39) we obtain

$$\mathbf{R}_v = \left.\frac{\partial U(\mathfrak{U}_v; \mathbf{h}_v)}{\partial \mathfrak{U}_v}\right|_{\{\mathbf{h}_v, \mathcal{D}_{coc}, \mathcal{H}\}=const} \quad \text{and } \mathbf{K}_{Tv} = \frac{\partial \mathbf{R}_v}{\partial \mathfrak{U}_v}\,. \tag{9.43}$$

As in the case of finite plasticity, the history variables stemming from Algorithm 1 are treated as fixed fields in (9.43) during the first variation, i.e. $\partial_{\mathfrak{U}_v} \mathbf{h}_v = \mathbf{0}$. Also the dissipation density $\mathcal{D}_{coc}$ and the latent heat $\mathcal{H}$ in (9.28) have to be kept constant. Element residuals and tangent matrices are assembled to the global system of nonlinear equations which then is solved with a Newton–Raphson scheme to obtain the global primary field $\mathfrak{U}$.

9.4 Representative Numerical Example

The performance of the proposed virtual element formulation for modeling finite strain thermo-elasto-plastic problems will be demonstrated by means of a representative numerical example. For comparison purposes, results of standard finite elements (FEM) are also included. The material parameters for metals, used in the simulations, are obtained from Hallquist (1984), Simo (1988a) and Simo and Miehe (1992) and outlined in Table 9.1.

Different mesh types are introduced for the computations using the virtual element method, see Appendix B. Regular meshes are employed, we use Q1 and Q2S for this two-dimensional problem. Furthermore Voronoi meshes with random size distribution, VOR, are applied. In the examples solutions obtained with the virtual element method are denoted by VE-XX where XX stands for one of the mesh types defined

Table 9.1 Material parameters used for the numerical examples

Parameter	Symbol	Value	Unit
Young's modulus	E	206.9	kN/mm^2
Poisson ratio	ν	0.29	–
Initial yield stress	Y_0	0.45	kN/mm^2
Infinite yield stress	Y_∞	1.165	kN/mm^2
Hardening coefficient	H	0.129	kN/mm^2
Saturation exponent	δ	16.93	–
Flow stress softening	ω_0	0.002	1/K
Hardening softening	ω_h	0.002	1/K
Dissipation factor	ζ	0.9	–
Thermal expansion coefficient	α_T	0.000012	1/K
Thermal conductivity	K_θ	0.045	kN/(s K)
Heat capacity	c	0.003588	kN/(mm^2 K)

above. All meshes represent a first order virtual element discretization, however with different number of nodes and geometry.

In order to test the robustness of VEM, several finite element formulations were selected for comparison:

- FE-Q1/H1 denotes a standard first order quadrilateral/Hexahedral finite element with linear interpolation for 2D/3D,
- FE-Q1P0 denotes a Hu-Washizu mixed finite element with two additional element degrees of freedom: the pressure p and the volume dilatation Θ. This yields a mixed finite element which does not lock in case of plastic incompressibility, see Simo et al. (1985) and Wriggers (2008).
- FE-CG4/CG9 denotes a modified enhanced assumed strain, representing a standard first order finite element with linear interpolation of the primary field and additionally a modified displacement gradient with internal 4/9 enhanced modes leading to $\boldsymbol{F} = \boldsymbol{1} + \nabla \boldsymbol{u} + \widetilde{\boldsymbol{H}}$. The enhanced displacement gradient $\widetilde{\boldsymbol{H}}$ is condensed out at element level, see Wriggers and Reese (1996), Wriggers and Korelc (1996) and Korelc et al. (2010).

The applicability of the virtual element method to a thermo-mechanical problem is illustrated using a steel bolt which is subjected to thermal and mechanical loading. The underlying forming process was first considered in Argyris and Doltsinis (1981) using finite element discretizations.

The thermo-mechanical response of the bolt is investigated for the two-dimensional case, for an extension to three dimensions see Aldakheel et al. (2019). The geometry of the specimen is depicted together with the Neumann boundary conditions in Fig. 9.3a. The size of the bolt is chosen to be: $H = 50$ mm and $L = 20$ mm.

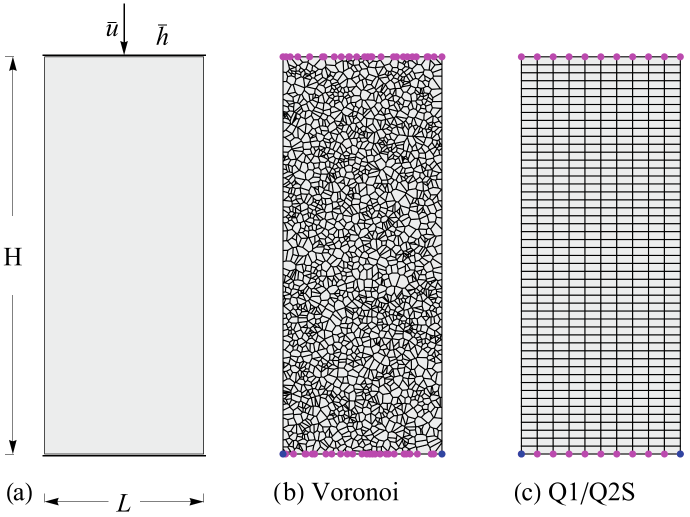
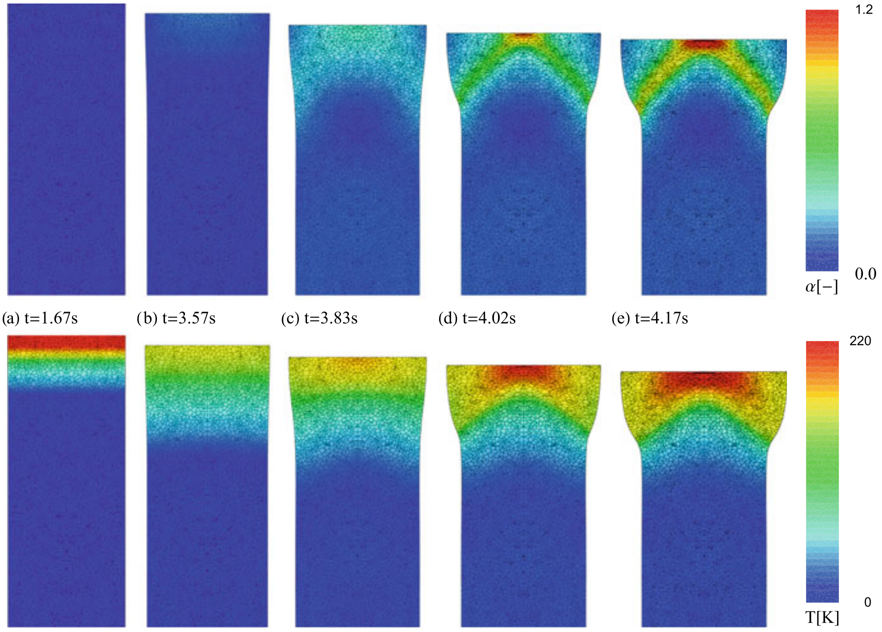


Fig. 9.3 Forming of a steel bolt—**a** Geometry and boundary conditions, **b** Voronoi mesh and **c** regular Q1/Q2S mesh

The simulation is performed under plane strain conditions. The vertical displacement is constrained in the lower and upper parts of the specimen, as depicted in Fig. 9.3a. A heat flux of $\bar{h} = 5$ kN/(mm s) is prescribed for $0 \le t < 1.667$ s at the top face. Then the heat flux is stopped ($\bar{h} = 0$) for 1.667 s $\le t < 3.333$ s, thus the specimen "rests" without any loading for 1.667 s. After this time period a vertical displacement with magnitude of $\bar{u} = 0.125H$ is applied at the upper part of the specimen for 3.333 s $\le t \le 4.167$ s.

The evolutions of the absolute temperature field θ and the equivalent plastic strain α are shown in Fig. 9.4 for different states of the forming process. The first applied thermal loading results in a thermo-elastic deformation and increase of the temperature, as depicted in Fig. 9.4a and f. At this stage, no plastic deformation occurs. Plasticity is initiated once the applied load reaches a certain threshold, see Fig. 9.4c. A cross shear localization starts to form later, as depicted in Fig. 9.4d. Thereafter thermo-plastic deformation continues to evolve until the final stage of the deformation in the bolt, see Fig. 9.4e and j. The corresponding thermo-mechanical load versus time curves for different element formulations are shown in Fig. 9.5. All virtual element meshes, VE-Q1, VE-Q2S and VE-VO yield the same result as the mixed FE-Q1P0 and CG4 finite elements. Figure 9.6 illustrates a good agreement between all element solutions—besides the locking FE-Q1 element—for this forming process.

The example demonstrates that the virtual element solutions have the same accuracy as specially designed mixed finite elements of the same order. It is interesting to note that a discretization based on virtual elements does not need substepping



(a) t=1.67s (b) t=3.57s (c) t=3.83s (d) t=4.02s (e) t=4.17s

(f) t=1.67s (g) t=3.57s (h) t=3.83s (i) t=4.02s (j) t=4.17s

Fig. 9.4 Forming of a steel bolt. Contour plots of the equivalent plastic strain in (**a**)–(**e**) and the absolute temperature field in (**f**)–(**j**) at different deformation states using VE-Voronoi

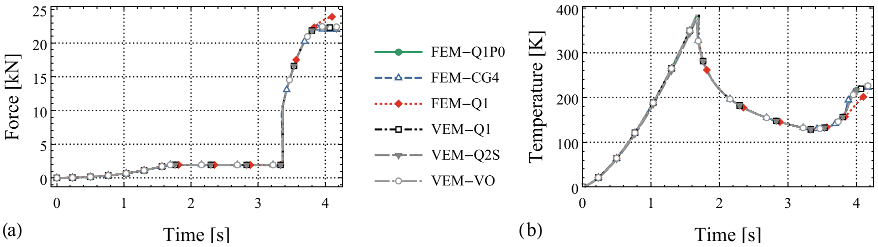


Fig. 9.5 Forming of a steel bolt. **a** Force-time and **b** temperature-time curves for different types of elements

underlining the robustness of this discretization scheme. As a result, larger load increments can be applied which makes the virtual element method more efficient when compared to mixed first order finite elements, especially in case of severe element distortions.

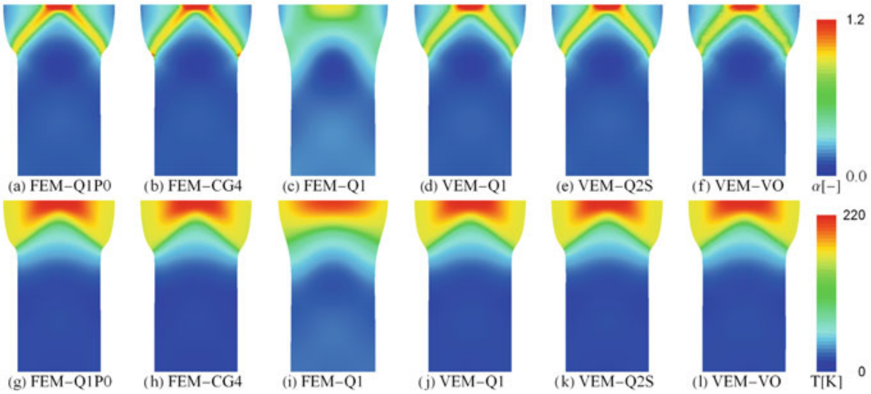


Fig. 9.6 Forming of a steel bolt. Distribution of the equivalent plastic strain α and the absolute temperature field T at the final deformation state for all elements

References

Aldakheel, F. 2017. Micromorphic approach for gradient-extended thermo-elastic-plastic solids in the logarithmic strain space. *Continuum Mechanics and Thermodynamics* 29 (6): 1207–1217.

Aldakheel, F., and C. Miehe. 2017. Coupled thermomechanical response of gradient plasticity. *International Journal of Plasticity* 91: 1–24.

Aldakheel, F., B. Hudobivnik, and P. Wriggers. 2019. Virtual elements for finite thermo-plasticity problems. *Computational Mechanics* 64: 1347–1360.

Anand, L., N.M. Ames, V. Srivastava, and S.A. Chester. 2009. A thermo-mechanically coupled theory for large deformations of amorphous polymers. Part i: Formulation. *International Journal of Plasticity* 25: 1474–1494.

Antonietti, P.F., G. Vacca, and M. Verani. 2022. Virtual element method for the Navier-Stokes equation coupled with the heat equation. arXiv:2205.00954

Argyris, J.H., and J.S. Doltsinis. 1981. On the natural formulation and analysis of large deformation coupled thermomechanical problems. *Computer Methods in Applied Mechanics and Engineering* 25: 195–253.

Bartels, A., T. Bartel, M. Canadija, and J. Mosler. 2015. On the thermomechanical coupling in dissipative materials: A variational approach for generalized standard materials. *Journal of the Mechanics and Physics of Solids* 82: 218–234.

Beirão da Veiga, L., A. Pichler, and G. Vacca. 2021. A virtual element method for the miscible displacement of incompressible fluids in porous media. *Computer Methods in Applied Mechanics and Engineering* 375: 113649.

Berrone, S., and M. Busetto. 2022. A virtual element method for the two-phase flow of immiscible fluids in porous media. *Computational Geosciences* 26 (1): 195–216.

Böhm, C., B. Hudobivnik, M. Marino, and P. Wriggers. 2021. Electro-magneto-mechanically response of polycrystalline materials: Computational homogenization via the virtual element method. *Computer Methods in Applied Mechanics and Engineering* 375: 113775.

Čanađija, M., and J. Mosler. 2011. On the thermomechanical coupling in finite strain plasticity theory with non-linear kinematic hardening by means of incremental energy minimization. *International Journal of Solids and Structures* 48: 1120–1129.

Cereceda, D., M. Diehl, F. Roters, D. Raabe, J.M. Perlado, and J. Marian. 2016. Unraveling the temperature dependence of the yield strength in single-crystal tungsten using atomistically-informed crystal plasticity calculations. *International Journal of Plasticity* 78: 242–265.

Cervera, M., C. Agelet De Saracibar, and M. Chiumenti. 1999. Thermo-mechanical analysis of industrial solidification processes. *International Journal for Numerical Methods in Engineering* 46 (9): 1575–1591.

Chapuis, A., and J.H. Driver. 2011. Temperature dependency of slip and twinning in plane strain compressed magnesium single crystals. *Acta Materialia* 59 (5): 1986–1994.

Cottrell, J.A., T.J.R. Hughes, and Y. Bazilevs. 2009. *Isogeometric analysis: Toward integration of CAD and FEA*. New York: Wiley.

Dhanush, V., and S. Natarajan. 2019. Implementation of the virtual element method for coupled thermo-elasticity in abaqus. *Numerical Algorithms* 80 (3): 1037–1058.

Hallquist, J.O. 1984. Nike 2d: An implicit, finite deformation, finite element code for analyzing the static and dynamic response of two-dimensional solids. Technical Report Rept. UCRL-52678, Lawrence Livermore National Laboratory, University of California, Livermore, CA.

Hudobivnik, B., F. Aldakheel, and P. Wriggers. 2018. Low order 3d virtual element formulation for finite elasto-plastic deformations. *Computational Mechanics* 63: 253–269.

Hughes, T.J.R., J.A. Cottrell, and Y. Bazilevs. 2005. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 194 (39–41): 4135–4195.

Korelc, J., and S. Stupkiewicz. 2014. Closed-form matrix exponential and its application in finite-strain plasticity. *International Journal for Numerical Methods in Engineering* 98: 960–987.

Korelc, J., and P. Wriggers. 2016. *Automation of finite element methods*. Berlin: Springer.

Korelc, J., U. Solinc, and P. Wriggers. 2010. An improved EAS brick element for finite deformation. *Computational Mechanics* 46: 641–659.

Lion, A. 2000. Constitutive modelling in finite thermoviscoplasticity: A physical approach based on nonlinear rheological models. *International Journal of Plasticity* 16 (5): 469–494.

Liu, X., R. Li, and Z. Chen. 2019. A virtual element method for the coupled stokes-darcy problem with the beaver-joseph-saffman interface condition. *Calcolo* 56 (4): 1–28.

Martins, J., D. Neto, J. Alves, M. Oliveira, H. Laurent, A. Andrade-Campos, and L. Menezes. 2017. A new staggered algorithm for thermomechanical coupled problems. *International Journal of Solids and Structures* 122: 42–58.

Miehe, C., J. Méndez Diez, S. Göktepe, and L. Schänzel. 2011. Coupled thermoviscoplasticity of glassy polymers in the logarithmic strain space based on the free volume theory. *International Journal of Solids and Structures* 48: 1799–1817.

Simo, J.C. 1988a. A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition: Part i. Continuum formulation. *Computer methods in applied mechanics and engineering* 66 (2): 199–219.

Simo, J.C. 1998b. Numerical analysis and simulation of plasticity. In *Handbook of numerical analysis*, vol. 6, ed. P.G. Ciarlet and J.L. Lions, 179–499. North-Holland.

Simo, J.C., and C. Miehe. 1992. Associative coupled thermoplasticity at finite strains: Formulation, numerical analysis and implementation. *Computer Methods in Applied Mechanics and Engineering* 98: 41–104.

Simo, J.C., R.L. Taylor, and K.S. Pister. 1985. Variational and projection methods for the volume constraint in finite deformation elasto-plasticity. *Computer Methods in Applied Mechanics and Engineering* 51: 177–208.

Stainier, L., and M. Ortiz. 2010. Study and validation of thermomechanical coupling in finite strain visco-plasticity. *International Journal of Solids and Structures* 47: 704–715.

Stainier, L., A. Cuitino, and M. Ortiz. 2002. A micromechanical model of hardening, rate sensitivity and thermal softening in bcc single crystals. *Journal of the Mechanics and Physics of Solids* 50 (7): 1511–1545.

Wriggers, P., C. Miehe, M. Kleiber, and J. Simo. 1989. *On the thermomechanical treatment of necking problems a, finite element solution*. In *Proceedings of COMPLAS II*, ed. D.R.J. Owen, E. Hinton and E. Onate. Swansea: Pineridge Press.

Wriggers, P. 2008. *Nonlinear finite elements*. Berlin, Heidelberg, New York: Springer.

Wriggers, P., and B. Hudobivnik. 2017. A low order virtual element formulation for finite elasto-plastic deformations. *Computer Methods in Applied Mechanics and Engineering* 327: 459–477.

Wriggers, P., and J. Korelc. 1996. On enhanced strain methods for small and finite deformations of solids. *Computational Mechanics* 18: 413–428.

Wriggers, P., and S. Reese. 1996. A note on enhanced strain methods for large deformations. *Computer Methods in Applied Mechanics and Engineering* 135: 201–209.

Wriggers, P., C. Miehe, M. Kleiber, and J. Simo. 1992. A thermomechanical approach to the necking problem. *International Journal for Numerical Methods in Engineering* 33: 869–883.

Wriggers, P., B. Reddy, W. Rust, and B. Hudobivnik. 2017. Efficient virtual element formulations for compressible and incompressible finite deformations. *Computational Mechanics* 60: 253–268.

Yang, Q., L. Stainier, and M. Ortiz. 2006. A variational formulation of the coupled thermo-mechanical boundary-value problem for general dissipative solids. *Journal of the Mechanics and Physics of Solids* 54: 401–424.

Zdebel, U., and T. Lehmann. 1987. Some theoretical considerations and experimental investigations on a constitutive law in thermoplasticity. *International Journal of Plasticity* 3 (4): 369–389.

Zienkiewicz, O.C., and A.H.C. Chan. 1989. Coupled problems and their numerical solution. In *Advances in computational nonlinear mechanics*, ed. I.S. Doltsinis, 139–176. Springer Vienna.

Zienkiewicz, O.C., and R.L. Taylor. 2000. *The finite element method*, vol. 1, 5th ed. Oxford, UK: Butterworth-Heinemann.

Chapter 10
Virtual Elements for Fracture Processes

Fracture processes occur in different environments and define the lifespan of many engineering structures. To illustrate the effects of fracture in more detail one can think of its negative implications like failure of large structures, e.g. bridges, ships and vessels. Further instances are shattered glass, a broken leg, a ruptured aorta, a torn sail, a broken car part, a cracked beam among many others. Fracture processes can also be on purpose and useful as in chip forming during cutting, tearing open of packages along precut lines and breaking off of chocolate pieces.

It is common to differentiate between brittle and ductile fracture. Furthermore, materials which are subjected to time-varying loads may fail at a stress level that is significantly lower than yield stress or ultimate strength. This phenomenon is known as fatigue failure which can cause fatal failure of a component. The engineering analysis of such processes benefits significantly from a precisely predictive computational tool that models fracture response during the design and manufacturing processes of products. This is especially true, when structural reliability has to be improved. Hereby, different fracture parameters determine the fatigue life of products under cyclic loading conditions, e.g. stress intensity factors, flaw sizes and fracture toughness. In total, the analysis of failure due to fracture plays a significant role for the safety of various engineering structures, but also for medical devices, in biomechanics and many more applications related to our daily life.

An efficient low order virtual element method for damage mechanics and crack-propagation in solids with elastic as well as elasto-plastic response is outlined in this chapter. Novel aspects comprise new robust cutting techniques through elements for crack propagation in two-dimensional solids using virtual elements, phase field methods for the detection and path-following of fracture, and a combination of cutting and phase field approaches for efficient solutions of engineering problems. Applications in fracture mechanics lead generally to problems which, from the mathematical point of view, have a low regularity. This is related to the fact that sharp interfaces are present in the model and additionally inelastic material response has to be taken into account, especially when crack growth is considered. These type of problems

P. Wriggers et al., *Virtual Element Methods in Engineering Sciences*,
https://doi.org/10.1007/978-3-031-39255-9_10

need usually discretization schemes with low order since higher order schemes are prone to oscillating response.

The theoretical background of discretization schemes for fracture using virtual elements can be found in several publications together with the associated algorithmic treatment, see e.g. Nguyen-Thanh et al. (2018), Benedetto et al. (2018), Aldakheel et al. (2018a, 2019), De Bellis et al. (2018), Hussein et al. (2019, 2020) and Benvenuti et al. (2022).

10.1 Fracture Analysis Using Damage Mechanics

A classical possibility to predict fracturing processes is provided by damage mechanics. This approach was developed over the last decades with focus on the formulation of continuum damage models that can predict irreversible phenomena related to the onset and evolution of micro-cracking, void formation and strain-softening in quasi-brittle materials. The starting point for this development can be found in Kachanov (1958), different phenomenological models have been proposed both accounting for isotropic, see Lemaitre (1996), Lubliner et al. (1989) and Simo and Ju (1987), and anisotropic, see Govindjee et al. (1995) and Lemaitre et al. (2001), damage response.

These damage models lead to mesh dependent solutions for discretization techniques, like the finite element method, once softening evolves. This is due to the fact that the governing differential equations loose ellipticity and thus resort to ill-posed boundary value problems. There exist different strategies to overcome mesh dependency. One possibility is to employ non-local continuum theories, see e.g. de Borst (1991) and de Borst et al. (1993), which possess intrinsic regularization properties due to the introduction of a characteristic length. Another one is viscous regularization, see e.g. Needleman (1988). Further solutions to the pathological mesh dependency in damage models are non-local constitutive models, see e.g. Bažant and Jirasek (2002) for a comprehensive survey, or local models properly enriched by a dependence of the material properties at the element level, as in Oliver (1989).

We focus here on an isotropic damage law which is formulated in the framework of linearized kinematics with different thresholds for tensions and compression states. Additionally, we adopt as regularization techniques one of the last two mentioned approaches to which we will refer as nonlocal and local.

In this section we will describe a local and non-local scalar isotropic damage model that is employed together with the virtual element method for the prediction of failure, more details can be found in De Bellis et al. (2018).

10.1.1 Governing Equations for Isotropic Damage Model

We assume for the solid small strains and linear elastic behaviour before damage initiation. Thus the geometrically linear model discussed in Sect. 2.2.1 can be used together with the kinematical relation (2.9), defining the strain $\boldsymbol{\varepsilon} = \nabla_s \boldsymbol{u}$, and the weak form in Sect. 2.3.1.

For isotropic damage a scalar variable d is introduced, satisfying $0 \leq d \leq 1$. This variable describes the development of micro-cracks, microvoids and micro-cavities in a material and reduces the strain energy function $\Psi^0(\boldsymbol{\varepsilon})$ with respect to the damage state

$$\Psi(\boldsymbol{\varepsilon}, d) = (1 - d)\, \Psi^0(\boldsymbol{\varepsilon}). \tag{10.1}$$

Note, that the reduced strain energy $\Psi(\boldsymbol{\varepsilon}, d)$ approaches zero for increasing damage $d \rightarrow 1$. The initial undamaged elastic energy is defined as Ψ^0, see (2.43).

The energetic consistency of the constitutive model is ensured by the fulfillment of the Clausius–Planck inequality, see e.g. Malvern (1969),

$$\dot{D} = \left(\boldsymbol{\sigma} - \frac{\partial \Psi}{\partial \boldsymbol{\varepsilon}} \right)^T \dot{\boldsymbol{\varepsilon}} - \frac{\partial \Psi}{\partial d} \dot{d} \geq 0 \tag{10.2}$$

where $\dot{D}$ is the rate of the mechanical energy dissipation defined for arbitrary infinitesimal variations $\dot{\boldsymbol{\varepsilon}}$. The constitutive relations follow from, see Coleman and Noll (1963),

$$\boldsymbol{\sigma} = \frac{\partial \Psi}{\partial \boldsymbol{\varepsilon}} = (1 - d)\, [\Lambda \ \mathrm{tr}(\boldsymbol{\varepsilon})\, \boldsymbol{I} + 2\mu \boldsymbol{\varepsilon}] = (1 - d)\, \boldsymbol{\sigma}^0 \tag{10.3}$$

and

$$\dot{D} = -\frac{\partial \Psi}{\partial d} \dot{d} = -\Psi^0 \dot{d} \geq 0. \tag{10.4}$$

with the stress, $\boldsymbol{\sigma}^0 = \Lambda \ \mathrm{tr}(\boldsymbol{\varepsilon})\, \mathbf{I} + 2\mu \boldsymbol{\varepsilon}$, for the initially undamaged isotropic elastic material.

Following e. g. Simo and Ju (1987), an equivalent effective stress τ is defined as a suitable energy norm of the undamaged stress tensor $\boldsymbol{\sigma}^0$. This measure will be applied to compare different material states. In this section we adopt a damage model that allows for different thresholds in tension and compression leading to

$$\tau = \left(\zeta + \frac{1 - \zeta}{n} \right) \sqrt{\Psi^0(\boldsymbol{\varepsilon})}, \quad \text{with} \quad \zeta = \frac{\sum_{i=1}^{3} \langle \sigma_i^0 \rangle}{\sum_{i=1}^{3} |\sigma_i^0|}, \tag{10.5}$$

where ζ is a weight factor depending on the elastic principal stresses σ_i^0, $\langle \bullet \rangle$ is the Macaulay bracket, and $n = f_c / f_t$ is the ratio between compressive f_c and tensile f_t strength of the material.

The damage criterion is defined in terms of a limit damage surface, i.e. a function $F(\tau^t, r^t)$, that divides the admissible stress space into an elastic domain (when $F < 0$) and a damage domain (when $F = 0$). The damage surface depends on the equivalent effective stress $\tau(t)$ and on a material parameter representing the damage threshold $r(t)$ at the current time t.

The most general form of $F(\tau, r)$ among different possibilities is

$$F(\tau, r) = G[\tau(t)] - G[r(t)], \quad \forall t \geq 0, \tag{10.6}$$

where $G(\bullet)$ is a suitable monotonic scalar function.

Damage is governed by the following evolution equations

$$\dot{r} = \dot{\gamma}, \quad \dot{d} = \dot{\gamma} \frac{\partial F}{\partial \tau}. \tag{10.7}$$

Here, equivalently to the plastic multiplier in rate independent plasticity, $\dot{\gamma}$ is a damage multiplier. The evolution equations of damage have to satisfy the following Karush–Kuhn–Tucker relations (loading-unloading conditions)

$$F(\tau, r) \leq 0, \quad \dot{\gamma} \geq 0 \qquad \text{and } \ \dot{\gamma}\, F(\tau, r) = 0. \tag{10.8}$$

It is possible to directly integrate the evolution of the internal variables, as in Simo and Ju (1987), and obtain

$$r(t) = \max\left[r(0), \max(\tau(s))\right], \quad 0 \leq s \leq t \quad d = G[r(t)], \tag{10.9}$$

where $r(0)$ is a characteristic parameter of the material, i.e. the initial damage threshold for the virgin material and $\max(\tau(s))$ is the maximum equivalent effective stress that occurs within the time span $0 \leq s \leq t$.

Among different possible choices of explicit functions for the scalar damage d, two options are frequently used, see e.g. Scotta et al. (2001):

1. A linear damage law

$$G_1[r(t)] = \frac{1}{1 + H_1}\left(1 - \frac{f_t}{r(t)}\right), \quad f_t \leq r(t) \leq \infty, \tag{10.10}$$

   with E being the Young's modulus, G_f the fracture energy per unit area and H_1 a constant defined by $H_1 = f_c^2/(2n^2 G_f E)$.
2. An exponential damage law

$$G_2[r(t)] = 1 - \frac{f_t}{r(t)} exp\left[H_2\left(1 - \frac{r(t)}{f_t}\right)\right], \quad f_t \leq r(t) \leq \infty, \tag{10.11}$$

   where $H_2 = \left(\frac{n^2 G_f E}{f_c^2} - \frac{1}{2}\right)^{-1} \geq 0$.

The numerical response of strain softening materials is characterized by strain localization and mesh dependency. The latter can be overcome by two alternative regularization techniques:

- The first approach introduces an explicit dependence of the constitutive model on the size of the numerical discretization element. Following Oliver (1989) and Scotta et al. (2001) among others, it is possible to modify the expressions of H_1 in (10.10) and H_2 in (10.11) as

$$H_1 = f_c^2 l_c^i/(2n^2 G_f E) \qquad \text{and} \;\; H_2 = \left(n^2 G_f E/(f_c^2 l_c^i) - 1/2\right)^{-1} \geq 0 \quad (10.12)$$

  with l_c^i being the characteristic length that in 2D has been defined either as $l_c^1 = \sqrt{\Omega_v}$, (Scotta et al. 2001), or as $l_c^2 = \Omega_v / \int_A (\partial\phi/\partial x)\,\mathrm{d}A$, (Oliver 1989), where Ω_v is the area of the element, ϕ is a non-dimensional, continuous and differentiable function describing the displacement in the localization band and x is a local coordinate, perpendicular to the localization band.
- The second approach introduces non-locality to the damage theory, as proposed by Pijaudier-Cabot and Bažant (1987) and widely adopted in later works, see Bažant and Jirasek (2002). The non-local damage theory relies on the non-local equivalent stress $\overline{\tau}$ that can be defined as the weighted average of the local equivalent effective stress τ over a representative spherical volume surrounding each material point $\mathbf{x}$

$$\overline{\tau} = \frac{1}{\Omega_r(\mathbf{x})} \int_{\Omega} \psi(\mathbf{x} - \mathbf{s})\tau \mathrm{d}s \quad (10.13)$$

  where Ω is the area of the considered body, $\Omega_r(\mathbf{x}) = \int_\Omega \psi(\mathbf{x} - \mathbf{s})\,\mathrm{d}s$ is the representative volume and $\psi(\mathbf{x} - \mathbf{s})$ is a properly chosen weighting function that usually takes the form, see Bažant and Pijaudier-Cabot (1989) and Scotta et al. (2001),

$$\psi(\mathbf{x} - \mathbf{s}) = exp\left(-\frac{||\mathbf{x} - \mathbf{s}||^2}{2(l_c^3)^2}\right). \quad (10.14)$$

  The internal length l_c^3 is a parameter that controls the localization zone. As pointed out in Bažant and Pijaudier-Cabot (1989), its value depend on the material considered and, i.e. in the case of standard concrete this parameter can be taken as $3d_a$, and thus is directly related to the maximum size d_a of aggregates.

  The non-local equivalent stress $\overline{\tau}$ replaces its local counterpart τ in the damage laws (10.10), for the linear case, and (10.11), for the exponential case.

10.1.2 Virtual Element Formulation for Damage

As in the previous sections, we use an alternative to the weak form for the construction of a virtual element for the analysis of isotropic damage. Again it is possible to introduce a pseudo potential energy

$$U(\mathbf{u}) = \int_\Omega \left[\Psi(\boldsymbol{\varepsilon}(\boldsymbol{u}), d) - \bar{\boldsymbol{f}} \cdot \boldsymbol{u}\right] d\Omega - \int_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \boldsymbol{u} \, d\Gamma \quad (10.15)$$

where the damage variable d has to be frozen in the first variation. As discussed in Chap. 8, this form has advantages when the software codes are automatically generated using *AceGen*, see Korelc and Wriggers (2016).

The virtual element is based on the geometrically linear model described in Sect. 10.1.1. The associated virtual element formulation can be found in Chap. 6 for a two-dimensional analysis. The formulation for the linear ansatz in Sect. 6.1.1, using Voigt notation, yields the projected constant strain $\hat{\boldsymbol{\varepsilon}}_\pi = \mathbb{P}^{(2,1)}_{\nabla\varepsilon}\,\mathbf{u}_v$, see (6.4), and the projected displacement field $\boldsymbol{u}_\pi = \mathbf{H}^{(2,1)}_u(X, Y)\,\mathbb{P}^{(2,1)}_u\,\mathbf{u}_v$, see (3.58). Here $\mathbf{H}^{(2,1)}_u$ contains the linear ansatz functions and $\mathbb{P}^{(2,1)}_u$ and $\mathbb{P}^{(2,1)}_{\nabla\varepsilon}$ are the projectors that describe the projected fields in terms of the nodal displacements $\mathbf{u}_v$ of the virtual element. Since the details of the projection were already provided in Chap. 6 only the changes related to the damage model are discussed in the following.

The pseudo potential energy in (10.15) can be written by summing up all the element contributions for the n_v virtual elements of area Ω_v. This results in

$$U(\mathbf{u}_h, d) = \sum_{v=1}^{n_v} [U_c(\boldsymbol{u}_\pi, d) + U_{stab}(\boldsymbol{u}_h - \boldsymbol{u}_\pi, d)] \tag{10.16}$$

where U_c is the consistency term related only to the projection of the displacements, and thus in case of a linear ansatz involves only a constant strain field over the elements, and U_{stab} is the stabilization term, required to avoid element rank deficiencies, see Wriggers et al. (2017).

The consistency term in (10.16) has the form

$$U_c(\boldsymbol{u}_\pi, d) = \int_{\Omega_v} \left(\Psi[\boldsymbol{\varepsilon}(\boldsymbol{u}_\pi), d] - \bar{\boldsymbol{f}} \cdot \boldsymbol{u}_\pi\right)\, \mathrm{d}\Omega - \int_{\Gamma_v} \bar{\boldsymbol{t}} \cdot \boldsymbol{u}_\pi \mathrm{d}\Gamma. \tag{10.17}$$

The pseudo strain energy can be defined in matrix form using the Voigt notation, see (6.5),

$$\Psi[\hat{\boldsymbol{\varepsilon}}, d] = (1 - d)\Psi^0[\hat{\boldsymbol{\varepsilon}}] = (1 - d)\frac{1}{2}\mathbf{u}_v^T\,[\mathbb{P}^{(2,1)}_{\nabla\varepsilon}]^T\,\mathbb{C}\,\mathbb{P}^{(2,1)}_{\nabla\varepsilon}\,\mathbf{u}_v. \tag{10.18}$$

Both, the damage variable d and the components of the strain tensor $\hat{\boldsymbol{\varepsilon}}$, are constant in each element Ω_v which allows the simple evaluation of the integral in (10.17)

$$U_c(\mathbf{u}_\pi, d) = (1 - d)\frac{\Omega_v}{2}\,\mathbf{u}_v^T\,[\mathbb{P}^{(2,1)}_{\nabla\varepsilon}]^T\,\mathbb{C}\,\mathbb{P}^{(2,1)}_{\nabla\varepsilon}\,\mathbf{u}_v. \tag{10.19}$$

Note that $\Psi[\hat{\boldsymbol{\varepsilon}}, d]$ is a non-linear function of $\mathbf{u}_v$ and the damage variable d.

The stabilization term has now to be derived for the case of scalar damage. This derivation is based on the energy stabilization procedure in Sect. 6.1.3

$$U_{stab}(\boldsymbol{u}_h - \boldsymbol{u}_\pi, d)) = \widehat{U}(\boldsymbol{u}_h, d) - \widehat{U}(\boldsymbol{u}_\pi, d) \tag{10.20}$$

where $\widehat{U}(\boldsymbol{u}_h, d)$ is computed using an internal mesh of triangles (T).

For damaged materials pseudo potential is modified

$$\widehat{\Psi}(\hat{\boldsymbol{\varepsilon}}(\bullet), d) = \frac{1}{2}\boldsymbol{\varepsilon}(\bullet)^T \widehat{\mathbb{C}}(d)\hat{\boldsymbol{\varepsilon}}(\bullet) \tag{10.21}$$

where the strain tensor $(\bullet)$ is in turn depending on $\boldsymbol{u}_h|_T$ or $\boldsymbol{u}_\pi$, and $\widehat{\mathbb{C}}(d)$, see (2.46), is a modified damaged isotropic constitutive tensor characterized by the Lamé parameters $\widehat{\Lambda}$ and $\widehat{\mu}$, that are computed from the modified values of Young's modulus $\widehat{E}$ and Poisson ratio $\widehat{\nu}$

$$\widehat{E} = (1-d)E \qquad \widehat{\nu} = 0.3 \tag{10.22}$$

with E being the Young's modulus of the actual material and d the damage variable. The value of the Poisson ratio is kept constant since it does not influence the convergence behaviour of the element.

A specific algorithmic treatment for damage models can be found in Oliver et al. (1990). It is here extended to the virtual element method. The integration of the evolution equation is performed within the time step $\Delta t = t - t_n$, the constitutive equations, briefly summarized in Box 1, are solved at the element level. In what follows quantities without an index represent the current time and quantities with index n are related to the time t_n.

In the particular case of nonlocal regularization, the value of τ is replaced by its nonlocal counterpart $\bar{\tau}$. It is evaluated, using Eqs. (10.13) and (10.14), by averaging the stresses related to virtual elements that included in the representative circular area surrounding the given element. The actual value of the element damage d is evaluated in the same way. The damage laws (10.10) and (10.11) are used in Box 1 to compute the damage variable d_v within the element Ω_v.

Given: $\hat{\boldsymbol{\varepsilon}}_v, d_{v\,n}, r_{v\,n}$ Find: d_v, r_v

$$\boldsymbol{\sigma}_v = \mathbb{C}\,\hat{\boldsymbol{\varepsilon}}_v = \mathbb{C}\,\mathbb{P}^{(2,1)}_{\nabla\varepsilon}\,\mathbf{u}_v$$
$$\tau_v = \left(\zeta + \tfrac{1-\zeta}{n}\right)\sqrt{\Psi^0[\hat{\boldsymbol{\varepsilon}}_v]}$$

$$r = \max\{r_{v\,n}, \tau_v\}$$
$$d_v = G(r_v)$$

Box 1 Damage evaluation at the element level

Note that the current damage is used within an element in (10.16) for both, the calculation of the consistency term and the calculation of the stabilization term.

Global equilibrium follows from minimization of (10.16). For this a Newton–Raphson algorithm is used. Thus the element residual $\mathbf{R}(\mathbf{u}_v, d_v) = \mathbf{R}_c(\mathbf{u}_v, d_v) + \mathbf{R}_{stab}(\mathbf{u}_v, d_v)$ and tangent matrix $\mathbf{K}(\mathbf{u}_v, d_v) = \mathbf{K}_c(\mathbf{u}_v, d_v) + \mathbf{K}_{stab}(\mathbf{u}_v, d_v)$ have to be computed. The residual vector and the tangent matrix are derived by applying the symbolic tool *AceGen*.

The element residual vector for the consistency term follows as

$$\mathbf{R}_c(\mathbf{u}_v, d_v) = \Omega_v \frac{\partial \Psi(\hat{\boldsymbol{\varepsilon}}, d_v)}{\partial \mathbf{u}_v}|_{d_v = const.} \tag{10.23}$$

where the formalism $d_v = const.$ stands for a constant value of d_v during the automatic differentiation process while $\hat{\boldsymbol{\varepsilon}}$ depends on $\mathbf{u}_v$ via the projection $\hat{\boldsymbol{\varepsilon}} = \mathbb{P}^{(2,1)}_{\nabla\varepsilon}\,\mathbf{u}_v$. The stabilization term is evaluated in the same way

$$\mathbf{R}_{stab}(\mathbf{u}_v, d_v) = \frac{\partial \widehat{U}(\mathbf{u}_v, d_v)}{\partial \mathbf{u}_v}|_{d_v = const.} - \Omega_v \frac{\partial \widehat{\Psi}(\hat{\boldsymbol{\varepsilon}}, d_v)}{\partial \mathbf{u}_v}|_{d_v = const.} \tag{10.24}$$

where the pseudo potential $\widehat{U}(\mathbf{u}_v, d_v)$ is computed using (10.20), while $\widehat{\Psi}(\hat{\boldsymbol{\varepsilon}}, d_v)$ is the modified positive definite pseudo energy directly evaluated for the virtual element. Analogously, the contributions to the tangent matrix are given by

$$\mathbf{K}_c(\mathbf{u}_v, d_v) = \frac{\partial \mathbf{R}_c(\mathbf{u}_v, d_v)}{\partial \mathbf{u}_v}, \qquad \mathbf{K}_{stab}(\mathbf{u}_v, d_v) = \frac{\partial \mathbf{R}_{stab}(\mathbf{u}_v, d_v)}{\partial \mathbf{u}_v}. \tag{10.25}$$

10.1.3 Numerical Examples

Several numerical examples illustrate the performances of the virtual elements for isotropic damage, adopting either local or non-local regularization techniques. Comparison with existing finite element formulations highlight the specific features of the virtual element method. All computations are based on a Newton–Raphson algorithm with load stepping. The meshes were automatically generated by the meshing tools in *Mathematica*. Mesh refinements, unless otherwise specified, are uniform in the sense that finer meshes are included in the coarser meshes, both for regular and distorted meshes.

Tension specimen test. A classical test proposed by Oliver (1989) is considered. The specimen is depicted in Fig. 10.1. The material parameters are chosen as Young's modulus $E = 200000$ kp/cm^2, Poisson ratio $\nu = 0.2$, tensile strength $f_t = 10$ kp/cm^2 and fracture energy $G_f = 0.125$ kp/cm. The computation is performed by prescribing a displacement u in horizontal direction. Due to symmetry, on half of the plate is discretized using meshes with different element types, including distorted eight node quadrilaterals and Voronoi elements. We adopt both regularization techniques presented in Sect. 10.1.1 in (10.10) and (10.11). Due to the specimen shape, damage will start at the center of the plate.

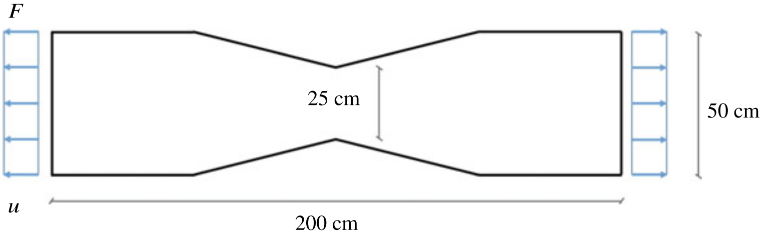


Fig. 10.1 Tension test: geometry and data

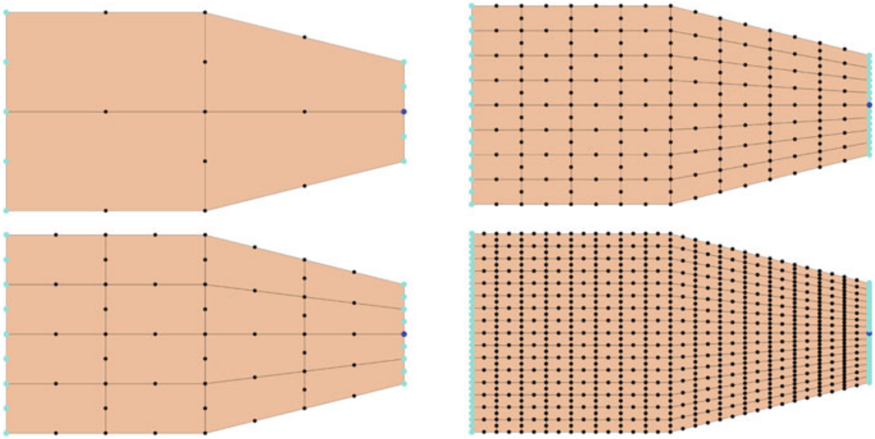


Fig. 10.2 8-node regular virtual element mesh: top left: B1, bottom left: B2, top right: B3, and bottom right: B4

First, the linear damage law, see (10.10), will be employed with the regularization (10.12) using l_c^2. In this case, the response computed in Oliver (1989) using finite elements is matched perfectly with a discretization of virtual elements with 8 nodes. The sequence of the regular meshes B1 (2 × 2), B2 (4 × 4), B3 (8 × 8) and B4 (16 × 16), shown in Fig. 10.2, yields the force-displacement curves depicted in Fig. 10.3. As reported in the paper by Oliver (1989), localization of damage occurs within the first row of elements, close to the neck which is also recovered by the virtual element computation.

The convergence trend of the global curves is perfectly matched and the finest mesh reproduces the expected behaviour correctly, as reported in Fig. 10.3. In a second step, the distorted meshes B5 and B6, depicted in Fig. 10.4, are employed, including virtual elements with non convex shapes. The mesh density corresponds to the meshes B3 and B4. The global curves are very close to the corresponding ones in Fig. 10.3 and are almost undistinguishable in the plot.

The second regularization approach is based on non local damage, see (10.13) and (10.14). Here the internal length $l_c^3 = 1$ cm is introduced. In this case finer meshes are required in order to exploit the nonlocal regularization technique. The discretization

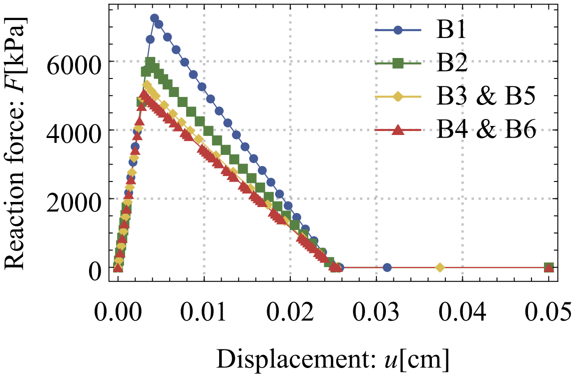


Fig. 10.3 Force-displacement curves for meshes B1, B2, B3 and B4

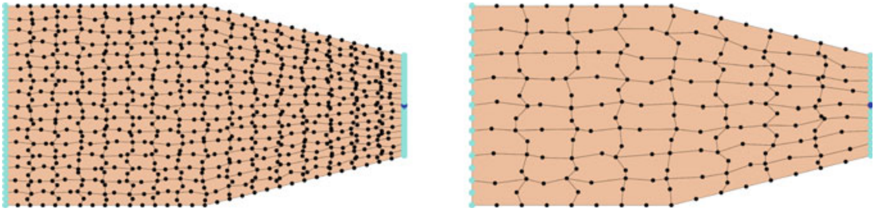


Fig. 10.4 Distorted meshes. 8-node VEM elements (left: B6 and right: B5)

is based on centroidal Voronoi tessellations, where the seeds coincide with centroids of the resulting Voronoi cells. Figure 10.5 shows the global curves which are obtained by meshes of 2000 (mesh V1) and 8000 (mesh V2) virtual elements, respectively. The two load-displacement curves match almost perfectly.

However, the load-displacement curves deviate from the curves obtained with the local regularization, see Fig. 10.3. This can be attributed to the different damage distributions within the specimen. In the second case, the damage emerges in the central part of the specimen, as expected, and develops within a curved region, as shown in Fig. 10.6. These damage plots, both for V1 and V2, correspond to the final state of the loading process. The two damage distributions are spread in the same region of the specimen, irrespective of the mesh density, confirming the effectiveness of the regularization technique.

Brazilian Test. In the second example experimental results and predictions by the virtual element method are compared. A splitting test of a cylindrical granite specimen is considered, which is described in Rocco et al. (1999a, b, c). Figure 10.7a presents the geometry of the specimen. The material parameters stem from experimental data which yield Young's modulus $E = 33900$ MPa, Poisson ratio $\nu = 0.2$, tensile strength $f_t = 10.1$ MPa and fracture energy $G_f = 0.167$ N/mm. The vertical load is applied through two bearing strips and two strain gauges ($L = 0.8D$, with D being the diameter of the specimen and L the gauge length of the extensometer)

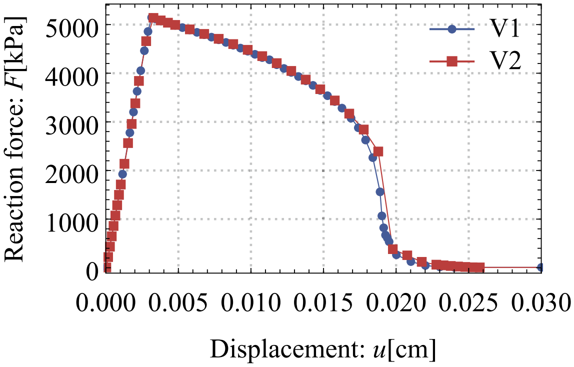


Fig. 10.5 Force-displacement curves for meshes V1 and V2 shown in Fig. 10.4

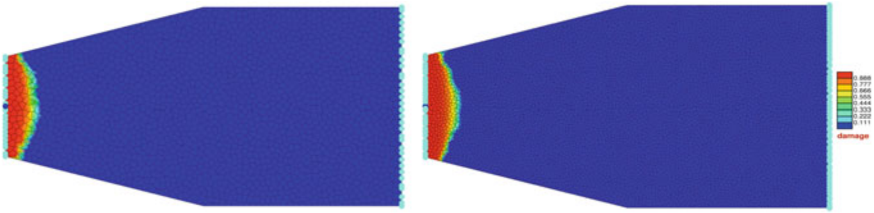
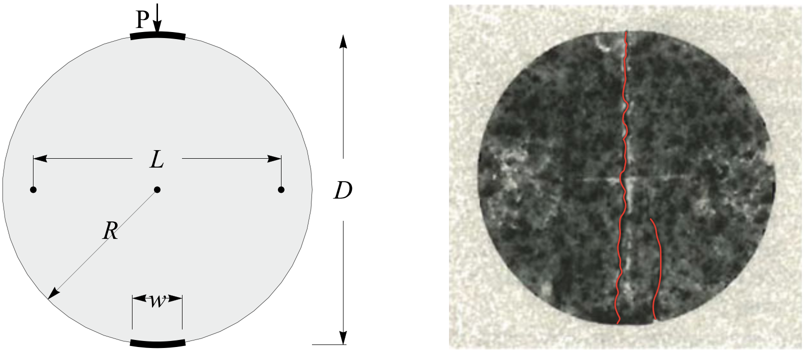


Fig. 10.6 Damage distributions using Voronoi meshes



(a) System

(b) Cracked plate, see Rocco et al. (1999a)

Fig. 10.7 Brazilian test

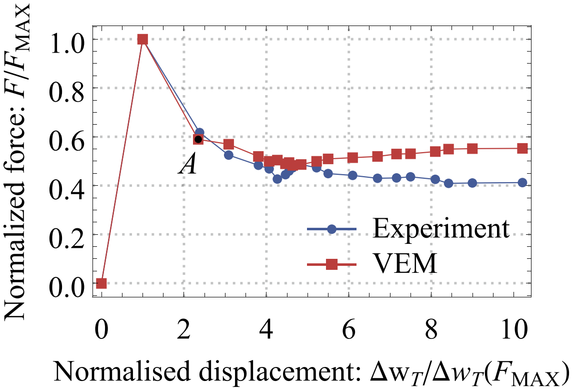


Fig. 10.8 Global curves load versus transverse deformation: comparison between experimental and numerical results

are fixed on the opposite faces of the cylinder to measure the diametric deformations along the axis orthogonal to the loading direction. The diameter of the specimen is $D = 120$ mm, the length of the bearing strips is $w = 19.2$ mm and the thickness of the specimen is $t_s = 30$ mm.

The results of the experimental work demonstrate that the cracking process starts at the center of the specimen and a first single crack grows vertically in loading direction. At the end of this first failure stage the specimen is split in two halves that start to resist the load separately. A subsequent failure mechanism is then observed, involving the formation of secondary cracks. In general this type of failure mechanism can induce a nearly vertical crack development at both sides of the bearing strips.

In the actual experimental case, a single crack occurs at the right bottom side of the bearing strips, propagating in sub-vertical direction. This is illustrated in Fig. 10.7b where the observed cracks are highlighted in red. Two peak loads occur in the global response curve as a direct consequence of this failure in two-steps.

The numerical tests are performed assuming plane strain conditions. The specimen is discretized using 2000 centroidal Voronoi virtual elements. The non-local damage model is adopted with a characteristic length l_c^3= 3 mm, this value is approximately $\simeq 3d_g$ with d_g being the estimated maximum size of grains in granite.

The comparison between the experimental and the numerical curves of global load versus the diametrical deformation in Fig. 10.8 illustrates that the highly fragile behaviour of the granite is well predicted by the numerical simulations. The vertical load F is normalized to the first peak load F_{MAX}, as well as the relative horizontal displacement $\Delta \mathrm{w}_T$ is normalized to the respective value $\Delta \mathrm{w}_T(\ F_{\mathrm{MAX}})$, as proposed in Rocco et al. (1999a). The peak load predicted in the numerical analysis is $F_{\mathrm{MAX}}^{VEM} =$ 64.5 kN. This prediction is in good agreement with the value obtained by the authors of the experimental work

$$F_{\mathrm{MAX}}^{EXP} = \frac{f_t \pi s D}{2[1-(b/D)^2]^{3/2}} \simeq 65 \text{ kN}. \tag{10.26}$$

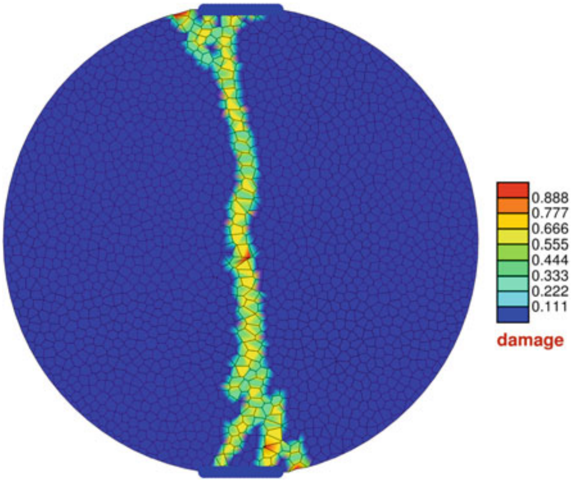


Fig. 10.9 Damage distribution in the specimen corresponding to point A in the global curve in Fig. 10.8

Also the post peak behaviour is predicted reasonably well since the magnitude of the load drop is comparable with the experimental data.

Figure 10.9 depicts the damage distributions corresponding to the point A of the load-displacement curve in Fig. 10.8. It demonstrates a good agreement of the experimental results in the left part of Fig. 10.7 with the numerical prediction. The central crack localizes and spreads along the vertical diametrical axis. Since the Voronoi elements are randomly generated, the analysis has been repeated for different meshes, characterized by about the same number of elements, to make sure that the damage path and the associated global response are not affected by the mesh. The same mechanism of rupture in the splitting test is recovered in all the simulations and only differences of about 5% are found for the peak value.

10.2 Brittle Crack-Propagation

Another classical fracture approach is based on linear fracture mechanics LFM which accounts for the singular behaviour of the stress field at the crack tip. Within the theoretical framework stress intensity factors (SIFs), introduced by Irwin (1956), are regarded as the fundamental quantity in fracture mechanics to measure the strength of the stress singularity in the vicinity of a crack tip. In the literature, several methods have been proposed to calculate stress intensity factors, among them are the displacement extrapolation method (Chan et al. 1970), the crack opening displacement (COD) (Paris and Sih 1965), the virtual crack extension (Hellen 1975), the J-integral (Eshelby 1974; Rice and Rosengren 1968) and the interaction integral method (Yau et al. 1980). In Barsoum (1974), Henshell and Shaw (1975) special types of elements were developed that base on the 8-nodes isoparametric formulation and use a specific element topology to model a singularity. This approach allows to capture the $r^{-1/2}$

singularity when the mid-side nodes are moved to the quarter position near the crack tip. A hybrid crack element was introduced in Tong et al. (1973), Kuna and Zwicke (1990), Karihaloo and Xiao (2001), which leads to a direct and accurate computation of the stress intensity factors as well as the coefficients of the higher order terms in the elastic asymptotic crack tip field.

In this regard, computational methods for the prediction of fracture mechanism are numerous. Those methods were explored that have potential advantages, such as flexibility in mesh generation and the choice of element shapes. When it comes to the modeling of moving discontinuities as in the case of crack propagation, finite elements are limited due to the required conforming mesh topology to track the crack path during the growth. Such limitation can be overcome by using re-meshing techniques in the vicinity of crack tip, which, however, leads to time-consuming simulations. Furthermore, the results can be mesh-dependent and difficult to validate. Among others, Bittencourt et al. (1996) used a local re-meshing algorithm to adjust the mesh after each step of propagation. Later, Bouchard et al. (2000) proposed an advanced automatic re-meshing technique to model crack propagation using the discrete crack approach. Despite the great success of these methods, the crack propagates only along the edges of the elements, resulting in a loss of accuracy in the crack path prediction. For an overview with regard to existing technologies for fracture with the finite element method see e.g. Kuna (2013).

An alternative to the finite element methodology are meshless methods Belytschko et al. (1994, 1995), Belytschko and Tabbara (1996), Fleming et al. (1998) which do not require the classical data structures of the mesh, which simplifies the application of the method for arbitrary crack growth. Further approaches for the numerical simulation of fracture processes are the boundary element method (Ingraffea et al. 1983; Portela et al. 1992) and the eXtended Finite Element Method (XFEM) (Belytschko and Black 1999; Moës et al. 1999; Pezeshki et al. 2018; Stolarska et al. 2001; Sukumar and Prévost 2003). In XFEM the standard finite element approximation is enriched by discontinuous functions and asymptotic fields to improve the singular solution in the vicinity of the crack tip. For specific problems, a flexible choice of element shapes is preferred. To this end, there have been interesting developments in the use of polygonal or polyhedral elements. Several methods were proposed in the direction of polygonal finite element method, see Sukumar (2004), Sukumar and Malsch (2006), Biabanaki et al. (2014) and Khoei et al. (2015). Another variant is the scaled boundary finite element method, see e.g. Yang and Deeks (2007), Ooi and Yang (2009), Ooi et al. (2012), Dai et al. (2015) and Song et al. (2018).

Each of these methods has its own specifications and thus needs experts for a correct and efficient application. In this chapter the virtual element method proposed in Beirão da Veiga et al. (2013) will be explored since it can bring some new features to the numerical solution of fracture problems. These are related to the advantage of the virtual element method to use arbitrary polygonal (2D) and polyhedral (3D) element shapes with arbitrary number of nodes.

Due to the low regularity of the crack propagation problem a low order virtual element formulation will be employed throughout this chapter. For most applications the continuum basis is a small strain linear elastic model that will be enhanced by the specific amendments to model cracks. Once a crack has been initiated, the direction of growth can be predicted using different propagation criteria: the maximum circumferential stress criterion, the maximum strain energy release rate criterion and the minimal strain energy density criterion. Here the maximum circumferential stress criterion, introduced in Erdogan and Sih (1963), will be considered. To determine this stress criterion, the stress intensity factors for mixed-mode loading conditions are calculated. The stress intensity factors are then numerically obtained by using the interaction integral, (Yau et al. 1980), which accurately calculates these factors. When the direction of crack growth is known, there are two possibilities to allow the crack to propagate: either along the element interfaces, or within the elements. The first one can be accomplished by the classical finite element method. The second case can be implemented within the virtual element method. Here it is possible to first introduce a crack within an element and later to split the corresponding element in the direction of the crack propagation. This allows the crack to propagate directly through elements. This cutting technique was developed for virtual elements in Hussein et al. (2019) which provides the basis for the next section.

10.2.1 Equations of Brittle Crack Propagation

Consider an elastic body $\Omega \subset \mathbb{R}^2$ bounded by Γ undergoing small strains. As shown in Fig. 10.10, the boundary Γ is subdivided into Neumann boundary conditions on Γ_N, Dirichlet boundary conditions on Γ_D and a discontinuity interface on Γ_c (crack line in 2-D and crack surface in 3-D) such that $\Gamma = \Gamma_N \cup \Gamma_D \cup \Gamma_c$. The discontinuity Γ_c is composed of the upper Γ_c^+ and the lower Γ_c^- crack faces. The equilibrium equation of the solid body Ω was provided in (2.22) for large strains. It has the same form for the linearized theory used here

$$\mathrm{Div}\,\boldsymbol{\sigma} + \bar{\boldsymbol{f}} = 0 \qquad \text{in } \Omega, \tag{10.27}$$

which contains the stress tensor $\boldsymbol{\sigma}$ and the body force $\bar{\boldsymbol{f}}$. The crack face Γ_c is assumed to be traction-free

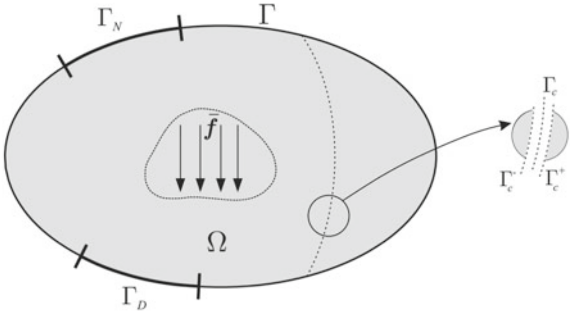


Fig. 10.10 Definition of solid with a crack and boundary conditions

$$\boldsymbol{t}^+ = \boldsymbol{\sigma}\, \boldsymbol{n}^+ = \mathbf{0} \qquad \text{on } \Gamma_c^+, \tag{10.28}$$

$$\boldsymbol{t}^- = \boldsymbol{\sigma}\, \boldsymbol{n}^- = \mathbf{0} \qquad \text{on } \Gamma_c^-, \tag{10.29}$$

where $\boldsymbol{n}^+$ and $\boldsymbol{n}^-$ are the outward unit normal vectors defined on Γ_c^+ and Γ_c^-, respectively.

The infinitesimal strain tensor $\boldsymbol{\varepsilon} = \frac{1}{2}[\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T]$ appears in the linear elastic constitutive relation that follows from the strain energy $\Psi = \Lambda \,/\, 2\, [\mathrm{tr}(\boldsymbol{\varepsilon})]^2 + \mu\, \mathrm{tr}(\boldsymbol{\varepsilon} \cdot \boldsymbol{\varepsilon})$, see (2.43), where Λ and μ are the Lamé parameters. The stress tensor $\boldsymbol{\sigma}$ is obtained from the strain energy Ψ by differentiation with respect to the strain tensor, see (2.45), leading to $\boldsymbol{\sigma} = \Lambda\ \mathrm{tr}(\boldsymbol{\varepsilon})\, \boldsymbol{1} + 2\mu\boldsymbol{\varepsilon}$.

The equilibrium equation (10.27) together with the strains and the constitutive Eq. (2.43) can be equivalently recast together with the boundary conditions on Γ_c in form of the principle of stationary elastic potential

$$U\big(\boldsymbol{u}\big) = \int\limits_{\Gamma_c} [\![\boldsymbol{u}]\!] \cdot \boldsymbol{\sigma}\, \boldsymbol{n}^c \;\mathrm{d}\Gamma + \int\limits_{\Omega} [\Psi(\boldsymbol{\varepsilon}) - \bar{\boldsymbol{f}} \cdot \boldsymbol{u}]\; \mathrm{d}\Omega - \int\limits_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \boldsymbol{u}\; \mathrm{d}\Gamma, \tag{10.30}$$

where $[\![\boldsymbol{u}+]\!] = \boldsymbol{u}^+ - \boldsymbol{u}^-$ is the displacement jump across the crack surface Γ_c with the unit normal vector $\boldsymbol{n}^c$. Due to the assumption of a traction-free crack face, the first integral in (10.30) can be neglected, which simplifies the potential above for the linear elastic case to

$$U\big(\boldsymbol{u}\big) = \int\limits_{\Omega} [\Psi(\boldsymbol{\varepsilon}(\boldsymbol{u})) - \bar{\boldsymbol{f}} \cdot \boldsymbol{u}]\; \mathrm{d}\Omega - \int\limits_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \boldsymbol{u}\; \mathrm{d}\Gamma. \tag{10.31}$$

10.2.2 Modeling Crack Propagation with Virtual Elements

The basis for the discretization of the crack propagation in brittle materials is a small strain elastic virtual element with linear ansatz. This ansatz function can be found in Sect. 3.1 in (3.5). The projection, see Sect. 3.1.4 in (3.40), yields the displacement gradient $\nabla \boldsymbol{u}_\pi = \mathbb{P}_{\nabla}^{(2,1)}\, \mathbf{u}_v$, see (3.51), and hence the linear strain $\boldsymbol{\varepsilon}_\pi$ follows as $\boldsymbol{\varepsilon}_\pi = \frac{1}{2}[\nabla \boldsymbol{u}_\pi + (\nabla \boldsymbol{u}_\pi)^T]$. Together with the explicit form of the linear ansatz $\boldsymbol{u}_\pi = \mathbf{H}_u^{(2,1)}(X, Y)\, \mathbb{P}_u^{(2,1)}\, \mathbf{u}_v$, see (3.58), the contributions of a virtual element Ω_v to the consistency part of the potential (10.31) can be computed, leading to

$$U_c^v(\boldsymbol{u}_\pi) = \int_{\Omega_v} \big[\Psi(\boldsymbol{\varepsilon}_\pi) - \bar{\boldsymbol{f}} \cdot \boldsymbol{u}_\pi\big]\; \mathrm{d}\Omega - \int_{\Gamma_v} \bar{\boldsymbol{t}} \cdot \boldsymbol{u}_h \mathrm{d}\Gamma. \tag{10.32}$$

The details of general virtual element formulation for small strain elastic materials are provided in Sect. 6.1. For the discretization of crack propagation the energy stabilization with inscribed triangles is used, see Sect. 6.1.3 which needs the specification of the stabilization parameter β.

The new ingredients regarding a virtual element formulation for the prediction of crack propagation are based on three stages:

1. Computation of stress intensity factors
2. Check of a criterion which yields the onset of crack propagation and the propagation direction
3. Introduction of a cutting algorithm in the virtual element scheme.

These three points will be discussed in detail below.

10.2.3 Computation of Stress Intensity Factors

The interaction integral method as a contour integral (I-integral) will be applied for the computation of the stress intensity factors.[1]

The I-integral is a generalization of the J-integral, see Rice (1968), and calculated as contour integral

$$I = \int\limits_{\Gamma} \left(W^{(1,aux)} \delta_{1j} - \boldsymbol{\sigma}_{ij} \frac{\partial u_i^{(aux)}}{\partial x_1} - \boldsymbol{\sigma}_{ij}^{(aux)} \frac{\partial u_i^{(1)}}{\partial x_1} \right) n_j \, \mathrm{d}\Gamma, \tag{10.33}$$

with

$$W^{(1,aux)} = \boldsymbol{\sigma}_{ij} \boldsymbol{\epsilon}_{ij}^{(aux)} = \boldsymbol{\sigma}_{ij}^{(aux)} \boldsymbol{\epsilon}_{ij}. \tag{10.34}$$

where the auxiliary fields ($u_i^{(aux)}, \boldsymbol{\sigma}_{ij}^{(aux)}, \boldsymbol{\epsilon}_{ij}^{(aux)}$) are given by the analytical Williams' solution, see Williams (1957). The actual fields ($u_i, \boldsymbol{\epsilon}_{ij}, \boldsymbol{\sigma}_{ij}$) are computed numerically, for more details see e.g. Hussein et al. (2019). It can be shown that the I-Integral is equivalent to the stress intensity factors K_{I} and K_{II}

$$I = \frac{2}{E'} (K_{\mathrm{I}} K_{\mathrm{I}}^{(aux)} + K_{\mathrm{II}} K_{\mathrm{II}}^{(aux)}). \tag{10.35}$$

By setting e.g. $K_{\mathrm{I}}^{(aux)} = 1$ and $K_{\mathrm{II}}^{(aux)} = 0$ in (10.35) the stress intensity factor for mode I can be determined as $K_{\mathrm{I}} = \frac{E'}{2} I$. In the same way the stress intensity factor follows for mode II: $K_{\mathrm{II}} = \frac{E'}{2} I$.

The I-Integral is evaluated using virtual elements close to the crack tip which is depicted by the end of the dashed line in the center of Fig. 10.11a. The following procedure is applied:

[1] When finite elements or XFEM are applied, the contour integral is most likely to be transformed into an equivalent area integral, see Moës et al. (1999). This, however, does not lead to accurate results since the low order virtual element formulation provides only constant stress and strain states within an element.

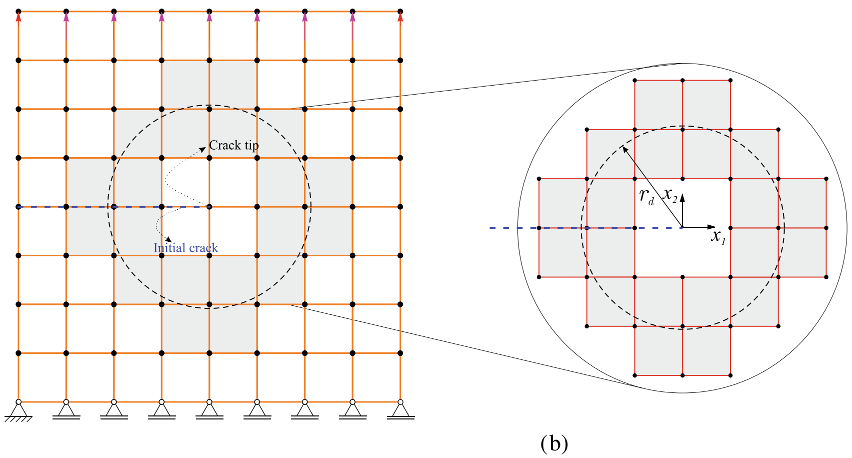


Fig. 10.11 Determination of the contour integral. Centered areas represent the crack tip elements and gray areas depict the elements cut by a circle of radius r_d

1. Find all elements included in a circle of radius r_d around the crack tip

$$r_d = \alpha \; h_{local}. \tag{10.36}$$

They are plotted with gray color in Fig. 10.11a, b. The factor α is a scaling factor and h_{local} is a characteristic length given by

$$h_{local} = \sqrt{\sum_{i=1}^{n_{tip}} \frac{A_{i,tip}}{n_{tip}}}, \tag{10.37}$$

where n_{tip} and $A_{i,tip}$ are the number and the area of the crack tip elements, respectively. The elements ($\Omega_{1,tip}$, $\Omega_{2,tip}$, $\Omega_{3,tip}$, $\Omega_{4,tip}$), sketched in Fig. 10.12a with yellow color, contain the crack tip node.
2. Select the set of elements for the computation of the I-integral. Here all edges shared by two adjacent elements will be neglected, as illustrated in Fig. 10.12b.
3. Eliminate all edges located within the circle, see Fig. 10.12c.
4. Compute the I-integral based on the set-up in Fig. 10.12c over all remaining edges as

$$I = \sum_{e=1}^{n_{edge}} I_e, \tag{10.38}$$

where n_{edge} is the number of edges of the contour around the circle.

By using (10.35), the stress intensity factors K_{I} (crack opening mode) and K_{II} (crack sliding mode) can be computed explicitly. The numerical integration of (10.33) can

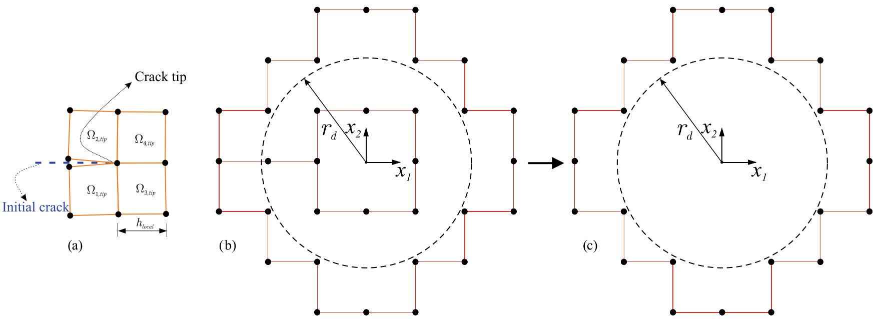


Fig. 10.12 Determination of the contour for edge integration

be performed by the trapezoidal or a Gaussian quadrature rule. In the numerical examples, Gauss quadrature with three Gauss points per edge is employed.

10.2.4 Stress Intensity Factor Analysis Using Virtual Elements

Two examples demonstrate the ability of the developed virtual element scheme to compute the stress intensity factors (SIFs) on the basis of the above introduced theoretical and algorithmic concept.

The double cantilever beam with a crack is used to investigate the performance of the virtual element formulation in bending dominated structural response with an edge crack with $a_0 = 10$ mm. The geometrical setup and the loading conditions of the specimen are depicted in Fig. 10.13.

The beam is considered to be thin with length to height ratio of $L\,/\,h = 20$ where $L = 20$ mm. Width and height of the cross section are $t = 2$ mm and $h = 2$ mm,

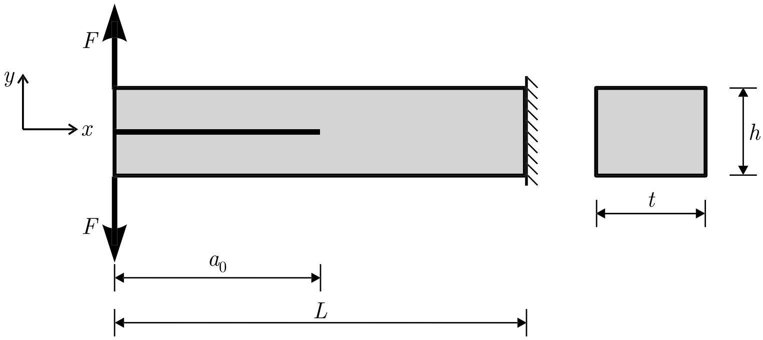
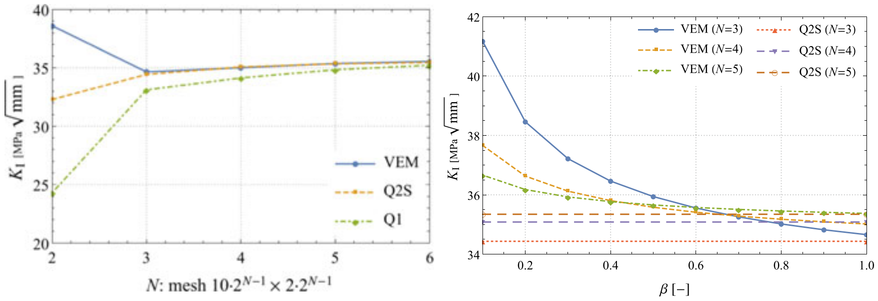


Fig. 10.13 Double cantilever beam with an edge crack. Geometry and boundary conditions



(a) Convergence study: stress intensity factor K_{I} (b) Influence of the stabilization parameter β

Fig. 10.14 Double cantilever beam with an edge crack

respectively. The specimen is clamped at the right end and loaded at the left side by a point load $F = 1$ N in y-direction. Young's modulus and the Poisson's ratio are given by $E = 10^3$ MPa and $\nu = 0.3$, respectively. Plane stress is assumed within this example.

The computation is performed with a linear virtual element defined by 8 nodes. The discretization is based on a regular mesh with rectangular shaped elements. A series of meshes with $10 \cdot 2^N \times 2^N$ elements in x and y-directions is used with the division parameter $N \in \{2, 3, 4, 5, 6\}$. The VEM results are compared with the solution of quadrilateral finite elements with 4-nodes (linear Q1 element) and 8-nodes (quadratic Q2S serendipity element). The Q2S element uses a quadratic interpolation for the displacement field and thus has superior convergence rates. The stress intensity factor analysis is based on the interaction integral, provided in Sect. 10.2.3, see (10.35).

The convergence behaviour of the stress intensity factor K_{I} is studied and illustrated in Fig. 10.14a for the different element formulations and mesh sizes. We notice that, the convergence response of the linear virtual element is as good as the quadratic Q2S element. However when using the Q1 element, due to its well known locking behaviour in bending situations, one needs a very fine mesh to achieve a convergent solution as shown in Fig. 10.14a.

The sensitive of the solution with respect to the stabilization parameter β is of interest for engineering applications. The impact of the parameter β on result for the stress intensity factor K_{I} is illustrated in Fig. 10.14b. It is visible that the increase of the mesh size N yields a better result which is close to the Q2S solution. However, the difference related to the value of K_{I} is small for all values of $\beta > 0.4$. Hence a good choice for the stabilization parameter is $\beta \in [0.4 - 0.6]$ which then was applied within the further numerical simulations.

A plate with an angled crack under tension is analyzed to compute mixed-mode stress intensity factors for different crack orientations. The geometry ($w =$ 10 mm, $a_0 = 0.5$ mm) and the loading ($\sigma = 2000$ MPa) of the specimen are given in Fig. 10.15. Plane strain conditions are assumed in this investigation. Young's modulus and the Poisson's ratio are given by $E = 3 \times 10^3$ MPa and $\nu = 0.25$, respectively.

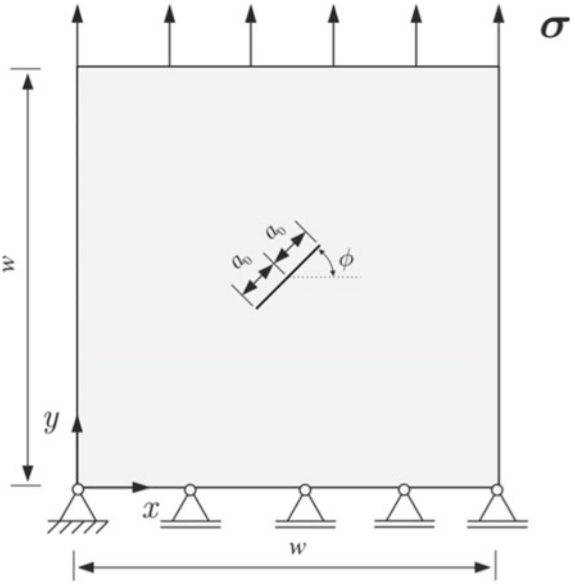


Fig. 10.15 Plate with angled-crack under tension. Geometry and boundary conditions

The specimen is discretized by a Voronoi mesh, as plotted in Fig. 10.16. Again, the linear, low order virtual elements are applied for the numerical simulations. Different crack orientations for the same spatial discretization are studied to illustrate the robustness of the proposed virtual element scheme. To achieve accurate results with the methodology described in Sect. 10.2.3, the mesh is refined around the cracked zone, see Fig. 10.16.

The results for the stress intensity factors are depicted in Table 10.1. The numerical stress intensity factors are compared for different crack angles ϕ with analytical results, see e.g. Sih et al. (1962). The analytical values follow from

$$K_{\text{I,ex.}} = \sigma\sqrt{a_0\pi}\ \cos^2(\phi) \quad \text{and} \quad K_{\text{II,ex.}} = \sigma\sqrt{a_0\pi}\ \sin(\phi)\cos(\phi), \qquad (10.39)$$

in terms of the tensile stress σ, the half crack length a_0 and the crack angle ϕ. The numerical simulation with virtual elements agrees very well with the analytical solution of Sih et al. (1962). This underlines the ability of the virtual element method to predict stress intensity factors based on the procedure in Sect. 10.2.3.

10.2.5 Propagation Criteria: Maximum Circumferential Stress Criterion

An equivalent stress intensity factor $K_{aq}(K_{\text{I}}, K_{\text{II}})$ has to be introduced in the mixed-mode loading cases as a measure for crack initiation, see e.g. Nuismer (1975). When

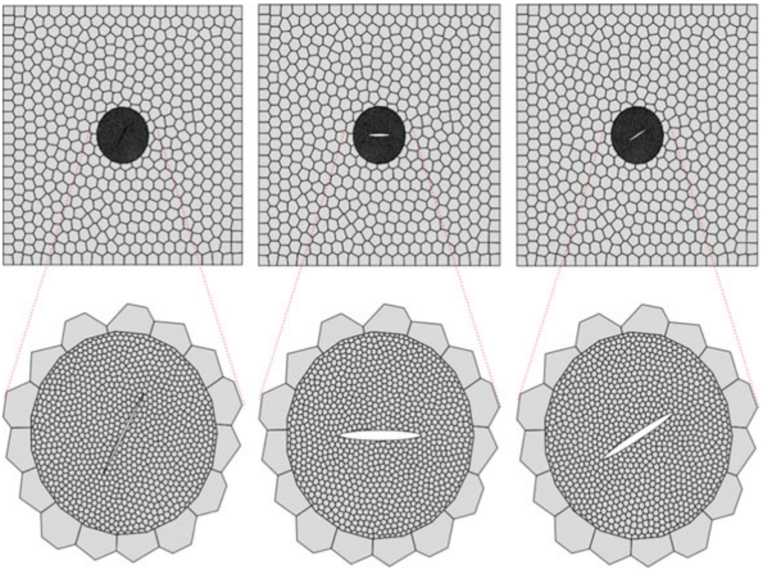


Fig. 10.16 Plate with angled-crack under tension. VEM mesh with different crack orientations: Left: $\phi = 60°$, middle: $\phi = 0°$ and right: $\phi = 30°$

Table 10.1 Comparison between numerical and analytical solutions of SIFs for different crack orientations ϕ

ϕ	K_{I}	$K_{\mathrm{I,ex.}}$	$K_{\mathrm{I}}/K_{\mathrm{I,ex.}}$	K_{II}	$K_{\mathrm{II,ex.}}$	$K_{\mathrm{II}}/K_{\mathrm{II,ex.}}$
0	2504.1	2506.6	0.9990	−0.9702	0.0000	–
10	2427.2	2431.0	0.9984	429.52	428.66	1.0020
20	2209.1	2213.4	0.9980	808.28	805.61	1.0033
30	1876.0	1878.0	0.9979	1089.8	1085.4	1.0041
40	1467.5	1471.0	0.9976	1239.5	1234.3	1.0042
50	1033.0	1035.7	0.9974	1240.5	1234.3	1.0050
60	624.84	626.66	0.9971	1091.9	1085.4	1.0060
70	291.92	293.22	0.9956	810.58	805.61	1.0062
80	74.830	75.584	0.9900	431.59	428.66	1.0068
90	0.0000	0.0000	–	0.0001	0.0000	–

the equivalent stress intensity factor K_{aq} exceeds a critical material parameter which is typically the critical fracture toughness $K_{\mathrm{I}c}$

$$K_{aq}(K_{\mathrm{I}}, K_{\mathrm{II}}) \geq K_{\mathrm{I}c} \tag{10.40}$$

the crack starts to propagate. This leads usually to a dynamical problem where the crack propagates with a certain velocity. Here we restrict the analysis to quasi-static crack growth where a certain crack length Δa is prescribed incrementally and the associated static problem is solved.

Once a crack has been initiated, the direction of growth can be predicted using different propagation criteria. In this section the maximum circumferential stress criterion is considered. Thus the stress state at a crack tip has to be evaluated. This leads to the circumferential stress $\sigma_{\theta\theta}$ and the shear stress $\tau_{r\theta}$ in the neighbourhood of the crack tip which, in case of linear elasticity, follow from an analytically solution with respect to the crack-tip field, see e.g. Gross and Seelig (2017),

$$\sigma_{\theta\theta} = \frac{1}{\sqrt{2\pi r}} \left[\frac{K_{\rm I}}{4} \left(3\cos\frac{\theta}{3} + \cos\frac{3\theta}{2} \right) + \frac{K_{\rm II}}{4} \left(-3\sin\frac{\theta}{2} - 3\sin\frac{3\theta}{2} \right) \right], \tag{10.41}$$

$$\tau_{r\theta} = \frac{1}{2\sqrt{2\pi r}} \cos\frac{\theta}{2} [K_{\rm I} \sin\theta + K_{\rm II}(3\cos\theta - 1)] \tag{10.42}$$

It is assumed that the crack will propagate from the crack tip in the direction of the maximum circumferential stress. The circumferential stress has its maximum for vanishing shear stress. Hence the shear stress in (10.42) is set to zero and after some re-arranging the propagation angle θ_c follows as a function of the stress intensity factors $K_{\rm I}$ and $K_{\rm II}$

$$\theta_{c,l} = -2 \arctan \left[\frac{2K_{\rm II}}{K_{\rm I} \left(1 + \sqrt{1 + 8\,(K_{\rm II}/K_{\rm I})^2} \right)} \right]. \tag{10.43}$$

10.2.6 Cutting Technique and Crack Update Algorithm

Growth and propagation of a crack needs special algorithmic treatment in which the crack path is constructed for a given loading situation. The main idea when using virtual element discretizations is the splitting of elements aligned in the direction of growth. In that way arbitrarily shaped elements with an arbitrary number of nodes will be produced, see Fig. 10.17. Such situations can only be handled by the virtual element method, being able to deal with any element shape.

The procedure of element splitting can be explained on the basis of the element depicted in Fig. 10.18 and the geometrical description in Fig. 10.19.

To visualize the algorithmic structure of the splitting procedure, a single virtual element Ω_2 is considered. It consists of the counter-clockwise ordered nodes and associated segments as shown in Fig. 10.18.

$$\text{nodes}\{N_4, N_5, N_6, N_9, N_{10}, N_{11}\}_{\Omega_2} \quad \text{and} \tag{10.44}$$

$$\text{segments}\ \{S_{4,5}, S_{5,6}, S_{6,9}, S_{9,10}, S_{10,11}, S_{11,4}\}_{\Omega_2}\,. \tag{10.45}$$

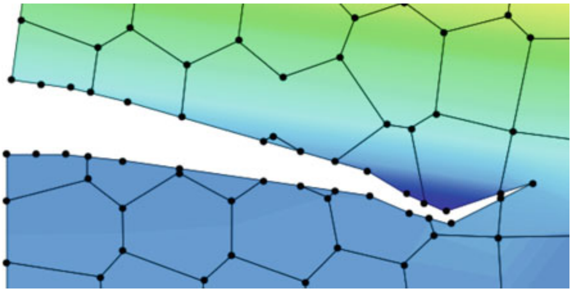


Fig. 10.17 Path of a crack through a virtual element discretization

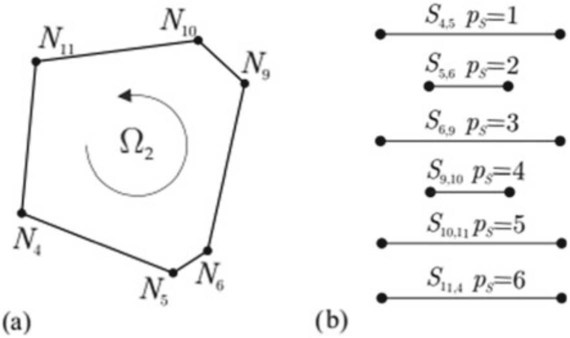


Fig. 10.18 **a** A virtual element Ω_2 composed of 6 nodes and **b** 6 element segments $\{S_{4,5}, \ldots, S_{11,4}\}_{\Omega_2}$ along with theirs position $p_S = 1, \ldots, 6$

As an example, the segment $S_{5,6}$ is composed of the second and the third local node $\{N_5, N_6\}_{\Omega_2}$.

Next we consider the whole domain, see Fig. 10.19. Here, the solid blue line on the left of Fig. 10.19a indicates a pre-existing crack. Furthermore the dashed red line represents a crack path. This crack path has to be inserted in the mesh. The crack path is defined by

1. the coordinates $\boldsymbol{X}_{1,tip}$, $\boldsymbol{X}_{2,tip}$ and $\boldsymbol{X}_{3,tip}$,
2. a given crack length Δa and
3. the angel $\theta_{c,I}$ between the crack extension and the local coordinate system x_1 which is predicted by (10.43).

At the beginning of the simulation, only the coordinates of the previous crack tip $\boldsymbol{X}_{1,tip}$ (node N_1) and the incremental length Δa of crack propagation are known. Now the coordinates of the new crack tip $\boldsymbol{X}_{2,tip}$ have to be determined. This is achieved by computing the stress intensity factors K_{I}, K_{II} and by evaluating Eq. (10.43) which yields the propagation angle $\theta_{c,l}$. Note that the angle $\theta_{c,l}$ is measured with respect to the local coordinate system x_1, x_2. Thus, a transformation to the global system X, Y is needed. The following expression defines the coordinates of the new crack tip $\boldsymbol{X}_{2,tip}$

$$\boldsymbol{X}_{2,tip} = \boldsymbol{X}_{1,tip} + \Delta a \left\{\cos\theta_{c,g} \, , \sin\theta_{c,g}\right\}^T . \tag{10.46}$$

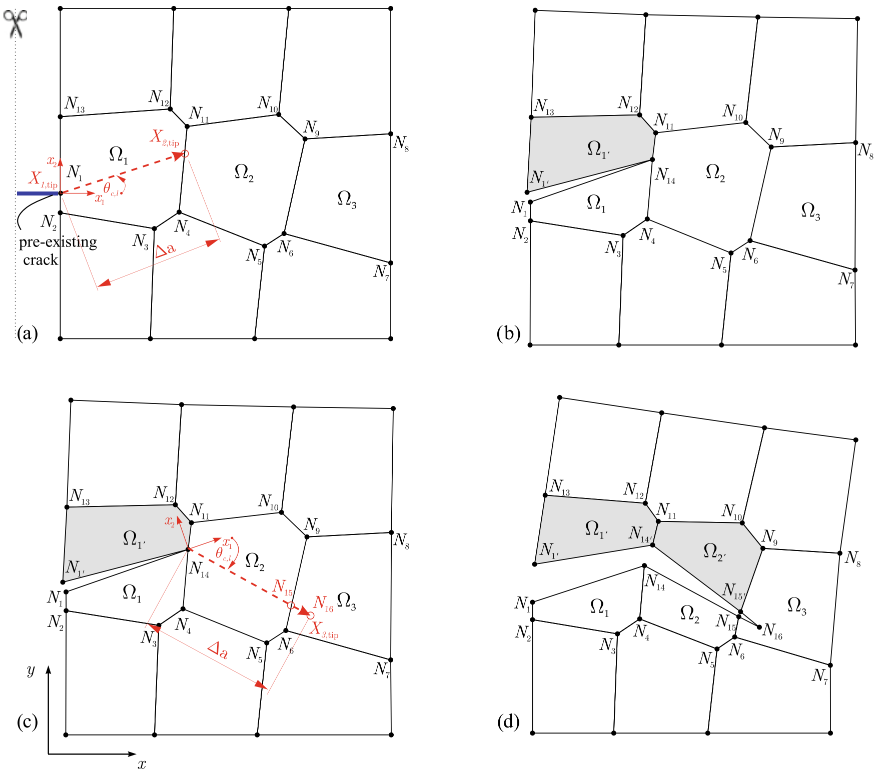


Fig. 10.19 A prescribed crack evolution using the introduced cutting technique procedure

with the global propagation angle $\theta_{c,g}$. This angle is computed from the previous crack angle $\theta_{prev,g}$ as

$$\theta_{c,g} = \theta_{c,l} + \theta_{prev,g} \tag{10.47}$$

Knowledge of the quantity $\boldsymbol{X}_{1,tip}$ (node N_1), allows access to the information of element Ω_1 in terms of:

$$\text{nodes}\{N_1, N_2, N_3, N_4, N_{11}, N_{12}, N_{13}\}_{\Omega_1} \quad \text{and} \tag{10.48}$$

$$\text{segments}\ \{S_{1,2}, S_{2,3}, S_{3,4}, S_{4,11}, S_{11,12}, S_{12,13}, S_{13,1}\}_{\Omega_1}\ . \tag{10.49}$$

From this information, the intersection between the segments of element Ω_1 and the new crack tip $X_{2,tip}$ can be determined by looping over all segments of the element Ω_1. Once the intercept segment $S_{4,11}$ is found, the position of the new node N_{14} (the new crack tip) in the set $\{N_1, \ldots, N_{13}\}_{\Omega_1}$ is known, see Fig. 10.19a and b. With this information the data structure of the element Ω_1 has to be updated:

$$\text{nodes}\left\{N_1, N_2, N_3, N_4, N_{14}, N_{11}, N_{12}, N_{13}\right\}_{\Omega_1} \quad \text{and} \tag{10.50}$$

$$\text{segments}\left\{S_{1,2}, S_{2,3}, S_{3,4}, S_{4,14}, S_{14,11}, S_{11,12}, S_{12,13}, S_{13,1}\right\}_{\Omega_1}. \tag{10.51}$$

For the crack propagation within the element Ω_1, the information of this element has to be spread in two groups. The first group belongs to the upper crack face, the second one is assigned to the lower crack face. The split of the information is obtained by executing two separate loops over the set of the nodes $\left\{N_1, \ldots, N_{13}\right\}_{\Omega_1}$. Hence the first group will be assigned to the newly introduced element $\Omega_{1'}$. Starting with the new crack tip (N_{14}), a loop over all nodes will be executed counterclockwise. It stops when the node N_1 of the previous crack tip is reached. Simultaneously, node N_1 is duplicated and replaced by $N_{1'}$. According to the new set $\left\{N_{14}, N_{11}, N_{12}, N_{13}, N_{1'}\right\}_{\Omega_{1'}}$, the segments are changed to $\left\{S_{14,11}, S_{11,12}, S_{12,13}, S_{13,1'}\right\}_{\Omega_{1'}}$, see Fig. 10.19b.

The information of the second group determines the remaining nodes of Ω_1. Starting with the previous crack tip (N_1) and ending with the new crack tip (N_{14}), the set $\left\{N_1, N_2, N_3, N_4, N_{14}\right\}_{\Omega_1}$ is obtained together with the segments $\left\{S_{1,2}, S_{2,3}, S_{3,4}, S_{4,14}, S_{14,1}\right\}_{\Omega_1}$, as depicted in Fig. 10.19b.

Finally the neighboring element to Ω_1 and $\Omega_{1'}$ has to be found. A search is needed for the element that shares the previous segment $S_{4,11}$. Finding the shared segment $S_{11,4}$ yields on one hand the neighboring element Ω_2 and on the other hand the position of the new node N_{14} (the new crack tip) within the set $\left\{N_4, \ldots, N_{11}\right\}_{\Omega_2}$. For example, the shared segment $S_{11,4}$ has the position $p_S = 6$ in the set of segments $\left\{S_{4,5}, \ldots, S_{11,4}\right\}_{\Omega_2}$. Hence the node N_{14} will be inserted into $\left\{N_4, \ldots, N_{11}\right\}_{\Omega_2}$ at the position $p_N = p_S + 1$. This step finalizes the first propagation of the crack by Δa in the direction of $\theta_{c,I}$. Further crack propagation steps are summarized in Fig. 10.19c and d and in Table 10.2, the latter provides a general description for the split algorithms.

10.2.7 Crack Propagation Simulations Based on the Cutting Technique

The efficiency and accuracy of the computational framework for crack propagation in isotropic brittle materials discussed above with the novel feature of a robust cutting technique will be demonstrated by means of some numerical examples. Hereby, the virtual element method confirms its flexibility in dealing with complex element shapes and arbitrary number of nodes in an element which are changed due to the cutting process during the simulation.

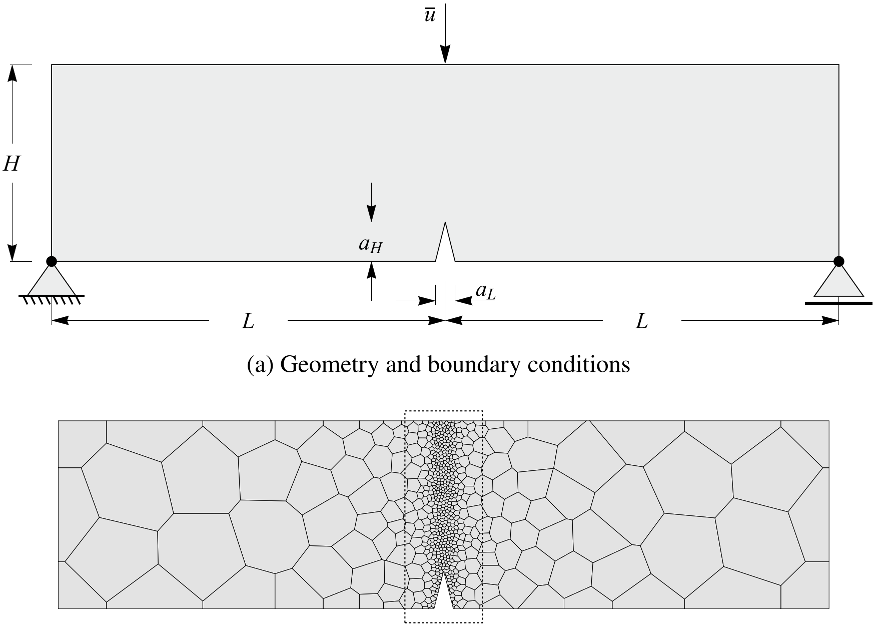
Three-point bending test. The crack propagation analysis based on the cutting algorithm described above can be investigated by means of a three point bending test, see e.g. Miehe and Gürses (2010). The crack propagation is analyzed in a beam with a pre-existing crack. The beam is simply supported and loaded in the middle on the

Table 10.2 A general description for the split algorithms

1. *Calculation of the new crack tip coordinates* $\boldsymbol{X}_{3,tip}$, see Fig. 10.19c
   a. Obtain K_{I}, K_{II} using (10.35)
   b. Compute $\theta_{c,l}$ using (10.43) and transform it to $\theta_{c,g}$
   c. Calculate $\boldsymbol{X}_{3,tip}$ similar to $\boldsymbol{X}_{2,tip}$ by using (10.46)
2. *Treatment of the crack tip elements*
   a. Find the elements having the previous crack tip node N_{14}**:** $\Rightarrow \{\Omega_{1'}, \Omega_1, \Omega_2\}_{\Omega_{tip}}$
   b. Find from $\{\Omega_{1'}, \Omega_1, \Omega_2\}_{\Omega_{tip}}$ the next element to be split **:** $\Rightarrow \Omega_2$
      i. Find the intercept segment with the prescribed crack path **:** $\Rightarrow S_{6,9}$, $p_S = 3$
      ii. Create a new node N_{15} at $p_S + 1 \Rightarrow \{N_4, N_5, N_6, N_{15}, N_9, N_{10}, N_{11}, N_{14}\}_{\Omega_2}$
3. *Splitting the information of* Ω_2 *in two groups*
   a. Construction of the upper crack face group
      i. Performing a loop over $\{N_4, \cdots, N_{14}\}_{\Omega_2}$, starting with N_{15} and ending with N_{14}**:** $\Rightarrow \{N_{15}, N_9, N_{10}, N_{11}, N_{14}\}_{\Box}$ and assign it to a new element **:** $\Rightarrow \Omega_{2'}$
   b. Construction of the lower crack face group
      i. Performing a loop over $\{N_4, \cdots, N_{14}\}_{\Omega_2}$, starting with N_{14} and ending with N_{15}**:** $\Rightarrow \{N_{14}, N_4, N_5, N_6, N_{15}\}_{\Box}$ and assign nodes to an existing element**:** $\Rightarrow \Omega_2$
4. *Treatment of the neighbouring elements to* Ω_2 *and* $\Omega_{2'}$
   a. Find the element that sharing the previous segment $S_{6,9}$**:** $\Rightarrow \Omega_3$, $S_{9,6}$, $p_S = 4$
   b. Insert node N_{15} at $p_S + 1$**:** $\Rightarrow \{N_6, N_7, N_8, N_9, N_{15}\}_{\Omega_3}$
   c. Since $X_{3,tip}$ lies inside Ω_3, we create a new node N_{16} with coordinate $X_{3,tip}$.
   d. Insert N_{16} and N_{15} at $p_S + 2$ and $p_S + 3$**:** $\Rightarrow \{N_6, N_7, N_8, N_9, N_{15}, N_{16}, N_{15}\}_{\Omega_3}$
5. *Cutting throught the mesh*, as plotted in Fig. 10.19d
   a. Find all top edges (including the previous top edge): $S_{tip} = \{S_{1',14}, S_{14,15}, S_{15,16}\}$
   b. Duplicate nodes in the edges, excluding endpoints: $\{N_{14}, N_{15}\} \Rightarrow \{N_{14'}, N_{15'}\}$
   c. Find all elements containing edges S_{tip}: $\{\Omega_{1'}, \Omega_{2'}, \Omega_3\}$, and update the segments in S_{tip} with the new nodes $\{N_{14'}, N_{15'}\}$
      e.g: $\{N_{15}, N_9, N_{10}, N_{11}, N_{14}\}_{\Omega_2'} \Rightarrow \{N_{15'}, N_9, N_{10}, N_{11}, N_{14'}\}_{\Omega_2'}$

upper flange. The geometry ($L = 4$ mm, $a_H = 0.4$ mm, $a_L = 0.2$ mm, $H = 2$ mm) and the loading ($\bar{u} = 0.04$mm) of the specimen are provided in Fig. 10.20a.

The material parameters used in the simulation are the Young's modulus and the Poisson's ratio given by $E = 2.08 \times 10^4$ MPa and $\nu = 0.3$, respectively. Plane strain conditions are assumed. The specimen is discretized using a Voronoi mesh with 621 virtual elements, as sketched in Fig. 10.20b.



(a) Geometry and boundary conditions

(b) Voronoi mesh

Fig. 10.20 Three-point bending test

Figure 10.21 demonstrates the crack path evolution along with the stress σ_{11} distribution for three different deformation stages up to final failure. Red and blue contour-plots represent the maximum tensile and compressive stresses, respectively. Tensile stresses are observed at the notch-tip when the beam is loaded. The crack is initiated in this tensile area and propagates straight to the load since the maximum tensile stress is in the middle of the beam. This results in a pure Mode-I failure. Starting with the Voronoi mesh of 621 virtual elements, see Fig. 10.21a, the crack path starts to develop with 632 elements. In Fig. 10.21b the crack starts to develop at the given displacement $\bar{u}_{c1}$. Due to the cutting procedure 650 elements are needed to follow the crack propagation up to $\bar{u}_{cm}$, Fig. 10.21c. At the final crack for the given displacement $\bar{u}_{cf}$ a mesh with 673 elements models the crack, see Fig. 10.21d. Note that only few additional virtual elements have to be generated within the cutting algorithm. This is sufficient to track the crack path accurately during the simulation.

The tensile test with two notches and holes relates to a problem that is in a more complex stress and strain state, see e.g. Miehe et al. (2007). It demonstrates the capability of virtual elements with flexible number of nodes and shapes to solve these type of crack problems. Some animal shaped elements are inserted in the mesh to account for non-convexity of the virtual elements.

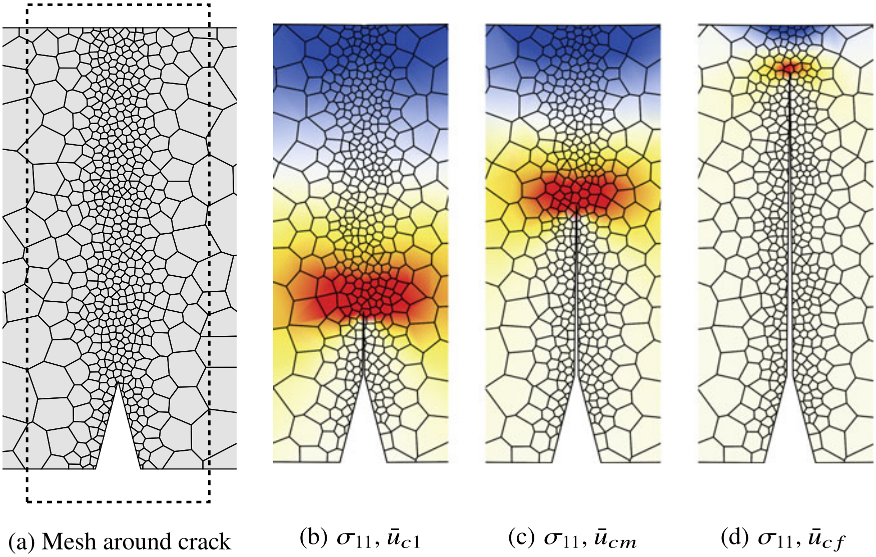
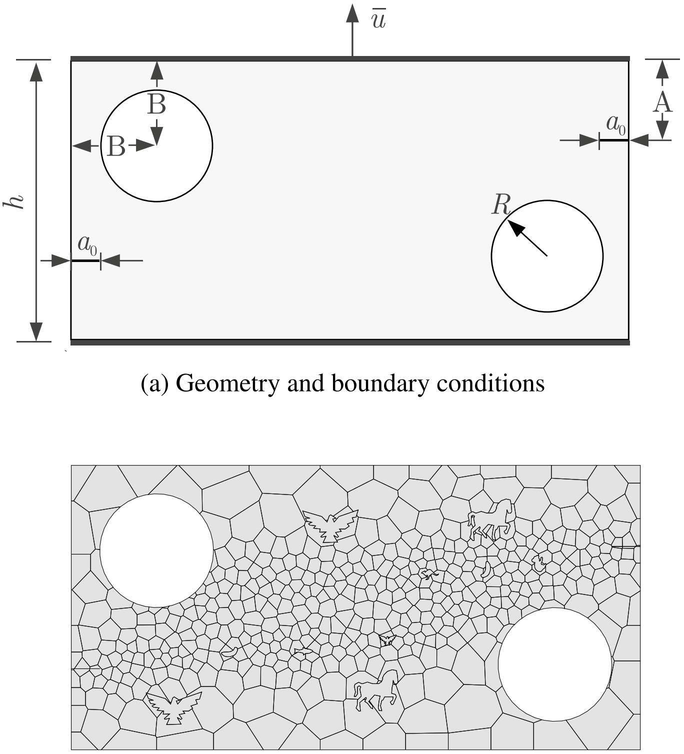

(a) Mesh around crack
(b) $\sigma_{11}, \bar{u}_{c1}$
(c) $\sigma_{11}, \bar{u}_{cm}$
(d) $\sigma_{11}, \bar{u}_{cf}$

Fig. 10.21 Three-point bending test—Contour plots of the stress σ_{11} during crack evolution

This tensile test illustrates crack-initiation and curved-crack-propagation. The geometrical setup ($h = 10$ mm, $R = 2$ mm, $a_0 = 1$ mm, $A = 2.85$ mm, $B = 3$ mm), the loading by a prescribed displacement ($\bar{u} = 0.1$mm) of the specimen, see Fig. 10.22a. The discretization with Voronoi cells is sketched Fig. 10.22b. Again, the low order virtual element formulation is employed. The mesh includes various shaped Voronoi cells around the fractured zones including animal shaped elements in order to demonstrate the capability of the cutting algorithm to deal with any shape of virtual elements. The material parameters used in the simulation are Young's modulus and the Poisson's ratio given by $E = 2 \times 10^5$ MPa and $\nu = 0.3$, respectively. Initial cracks are inserted at the lower left and upper right edge, see Fig. 10.22a. The specimen is loaded by the prescribed vertical displacement $\bar{u}$ at the upper edge while the lower edge is fixed in vertical direction.

Figure 10.23 depicts the crack path evolution and the distribution of the shear stress σ_{12} for different deformation stages. The cracks initiate at the tip of the two notches at time t_0. Thereafter, the left and the right cracks tend to propagate towards the holes as shown in Fig. 10.23b for t_1. Once the cracks passed the holes at time t_2, see Fig. 10.23c, they continue to propagates horizontally until final failure in Fig. 10.23d at time t_3 when they reach the hole opposite to there starting position. The final failure is related to mixed-mode fracture.



(a) Geometry and boundary conditions

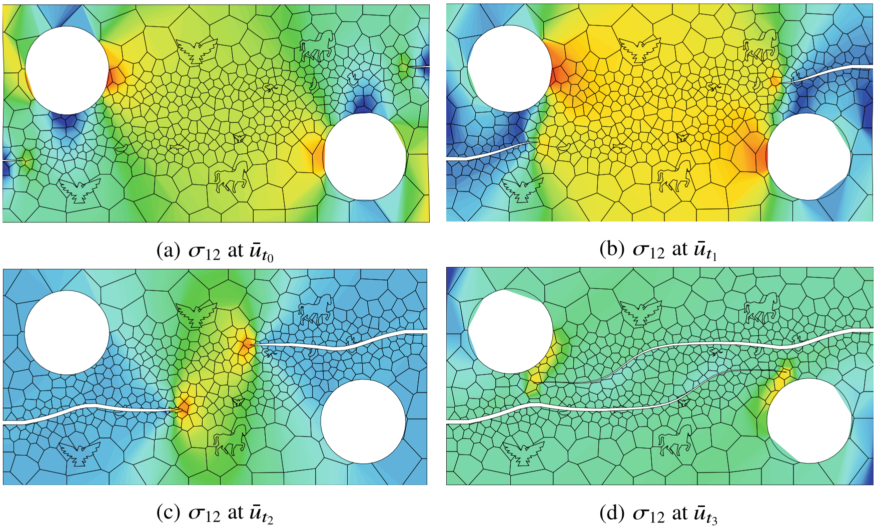
(b) Voronoi mesh

Fig. 10.22 Tensile test with two notches and holes

All examples in this section demonstrate that the virtual element method is very efficient and robust for applications in linear fracture mechanics. When inelastic effects have to be considered the criteria based on the analytical solutions in (10.41) and (10.43) are no longer valid. Then the approach described in Sect. 10.4 has to be applied which allows to include inelastic effects and is very efficient as well.

10.3 Phase Field Methods for Brittle Fracture Using Virtual Elements

Instead of sharp cracks, one can introduce a zone in the solid in which the resistance of the material is very low such that a crack can be approximated. One of the possibilities was discussed in Sect. 10.1 using damage mechanics. Here we formulate a smoothed crack field by employing the phase field method to model crack propagation in the


(a) σ_{12} at $\bar{u}_{t_0}$

(b) σ_{12} at $\bar{u}_{t_1}$

(c) σ_{12} at $\bar{u}_{t_2}$

(d) σ_{12} at $\bar{u}_{t_3}$

Fig. 10.23 Tensile test of the notched specimen with two holes—Crack trajectory for different deformation stages

situation of brittle fracture. This methodology allows to capture the progression of cracks in different materials which are prone to brittle failure, like concrete, glass and ceramics.

10.3.1 Governing Equations for Elasticity

The response of the solid is described at a material point $\boldsymbol{X} \in \Omega$ and time $t \in \mathcal{T} = [0, T]$ by the displacement field $\boldsymbol{u}(\boldsymbol{X}, t)$, as shown in Fig. 2.2. Small deformations are assumed which leads to the symmetric strain tensor $\boldsymbol{\varepsilon} = \nabla_s \boldsymbol{u}$, see (2.9), where the short notation $\boldsymbol{u}(\boldsymbol{X}, t) = \boldsymbol{u}$ was used.

In brittle fracture it is sufficient to describe the constitutive behaviour of the solid under small strain conditions by a homogeneous isotropic linear elastic material. The related strain energy density function $\Psi = \Lambda \, / \, 2(\mathrm{tr}(\boldsymbol{\varepsilon}))^2 + \mu \, \mathrm{tr}(\boldsymbol{\varepsilon} \cdot \boldsymbol{\varepsilon})$, with the Lamé constants Λ and μ was provided in (2.43).

The variational formulation of the equilibrium equation (10.27) provides again the basis for numerical solution schemes. The principle of stationary elastic potential $U(\boldsymbol{u})$ is chosen as basis for the development of the method. Here it is written for the small strain case, using the above defined strain energy density function Ψ with the strain tensor $\boldsymbol{\varepsilon}$

$$U(\boldsymbol{u}) = \int\limits_{\Omega} \Psi(\boldsymbol{\varepsilon})\, \mathrm{d}\Omega - \int\limits_{\Omega} \bar{\boldsymbol{f}} \cdot \boldsymbol{u}\, \mathrm{d}\Omega - \int\limits_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \boldsymbol{u}\, \mathrm{d}\Gamma. \tag{10.52}$$

By minimizing the potential $U(\boldsymbol{u})$ the displacement $\boldsymbol{u}$ can be computed for a given volume loading $\bar{\boldsymbol{f}}$ and traction loading $\bar{\boldsymbol{t}}$.

10.3.2 Regularization of a Sharp Crack Topology

Contrary to the approach in Sects. 10.2.1 and 10.2.5 the crack is not modeled as a sharp crack but regularized using the phase field method. A model problem illustrates this approach where a sharp crack in the left image of Fig. 10.24 is described by a new field variable $d(\boldsymbol{X}, t) \in [0, 1]$, where $d(\boldsymbol{X}, t) = 0$ characterizes the unbroken and $d(\boldsymbol{X}, t) = 1$ the fully broken state of the material. This variable d is called the crack phase field and can be linked to the continuum theory of damage, where a scalar damage field describes the development of micro-cracks in a solid, see Sect. 10.1.

Following Miehe et al. (2010a) the non-smooth phase field can be approximated over the crack surface by the exponential function

$$d(X) = e^{-\frac{|X|}{l}} \tag{10.53}$$

in terms of the local coordinate X, outlined in Fig. 10.24. As can be seen in the right image, the regularization depends on the length scale parameter l and yields for $l \to 0$ the sharp crack topology shown in the left image of Fig. 10.24. The exponential function in (10.53) is designed such that $d(0) = 1$ and $d(\pm\infty) = 0$. It is the solution of the ordinary differential equation

$$l^2 d''(X) - d(X) = 0 \tag{10.54}$$

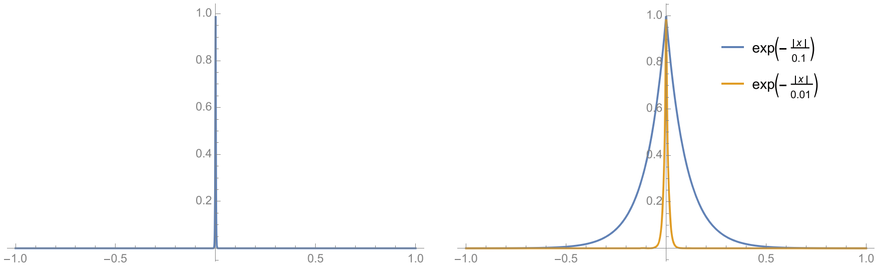


Fig. 10.24 Description of a crack—sharp crack topology (left), and regularized crack topology (right) using different length parameters $l = 0.1$ and $l = 0.01$

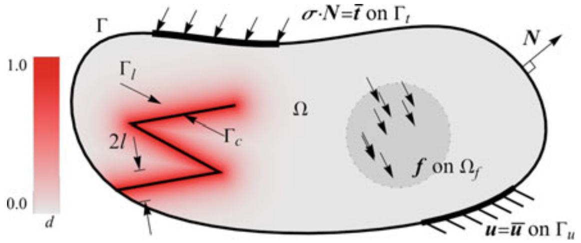

Γ
$\sigma \cdot N = \bar{t}$ on Γ_t
N
1.0
0.0
d
Γ_l
Ω
$2l$
Γ_c
f on Ω_f
$u = \bar{u}$ on Γ_u

Fig. 10.25 Solid with a regularized crack and boundary conditions

subjected to the boundary conditions above. A variational functional can be associated with the above differential equation. It is defined as

$$\Gamma_l(d) = \frac{1}{2l} \int_x \left[d^2 + l^2 d'^2 \right] \mathrm{d}X. \tag{10.55}$$

Like in damage mechanics the phase field can only grow: $\dot{d} \geq 0$. This is equivalent to the fact that a cracked body cannot be reversed under normal circumstances leading to a body without cracks. The extension of this idea to a two- or three-dimensional problem follows by introducing a regularization of the sharp crack topology $\Gamma_c \to \Gamma_l$ by the a crack surface density function as outlined in Miehe et al. (2010a)

$$\Gamma_l(d) = \int_\Omega \gamma(d, \nabla d)\, \mathrm{d}\Omega \quad \text{with} \quad \gamma(d, \nabla d) = \frac{1}{2l}(d^2 + l^2 |\nabla d|^2). \tag{10.56}$$

Hence the crack is modeled by the crack surface density function γ per unit volume of the solid and the fracture length scale parameter l that governs the width of the diffuse crack, as shown in Fig. 10.25. Note, that the phase field has to be introduced in addition to the displacement field $\boldsymbol{u}(\boldsymbol{X}, t)$.

The regularization function γ in (10.56) was introduced first by Bourdin et al. (2000). However, function γ already appears in the approximation by Ambrosio and Tortorelli (1990) for the functional of image segmentation of Mumford and Shah (1989). In addition, Borden et al. (2014) suggested the following higher order approximation

$$\gamma(d, \nabla d) = \frac{1}{4l}[d^2 + 2l^2 |\nabla d|^2 + l^4 (\Delta d)^2]. \tag{10.57}$$

Based upon the above arguments a given sharp crack topology Γ_c can be modeled by the regularized crack phase-field d which is obtained from the minimization principle of the diffusive crack topology Γ_l

$$d = \min_d \{\Gamma_l(d)\} \quad \text{with} \quad d = 1 \text{ on } \Gamma_c \in \Omega, \tag{10.58}$$

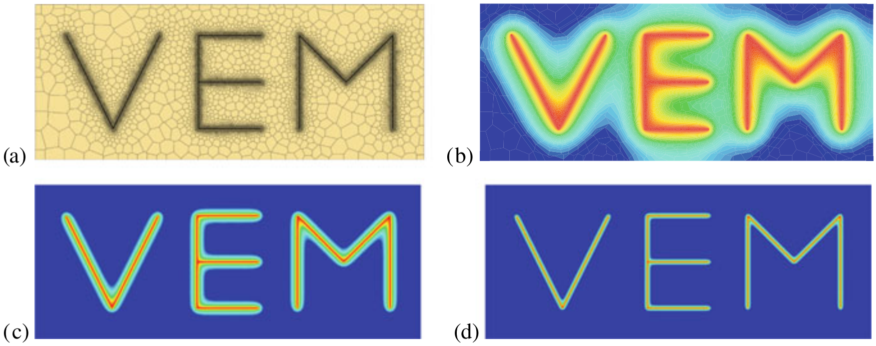


Fig. 10.26 A purely geometric approach to phase-field fracture based on virtual element method. Solutions or different fracture length scales $l_b > l_c > l_d$ are depicted in (**b**) to (**d**), according to Aldakheel et al. (2018a)

yielding the Euler equation, which fulfills the Neumann-type boundary condition

$$d - l^2 \Delta d = 0 \quad \text{in } \Omega \quad \text{and} \quad \nabla d \cdot \boldsymbol{N} = 0 \quad \text{on } \partial\Omega, \tag{10.59}$$

where Δ is the Laplacian and $\boldsymbol{N}$ the outward normal vector of the boundary Γ.

As an example, a rectangular area Ω with given letters "VEM" denoting a sharp crack Γ_c is investigated, see Fig. 10.26. The minimization problem (10.58) for the phase field is solved on this domain with the prescribed Dirichlet boundary condition $d = 1$ on $\Gamma_c \in \Omega$. Geometry and discretization of the specimen using a Voronoi mesh can be found in Fig. 10.26a. The formulation is based on a virtual element discretization with linear ansatz. In the limit the above variational principle results for $l \rightarrow 0$ in a sharp crack surface Γ_c, as plotted in Fig. 10.26d for the specimen with a Voronoi mesh. The computational results with three different length scale parameters l allow the following observations:

- The length scale l, which is user defined, has to be selected with care. A too large value does not result in a sharp crack, see Fig. 10.26b. Hence, the correct fracture behaviour of a specimen will not be captured.
- If a very small length l is selected then a very fine mesh is needed to resolve this phase field, see Fig. 10.26d.

Thus, an efficient treatment of fracture with the phase field method needs a length scale parameter l that on one hand approximates the solution correctly and on the other hand is large enough such that the mesh density is minimal.

10.3.3 Variational Formulation of Brittle Fracture

In the following, we outline a variational approach to brittle fracture in elastic solids at small strains when using the phase field approach. Francfort and Marigo (1998) have demonstrated that for a given crack set Γ_c the total energy of the body can be formulated as

$$\Psi(\boldsymbol{\varepsilon}, \Gamma_c) = \Psi_s(\boldsymbol{\varepsilon}) + \Psi_c(\Gamma_c), \tag{10.60}$$

with the elastic strain energy Ψ_s and the fracture energy Ψ_c. The latter is needed to create the crack surface. According to the Griffith's theory, the fracture energy can be expressed in variational form as

$$\Psi_c(\Gamma_c) = \int\limits_{\Gamma_c} G_c \, \mathrm{d}\Gamma, \tag{10.61}$$

with the critical fracture energy density G_c, also referred to as the critical energy release rate or the fracture toughness in the context of cohesive zone models, as outlined in the work by Borden et al. (2014). By using Eqs. (10.60)–(10.61), the total potential energy of the body is given by

$$\Psi(\boldsymbol{\varepsilon}, \Gamma_c) = \int\limits_{\Omega} \psi(\boldsymbol{\varepsilon}) \, \mathrm{d}\Omega + \int\limits_{\Gamma_c} G_c \, \mathrm{d}\Gamma. \tag{10.62}$$

This function contains the unknown sharp crack topology Γ_c, which makes minimizing this function very difficult. Bourdin et al. (2000) suggested the approximation of the fracture energy $\Psi_c(\Gamma_c)$ by using a regularized formulation

$$\Psi_c(\Gamma_c) = \int\limits_{\Gamma_c} G_c \, \mathrm{d}\Gamma \approx \int\limits_{\Omega} G_c \gamma(d, \nabla d) \, \mathrm{d}\Omega, \tag{10.63}$$

with the regularization function γ. This function was renamed by Miehe et al. (2010b) as crack surface density function. It contains the phase field parameter d and the fracture length scale l, as can be seen from Eq. (10.56). Due to the regularization of the sharp crack Γ_c, the stored strain energy ψ is now expressed as

$$\Psi_s(\boldsymbol{\varepsilon}, d) = \int\limits_{\Omega} g(d) \; \psi(\boldsymbol{\varepsilon}) \, \mathrm{d}\Omega, \quad \text{with} \quad g(d) = (1 - d)^2, \tag{10.64}$$

where $g(d)$ is a degradation function that models the degradation of the stored elastic energy of the solid due to fracture. It interpolates between the unbroken response for $d = 0$ and the fully broken state at $d = 1$ by satisfying the constraints

$$g(0) = 1 \quad , \quad g(1) = 0 \quad , \quad g'(1) = 0. \tag{10.65}$$

Experimental observations show that brittle material cracks in tension states. In order to enforce crack evolution only in tension, the strain energy density function ψ of the solid is additively decomposed into a positive part ψ^+ due to tension and a negative part ψ^- due to compression, see Miehe et al. (2010b),

$$\psi = g(d)\;\psi^+ + \psi^- \quad \text{with} \quad \psi^\pm = \frac{\Lambda}{2}\langle \operatorname{tr}(\boldsymbol{\varepsilon})\rangle_\pm^2 + \mu \operatorname{tr}((\boldsymbol{\varepsilon}_\pm)^2). \tag{10.66}$$

Here the bracket operators $\langle \bullet \rangle_+ = (\bullet + |\bullet|)/2$ and $\langle \bullet \rangle_- = (\bullet - |\bullet|)/2$ were introduced. The strain tensor $\boldsymbol{\varepsilon}$ is divided into a positive $\boldsymbol{\varepsilon}_+$ and a negative $\boldsymbol{\varepsilon}_-$ part

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_+ + \boldsymbol{\varepsilon}_- \quad \text{with} \quad \boldsymbol{\varepsilon}_+ = \sum_{i=1}^{3} \langle \varepsilon_i \rangle_+ \boldsymbol{N}_i \otimes \boldsymbol{N}_i \; , \; \boldsymbol{\varepsilon}_- = \sum_{i=1}^{3} \langle \varepsilon_i \rangle_- \boldsymbol{N}_i \otimes \boldsymbol{N}_i \; , \tag{10.67}$$

where ε_i are the principal strains and $\boldsymbol{N}_i$ the associated principal strain directions for $i = 1..3$. With these expressions at hand, the stress tensor $\boldsymbol{\sigma}$ can be obtained as

$$\boldsymbol{\sigma} = g(d)\frac{\partial \psi(\boldsymbol{\varepsilon})}{\partial \boldsymbol{\varepsilon}} = g(d)\frac{\partial \psi^+}{\partial \boldsymbol{\varepsilon}} + \frac{\partial \psi^-}{\partial \boldsymbol{\varepsilon}}. \tag{10.68}$$

With all above introduced equations, the total potential ψ_{tot} of the phase field problem can be written as a sum of the elastic and the fracture energy

$$\psi_{tot}(\boldsymbol{\varepsilon}, d, \nabla d) = \psi_{elas}(\boldsymbol{\varepsilon}, d) \; + \; \psi_{frac}(d, \nabla d) = \underbrace{g(d)\psi^+ + \psi^-}_{\text{elastic energy}} + \underbrace{G_c \gamma(d, \nabla d)}_{\text{fracture energy}}. \tag{10.69}$$

To ensure crack irreversibility in the sense that the cracks can only grow (i.e. $\dot{d} \geq 0$), Miehe et al. (2010a) introduced a history-field variable $\mathcal{H}$ as a crack driving force, which depends on the maximum positive strain energy ψ^+

$$\mathcal{H}(\boldsymbol{X}, t) = \max_{s \in [0,t]} \psi^+\big(\boldsymbol{\varepsilon}(\boldsymbol{X}, s)\big). \tag{10.70}$$

The evolution of the regularized crack surface functional (10.56) is driven by the constitutive functions. This can be formulated by postulating a global evolution equation of regularized crack surface as

$$\frac{d}{dt}\Gamma_l(d) = \int_\Omega \frac{\partial \gamma_l(d, \nabla d)}{\partial d}\, \dot{d}\, \mathrm{d}\Omega = \frac{1}{l}\int_\Omega [\; (1-d)\mathcal{H} - \eta \dot{d} \;]\, \dot{d}\, \mathrm{d}\Omega \geq 0 \tag{10.71}$$

where $\eta \geq 0$ is a material parameter that characterizes the viscosity of the crack propagation. This evolution statement provides the local equation for the evolution of the crack phase-field in the domain Ω along with its homogeneous Neumann boundary condition as

$$\eta \dot{d} = (1-d)\mathcal{H} - [\, d - l^2 \Delta d \,] \qquad \text{with} \qquad \nabla d \cdot N = 0 \quad \text{on} \quad \Gamma \tag{10.72}$$

Based on this local form of the crack phase-field evolution the fracture energy can be redefined as

$$\psi_{frac}(d, \nabla d, \mathcal{H}) = G_c \gamma_l(d, \nabla d) + \frac{\eta}{2\Delta t}(d - d_n)^2 + g(d)\mathcal{H}, \tag{10.73}$$

where a backward Euler scheme is introduced for the time discretization of $\dot{d}$ leading to $\dot{d} = (d - d_n) \,/\, \Delta t$ with $\Delta t = t - t_n > 0$ being the time step which is employed to solve the fracture problem. The material parameter $\eta \geq 0$ characterizes the viscosity of the crack propagation.

The above introduced variables characterize the brittle failure response of a solid. The response is based on two global primary fields, the displacement field $\boldsymbol{u}$ and the crack phase-field d, that are summarized as global primary field variables in $\mathfrak{U} = \{\boldsymbol{u}, d\}$. The subsequent modeling of the phase field approach for brittle fracture is based on a set of constitutive state variables $\mathfrak{C} = \{\boldsymbol{\varepsilon}, d, \nabla d, \mathcal{H}\}$.

The development of virtual elements that handle phase field brittle fracture in elastic solids can start from a pseudo potential density functional instead of using the weak form. This has advantages when the code is automatically generated using the software tool *AceGen*, see Korelc and Wriggers (2016). The specific pseudo potential density functional can then be written as

$$\psi_{tot}(\mathfrak{C}) = g(d)\psi^+ + \psi^- + G_c \gamma_l(d, \nabla d) + \frac{\eta}{2\Delta t}(d - d_n)^2 + (1-d)^2\mathcal{H} \tag{10.74}$$

which leads to a total pseudo potential

$$\Pi(\mathfrak{U}) = \int_\Omega \psi_{tot}(\mathfrak{C})\, \mathrm{d}\Omega \; - \; \Pi_{\text{ex}}(\boldsymbol{u}), \tag{10.75}$$

with the external load $\Pi_{\text{ext}}(\boldsymbol{u})$. We note, that the crack driving force $\mathcal{H}$ and the degradation function $g(d)$ have to be kept constant during the derivation procedure of residual to obtain the correct weak form of the problem.

10.3.4 Formulation of the Virtual Element Method

This section provides a summary for the formulation of the virtual element method including the crack phase field d. The virtual element method relies on the split of the ansatz space $\mathfrak{U}_h = \{\boldsymbol{u}_h \,, d_h\}$ into a part $\mathfrak{U}_\pi$ representing the projected primary fields and a remainder

$$\mathfrak{U}_h = \mathfrak{U}_\pi + (\mathfrak{U}_h - \mathfrak{U}_\pi). \tag{10.76}$$

The projected primary field $\mathfrak{U}_\pi$ is approximated by using a linear ansatz function[2] with respect to the coordinate $\boldsymbol{X} = \{X, Y\}$. With this function, the nodal values can be projected on to the element area, in the case of two-dimensional space. The projection $\mathfrak{U}_\pi$ is defined at element level by a polynomial ansatz function $\mathbf{N}_\pi$ as

$$\mathfrak{U}_\pi = \begin{Bmatrix} u_{\pi x} \\ u_{\pi y} \\ d_\pi \end{Bmatrix} = \mathbf{a} \cdot \mathbf{N}_\pi = \begin{bmatrix} a_1 \; a_4 \; a_7 \\ a_2 \; a_5 \; a_8 \\ a_3 \; a_6 \; a_9 \end{bmatrix} \begin{Bmatrix} 1 \\ X \\ Y \end{Bmatrix}, \tag{10.77}$$

with the projected variables for the displacements $u_{\pi x}$, $u_{\pi y}$ and the phase-field d_π, the unknown parameter set $\mathbf{a}$ and the ansatz function $\mathbf{N}_\pi$.

The first step to construct the method is to obtain the unknown parameter set $\mathbf{a}$ in (10.77). In case of a linear polynomial ansatz function, the most simplest and efficient way is to use (3.60) and (3.62) to compute the projection. This procedure is based on tensor notation and determines the projection $\mathfrak{U}_\pi$ for a linear ansatz function for a virtual element Ω_v, see Sect. 3.1.5,

$$\nabla \mathfrak{U}_\pi = \frac{1}{\Omega_v} \int\limits_{\Omega_v} \nabla\, \mathfrak{U}_h \; \mathrm{d}\Omega = \frac{1}{\Omega_v} \int_{\Gamma_v} \mathfrak{U}_h \otimes \boldsymbol{N} \;\; \Gamma = \mathbb{D}_{\nabla}^{(2,1)}\, \mathfrak{U}_v. \tag{10.78}$$

Here $\boldsymbol{N}$ is the normal at the boundary Γ_v of the domain Ω_v. Furthermore, $\mathbb{D}_{\nabla}^{(2,1)}$, see e.g. (3.51), is an operator that links the projected gradient to the nodal degrees of freedom $\mathfrak{U}_v$.[3] The set of nodal degrees of freedom of a virtual element Ω_v are defined as

$$\mathfrak{U}_v = \left\langle \mathfrak{U}_1 \, \mathfrak{U}_2 \, \ldots \, \mathfrak{U}_{n_V} \right\rangle^T \tag{10.79}$$

where the nodal degrees of freedom at node K are given by $\mathfrak{U}_K = \left\langle u_{Kx} \, u_{Ky} \, d_K \right\rangle^T$.

The projection in (10.78) does not determine the ansatz $\mathfrak{U}_\pi$ in (10.77) completely and has to be supplemented by a further condition to obtain all constants $\mathbf{a}$. For this purpose we adopt the condition that equalizes the sum of the nodal values of $\mathfrak{U}_h$ and the nodal values of its projection $\mathfrak{U}_\pi$. This condition yields for each element Ω_v

$$\sum_{K=1}^{n_V} \mathfrak{U}_\pi(\mathbf{X}_K) = \sum_{K=1}^{n_V} \mathfrak{U}_h(\mathbf{X}_K), \tag{10.80}$$

where $\mathbf{X}_K$ are the coordinates of the nodal point K and the sum includes all n_V boundary nodes. Using (3.62) it follows

[2] Higher order polynomial degrees are possible and have to be developed along the lines presented in Chap. 5.

[3] The operator $\mathbb{D}_{\nabla}^{(2,1)}$ does not have to be computed explicitly when using automatic software generation as in Sect. 3.1.5, see also (3.51).

$$\begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \sum_{K=1}^{n_V} [\mathfrak{U}_K - \nabla \mathfrak{U}_\pi \, \mathbf{X}_K] \tag{10.81}$$

The two conditions (10.78) and (10.81) yield all unknowns **a** of the projection $\mathfrak{U}_\pi$ in terms of the nodal values

$$\mathfrak{U}_\pi = \mathbf{H}^{(2,1)}(X, Y) \, \mathbb{D}_{\mathfrak{U}}^{(2,1)} \, \mathfrak{U}_v . \tag{10.82}$$

The development of virtual elements that handle phase-field brittle fracture in elastic solids can start from the pseudo potential density functional (10.75). The potential function is split into a consistency term $\Pi_c(\mathfrak{U}_\pi)$, that depends on the projection $\mathfrak{U}_\pi$ and an associated stabilization term $\Pi_{stab}(\mathfrak{U}_h - \mathfrak{U}_\pi)$ in terms of the reminder in (10.76). By exploiting the split in (10.76), the pseudo potential function defined in (10.75) can be rewritten by assembling all element contributions for the n_v virtual elements as

$$\Pi(\mathfrak{U}) = \mathop{\mathbb{A}}_{v=1}^{n_v} [\Pi_c(\mathfrak{U}_\pi)|_v + \Pi_{stab}(\mathfrak{U}_h - \mathfrak{U}_\pi)|_v] \,, \tag{10.83}$$

where v denotes a single virtual element within the discretization and the label $\Box|_v$ represents the quantities for an element v.

The primary field variables $\mathfrak{U}_\pi$ are linear functions and their gradient $\nabla \mathfrak{U}_\pi$ is constant over the domain of the virtual element Ω_v. As a consequence, the potential $\psi_{tot}(\mathfrak{C}_\pi)$ is integrated by evaluating the function at the element centroid $\mathbf{X}_c$ and multiplying it with domain size Ω_v. By neglecting the body and traction forces, the consistency term $\Pi_c(\mathfrak{U}_\pi)$ in (10.83) can be obtained for every virtual element as

$$\Pi_c(\mathfrak{U}_\pi)|_v = \int_{\Omega_v} \psi_{tot}(\mathfrak{C}_\pi) \, \mathrm{d}\Omega = \psi_{tot}(\mathfrak{C}_\pi)|_{v,c} \, \Omega_v , \tag{10.84}$$

where Ω_v refers to the element area. In this equation, the label $\Box|_{v,c}$ denotes element quantities that are evaluated at the element centroid $\mathbf{X}_c$.

It remains to provide a way to compute the second part $\Pi_{stab}(\mathfrak{U}_h - \mathfrak{U}_\pi)$ in (10.83). Again, the energy stabilization, see e.g. Sect. 6.1.3, is applied, leading to an approximation of the primary fields $\mathfrak{U}$ in (10.76) by subdividing every virtual element into n_T triangular linear finite elements. Thus, the stabilization term follows as

$$\Pi_{stab}(\mathfrak{U}_h - \mathfrak{U}_\pi)|_v = \beta \left(\sum_{i=1}^{n_T} \Omega_v^i \; \psi_{tot}(\mathfrak{C}_h)|_{v,c} - \psi_{tot}(\mathfrak{C}_\pi)|_{v,c} \, \Omega_v \right) , \tag{10.85}$$

where $\psi_{tot}(\mathfrak{C}_h)|_{v,c}$ is the total potential function evaluated at the centroid $\mathbf{X}_c^i$ of one triangular element and Ω_v^i is the area of the i th triangle in the element Ω_v. The quantity β can be taken from the interval $0 < \beta \leq 1$. Choices for β are discussed in the next section.[4]

By using the software tool *AceGen*, the residual $\mathbf{R}_v$ and the stiffness matrix $\mathbf{K}_v$ of the virtual element can be derived as

$$\mathbf{R}_v = \left. \frac{\partial \Pi(\mathfrak{U})|_v}{\partial \mathfrak{U}_v} \right|_{g(d)=const.\ \text{and}\ \mathcal{H}=const.} \quad \text{and} \quad \mathbf{K}_v = \frac{\partial \mathbf{R}_v}{\partial \mathfrak{U}_v}, \tag{10.86}$$

where $\mathfrak{U}_v$ are the unknown primary fields (degrees of freedom) of the virtual element Ω_v, see (10.79). Finally, the global residual $\mathbf{R}$ and stiffness matrix $\mathbf{K}$ are formulated by assembling all individual element contributions for all elements n_v of the domain

$$\mathbf{R} = \overset{n_v}{\underset{v=1}{\mathbb{A}}} \mathbf{R}_v \quad \text{and} \quad \mathbf{K} = \overset{n_v}{\underset{v=1}{\mathbb{A}}} \mathbf{K}_v. \tag{10.87}$$

10.3.5 Numerical Examples for Brittle Fracture Using Phase Field

The performance of the low order virtual element formulation for phase-field modeling of brittle fracture will be demonstrated by means of two representative numerical examples.

The analysis of a crack in a Bi-Material plate models fracture phenomena of a bi-material specimen under tensile loading. The aim is to demonstrate crack phase-field initiation and branching within crack propagation. The geometrical setup and the loading conditions of the specimen are depicted in Fig. 10.27b. The specimen has a length and width of $2\,L$ with $L = 50$ mm and the diameter of the notch is $D = 10$ mm.

The right edge of the plate is fixed in x-direction and a vertical, increasing displacement is applied at the top and bottom edges until final failure. Material B is stiffer than material A. Material B is purely elastic and will not fracture. Young's modulus is chosen for material A as $E^A = 100$ kN/mm^2 and for material B as $E^B = 200$ kN/mm^2, Poisson's ratio is set to $\nu = 0.2$ for both materials. The viscosity of the crack propagation is given by $\eta^A = 10^{-6}$ kNs/mm^2, the critical energy release rate for material A has the value $g_c^A = 10^{-4}$ kN/mm and the stabilization parameter of the virtual element formulation was set set to $\beta = 0.4$.

Different mesh types (triangular mesh with six nodes (VE-T2) and Voronoi mesh) are studied to illustrate the robustness of the proposed virtual element scheme with linear ansatz order. These are compared to results obtained with a quadratic T2 triangular finite element solution. Mesh refinement is applied in the expected fracture zone, see Fig. 10.27a and c.

[4] According to *Remark 4.1* the potential $\Pi(\mathfrak{U}_h)$ in (10.83) depends for β approaching zero only on the projection part $\Pi_c(\mathfrak{U}_\pi)|_v$, leading to rank deficiency. However when $\beta = 1$ pure FEM results will be reproduced which are related to the internal mesh.

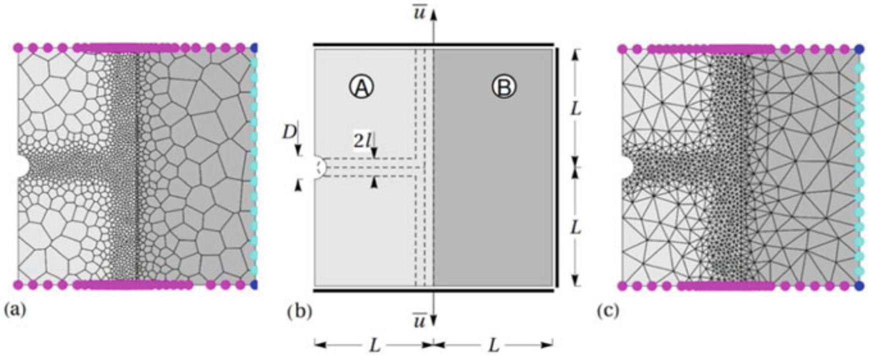


Fig. 10.27 Bi-material specimen. Geometry and boundary conditions in (**b**) along with a virtual element Voronoi mesh in (**a**) and a triangular finite element mesh in (**c**)

Figure 10.28 shows the contour plot of the fracture phase-field d simulated using the virtual element formulations with a Voronoi mesh. The analysis is performed for two different length scales $l_1 = 1.25$ mm and $l_2 = 5.0$ mm with the same length/mesh ratio $r = 8$. The crack phase-field initiates at the notch-tip and propagates horizontally up to the interface between the two materials A and B. Thereafter it branches along the interfaces vertically until final failure. A sharp crack pattern is obtained for the smaller length scale parameter l, as demonstrated in Fig. 10.28c.

Load-displacement curves for different length scales, using finite and virtual elements formulations, are displayed in Fig. 10.29a. All simulations show similar behaviour before crack initiation. Thereafter, the force drops sharply to a lower level due to cracking. This load level represents the residual forces of the undamaged material B of the specimen.

Next, the influence of the stabilization parameter β used in the virtual element formulation is illustrated in Fig. 10.29b when employing the Voronoi mesh. It can be observed that comparable load-displacement curves (F-$\bar{u}$ curves) are obtained. Hence, the simulation results are independent of the parameter β.

The tensile test with two notches or holes is concerned with the capability and the flexible choice of the number of nodes in a virtual element. To this end, a numerical tension test of two specimen is performed. The first one is a double edge notched specimen and second the one is a specimen with two holes. The purpose of this test is to illustrate the effects of the hole/notch shape on the crack-initiation and curved-crack-propagation, representing a mixed-mode fracture.

The geometrical setup and the loading conditions of the notched specimen are depicted in Fig. 10.30a and for the holes-specimen in Fig. 10.30b. The size of the specimen is chosen to be: $L = 20$ mm, $H = 10$ mm, $R = 2$ mm, $A = 3$ mm and $B = 1$ mm.

The plate is fixed at the bottom edge in x- and y-directions and the top edge in x-direction. The shear modulus is chosen as $\mu = 8.0$ kN/mm^2, Poisson's ratio $\nu = 0.3$,

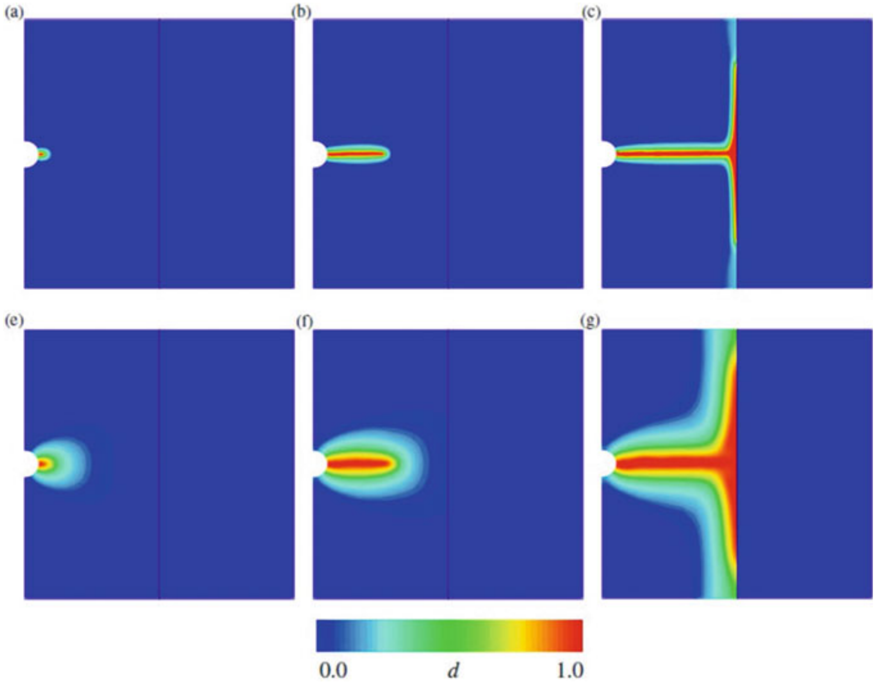


Fig. 10.28 Bi-material specimen. Contour plots of the fracture phase-field d for two fracture length scales l and three different deformation states up to final rupture. **a**–**c** $l_1 = 1.25$ mm and **e**–**g** $l_2 = 5.0$ mm

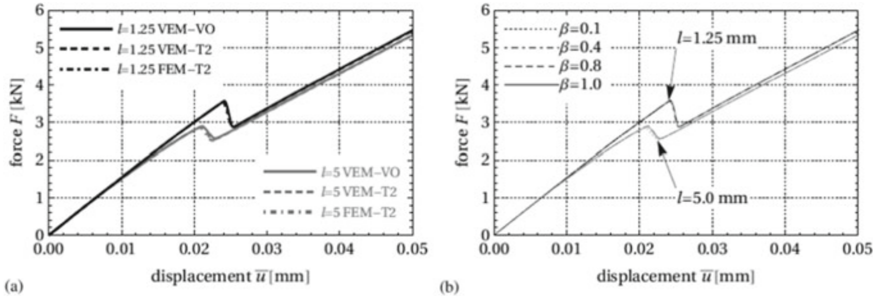


Fig. 10.29 Load–displacement curves for the Bi-material test with two different fracture length scales. **a** A comparison between different VEM and FEM discretizations and **b** numerical study of the stabilization parameter influence β

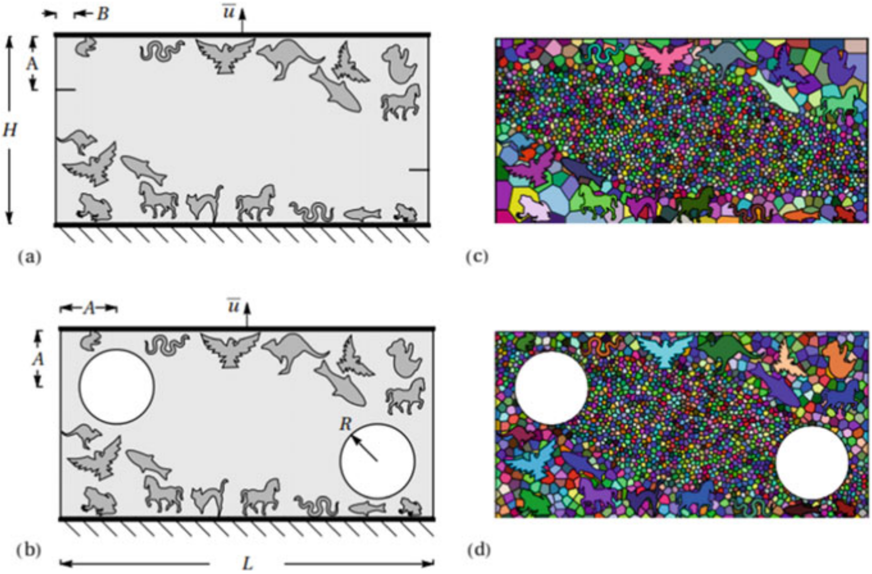


Fig. 10.30 Tensile test with two notches or holes. **a** and **b** Geometry and boundary conditions. **c** and **d** VEM with Voronoi mesh

the viscosity of the crack propagation as $\eta = 10^{-6}$ kNs/mm^2, the critical energy release rate as $g_c = 1.0 \times 10^{-3}$ kN/mm, the stabilization parameter as $\beta = 0.4$ and the fracture length scale parameter as $l = 0.1$ mm. The specimen is loaded until final failure by an increasing vertical displacement which is prescribed at the top edge.

To show the flexibility and generality of the virtual element formulation, a mesh with various animal-shaped, non-convex Voronoi cells (bird, horse, snake, frog, koala, fish, kangaroo,…) is employed in the undamaged zones. A refined mesh using a standard Voronoi cells is used in the expected fracture zone as outlined in Fig. 10.30c, d.

The crack phase-field initiates at the two notch-tips, see Fig. 10.31a or at the two holes, see Fig. 10.31e. Thereafter, the left crack propagates towards the bottom surface while the right crack propagates towards the top surface, resulting in a mixed-mode fracture, see Fig. 10.31b–d for the notched specimen and Fig. 10.31f–h for the specimen with holes. It is interesting to note, that the crack path in the middle of the plate is almost the same for both specimen. The complex animal-shaped virtual elements do not influence the solution which is related to the fact that all virtual elements with straight edges pass the patch test by construction.

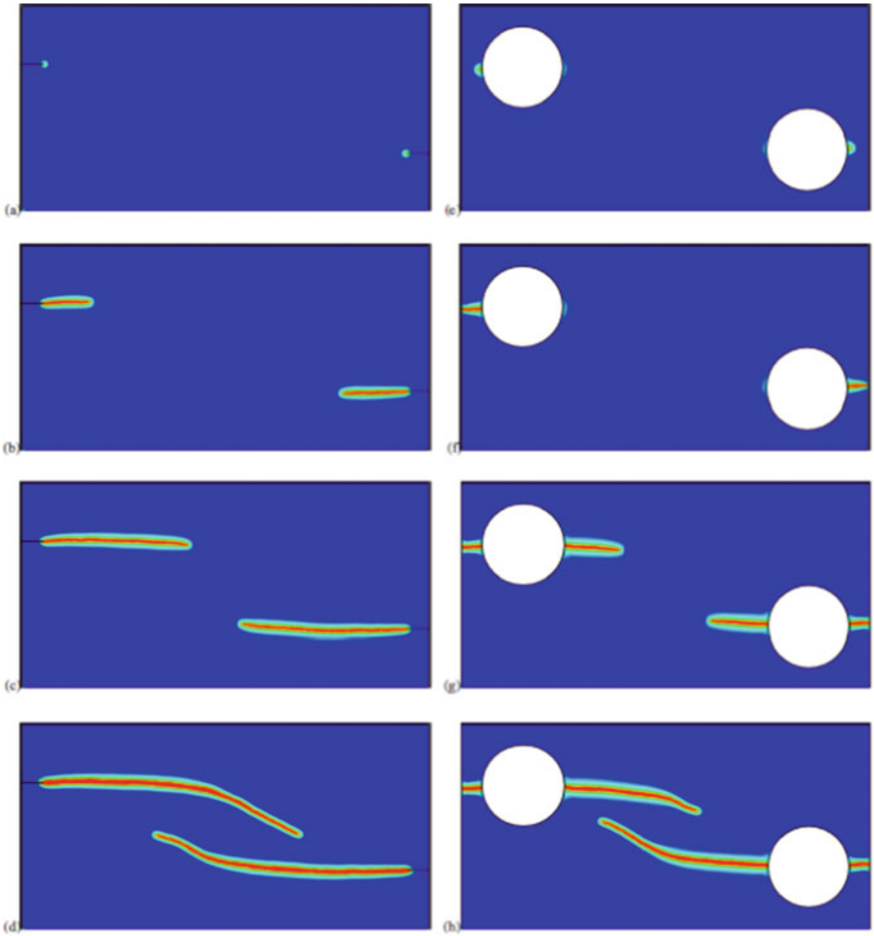


Fig. 10.31 Contour plots of the fracture phase-field d for the double edge notched specimen in (**a**)–(**d**) and for the two-holes specimen in (**e**)–(**h**)

10.4 Phase Field Methods for Ductile Fracture Using Virtual Elements

The phase field approach for ductile fracture enables to capture the progression of cracks in metals, like steel and alloys. Contrary to brittle fracture the solids undergo finite deformation near cracks which will be described by a finite plasticity model based on a pseudo-potential density functional as introduced in Chap. 8. This potential contains a degrading elastic-plastic part and a contribution due to fracture described by the phase field approach. The numerical simulation and prediction of the crack growth is based on the virtual element method.

10.4.1 Governing Equations for Phase Field Ductile Fracture

This section outlines the theory of ductile fracture in elastic-plastic solids at large deformations. The basic relations for finite strain J_2-plasticity can be found in Sect. 2.2.3.2. They describe the inelastic response of a solid by the elastic left Cauchy-Green tensor $\boldsymbol{b}_e = \boldsymbol{F}_e \boldsymbol{F}_e^T = \boldsymbol{F}\boldsymbol{C}_p^{-1}\boldsymbol{F}^T\boldsymbol{C}_p$, see (2.74), where $\boldsymbol{C}_p = \boldsymbol{F}_p^T \boldsymbol{F}_p$ is the plastic part of the right Cauchy-Green tensor. The applied elasto-plastic model including hardening is described in Eqs. (2.76) to (2.81).

The above introduced variables characterize the ductile failure response of a solid. The general formulation is based on the global primary displacement field $\boldsymbol{u}$ and the global phase field d in $\mathfrak{U}$. The constitutive state variables are summarized in $\mathfrak{C}$

$$\mathfrak{U} = \{\boldsymbol{u}, d\} \quad \text{and} \quad \mathfrak{C} = \{\boldsymbol{b}_e, \alpha, \mathcal{H}, d, \nabla d\}. \tag{10.88}$$

Constitutive phase field modeling of ductile fracture is based on the definition of a pseudo-energy density per unit volume. It consists of degrading elastic W_{elas} and plastic W_{plas} energies as well as a contribution due to fracture W_{frac}, which together describe the accumulated dissipative energy,

$$W(\mathfrak{C}) = W_{elas}(\boldsymbol{b}_e, d) + W_{plas}(\alpha, d) + W_{frac}(\mathcal{H}, d, \nabla d). \tag{10.89}$$

The elastic contribution is provided by a Neo-Hookean strain energy function for a homogeneous compressible isotropic elastic material

$$W_{elas}(\boldsymbol{b}_e, d) = g(d)\,\left[\psi_{vol}(\boldsymbol{b}_e) + \psi_{iso}(\boldsymbol{b}_e)\right], \tag{10.90}$$

with the volumetric and isochoric part

$$\psi_{vol}(\boldsymbol{b}_e) = \frac{\kappa}{4}(I_3 - 1 - \ln I_3) \quad \text{and} \quad \psi_{iso}(\boldsymbol{b}_e) = \frac{\mu}{2}(I_3^{-1/3} I_1 - 3)\,, \tag{10.91}$$

in terms of the bulk modulus κ, the shear modulus μ, the degradation function $g(d) = (1-d)^2$ and the invariants: $I_1 = \operatorname{tr} \boldsymbol{b}_e$ and $I_3 = \det \boldsymbol{b}_e$. The plastic contribution is assumed to have the form

$$W_{plas}(\alpha, d) = g(d)\ \psi_p(\alpha) \quad \text{with} \quad \psi_p = Y_0\,\alpha + \frac{H}{2}\alpha^2 + (Y_\infty - Y_0)\big(\alpha + \exp[-\delta\alpha]/\delta\big) \tag{10.92}$$

with the initial yield stress Y_0, infinite yield stress $Y_\infty \geq Y_0$, the isotropic hardening modulus $H \geq 0$ and the saturation parameter δ.

In order to enforce a crack evolution only in tension, the volumetric elastic energy is additively decomposed into a positive part ψ_{vol}^+ due to tension and a negative part ψ_{vol}^- due to compression[5] as

[5] There exist different possibilities to introduce the crack evolution due to tension, for an overview see Ambati et al. (2015a).

$$W_{elas} = g(d)\,\left[\psi^{+}_{vol}(\boldsymbol{b}_e) + \psi_{iso}(\boldsymbol{b}_e)\right] + \psi^{-}_{vol}(\boldsymbol{b}_e) \quad \text{with} \quad \psi^{\pm}_{vol} = \frac{\kappa}{4}(I^{\pm}_3 - 1 - \ln I^{\pm}_3) \tag{10.93}$$

in terms of the positive I_3^+ and the negative I_3^- third invariant, defined as

$$\begin{aligned} I_3^+ &= \max\{I_3, 1\} = \langle I_3 - 1\rangle_+ + 1 = \tfrac{1}{2}\left\{\big(I_3 - 1\big) + \big|I_3 - 1\big|\right\} + 1 \\ I_3^- &= \min\{I_3, 1\} = \langle I_3 - 1\rangle_- + 1 = \tfrac{1}{2}\left\{\big(I_3 - 1\big) - \big|I_3 - 1\big|\right\} + 1 \end{aligned} \tag{10.94}$$

Following the Coleman–Noll procedure, the Kirchhoff stresses tensor $\boldsymbol{\tau}$ and the first Piola–Kirchoff stress tensor $\boldsymbol{P}$ are obtained from the elastic strain energy function $W_{elas}(\boldsymbol{b}_e, d)$ in (10.93) for isotropic material as

$$\boldsymbol{\tau} = 2\boldsymbol{b}_e \frac{\partial W_{elas}}{\partial \boldsymbol{b}_e} \quad \text{and} \quad \boldsymbol{P} = \boldsymbol{\tau}\boldsymbol{F}^{-T} \tag{10.95}$$

The fracture part of pseudo-energy density (10.89) takes the form

$$W_{frac}(\mathcal{H}, d, \nabla d) = 2\frac{\psi_c}{\zeta}\; l\; \gamma_l(d, \nabla d) + \frac{\eta_f}{2\,\Delta t}(d - d_n)^2 + g(d)\,\mathcal{H} \tag{10.96}$$

where $\Delta t = t - t_n > 0$ denotes the time step, $\psi_c > 0$ is a critical fracture energy and ζ controls the post-critical range after crack initialization. The history field $\mathcal{H}$ is defined by

$$\mathcal{H} = \max_{s\in[0,t]} D(\boldsymbol{b}_e, \alpha; s) \geq 0 \quad \text{with} \quad D = \Big\langle \psi^{+}_{vol} + \psi_{iso} + \psi_p - \psi_c \Big\rangle_+ \tag{10.97}$$

with the Macaulay bracket $\langle x\rangle_+ = (x + |x|)/2 = \max(x, 0)$, that ensures the irreversibility of the crack evolution.

The finite elasto-plastic model requires additionally the formulation of a yield function, a hardening law and an evolution equation for the plastic variables. The yield function restricts the elastic region. By assuming J_2-plasticity with nonlinear isotropic hardening the yield function has the form

$$\chi = \sqrt{3/2}\;|\boldsymbol{f}^p| - r^p \quad \text{with} \quad \boldsymbol{f}^p = \operatorname{dev}[\boldsymbol{\tau}] = \boldsymbol{\tau} - \frac{1}{3}\operatorname{tr}[\boldsymbol{\tau}]\boldsymbol{1} \quad \text{and} \quad r^p = \frac{\partial W_{plas}}{\partial \alpha} \tag{10.98}$$

in terms of the deviatoric plastic driving force $\boldsymbol{f}^p$ and the resistance force r^p. With the yield function at hand, we define the dual dissipation function for visco-plasticity according to a Perzyna-type model as

$$\Phi^*(\boldsymbol{f}^p, r^p) = \frac{1}{2\eta_p}\Big\langle \sqrt{3/2}\;|\boldsymbol{f}^p| - r^p \Big\rangle_+^2 \tag{10.99}$$

with η_p being the viscosity parameter of the rate dependent plastic deformation. The evolution equations for the plastic variables are

$$-\frac{1}{2}\mathcal{L}_v \boldsymbol{b}_e = \dot{\gamma}\,\boldsymbol{n}\,\boldsymbol{b}_e \quad \text{with } \boldsymbol{n} = \frac{\partial \chi}{\partial \boldsymbol{f}^p} \quad \text{and } \dot{\alpha} = \dot{\gamma} = \frac{1}{\eta_p}\langle\, \chi\,\rangle_+\,, \tag{10.100}$$

where $\mathcal{L}_v$ denotes the Lie derivative in time. The evolution Eq. (10.100) can be recast with (2.74) in an alternative form

$$\dot{\boldsymbol{C}}_p^{-1} = -2\,\dot{\gamma}\,\boldsymbol{F}^{-1}\,\boldsymbol{n}\,\boldsymbol{F}\,\boldsymbol{C}_p^{-1} \tag{10.101}$$

which will be employed later within the algorithmic treatment of plasticity. The Karush–Kuhn–Tucker conditions for the elasto-plastic model are

$$\chi \le 0\,, \quad \dot{\gamma} \ge 0 \quad \text{and } \chi\,\dot{\gamma} = 0. \tag{10.102}$$

The pseudo potential density functional is used for the material modeling and computation within the *AceGen* coding environment. This functional depends on the elastic and the fracture parts. Note that the plastic history variables and the crack driving force have to be kept constant during the first variation

$$\Pi(\mathfrak{U}, \boldsymbol{h}) = \int_\Omega W(\mathfrak{C})\,\mathrm{d}\Omega - \Pi_{ext}(\boldsymbol{u}) \quad \text{with } \Pi_{ext}(\boldsymbol{u}) = \int_\Omega \bar{\boldsymbol{f}} \cdot \boldsymbol{u}\,\mathrm{d}\Omega + \int_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \boldsymbol{u}\,\mathrm{d}\Gamma \tag{10.103}$$

where $\boldsymbol{h} = \{\boldsymbol{C}_p^{-1}, \alpha, \mathcal{H}\}$ contains the history field array for the plastic strain measures and the crack driving force.

10.4.2 Formulation of the Virtual Element Method

For the construction of the virtual element method for ductile fracture, we follow the same formulations as in Chap. 8 for the finite deformation virtual element with elasto-plasticity along with the phase field formulation introduced in Sect. 10.3.4. Thus for the simplest form of a two-dimensional low order virtual element a linear ansatz, see (10.77) for the pure mechanical case,

$$\mathfrak{U}_\pi = \begin{Bmatrix} u_{\pi\,x} \\ u_{\pi\,y} \\ d_\pi \end{Bmatrix} = \mathbf{a} \cdot \mathbf{N}_\pi = \begin{bmatrix} a_1 & a_4 & a_7 \\ a_2 & a_5 & a_8 \\ a_3 & a_6 & a_9 \end{bmatrix} \begin{Bmatrix} 1 \\ X \\ Y \end{Bmatrix} \tag{10.104}$$

yields the gradient of the projection as provided in (10.78)

$$\nabla \mathfrak{U}_\pi = \mathbb{D}_\nabla^{(2,1)}\,\mathfrak{U}_v$$

and the projected function which follows from (10.80) as, see (10.82),

$$\mathfrak{U}_\pi = \mathbf{H}^{(2,1)}(X, Y)\, \mathbb{D}_{\mathfrak{U}}^{(2,1)}\, \mathfrak{U}_v$$

where she set of nodal degrees of freedom of a virtual element Ω_v are defined as, see (10.79),

$$\mathfrak{U}_v = \left\langle \mathfrak{U}_1\, \mathfrak{U}_2\, \ldots\, \mathfrak{U}_{n_V} \right\rangle^T$$

with the nodal degrees of freedom at node K given by $\mathfrak{U}_K = \left\langle u_{Kx}\, u_{Ky}\, d_K \right\rangle^T$.

The pseudo potential function (10.103) can now be approximated by assembling all virtual element contributions

$$\Pi(\mathfrak{U}, \boldsymbol{h}) = \mathop{\mathbb{A}}_{v=1}^{n_v} \left[\Pi_c(\mathfrak{U}_\pi, \boldsymbol{h}_\pi)|_v + \Pi_{stab}(\mathfrak{U}_h - \mathfrak{U}_\pi, \boldsymbol{h}_\pi)|_v \right]. \tag{10.105}$$

More details related to the determination of the projection $\mathfrak{U}_\pi$ for the displacement field and the computation of the history variables $\boldsymbol{h}_\pi$ as well as the computations of the consistency term and the energy stabilization can be found in Sects. 10.3.4 and 8.1.

10.4.3 Numerical Ductile Fracture Simulations

The performance of the proposed virtual element formulation for ductile fracture is demonstrated by means of two representative numerical examples. The results are compered with numerical simulations employing the finite element method. The material parameters used in this section are the same for both examples and provided in Table 10.3.

To illustrate the capability and the flexible choice of the number of nodes within a virtual element, various animals-shaped Voronoi cells (bird, horse, snake, frog, koala, fish, kangaroo,...) are employed in the undamaged as well as the damaged zones (i.e. an area of interest) for the virtual element formulation in the following examples.

The single-edge notched shear test is defined by a square plate with a horizontal notch placed at the middle height from the left outer surface to the center of the specimen. The geometrical setup and the loading conditions of the specimen are depicted in Fig. 10.32. The size of the square specimen is chosen to be $L = 0.5$ mm. The bottom edge of the plate is fixed and an increasing shear loading, by prescribing the horizontal displacements, is applied at the top edge until the plate is fully broken.

The specimen is discretized with different virtual elements including animal shaped Voronoi cells, see Fig. 10.32 (right) and finite elements (FEM), see Fig. 10.32 (left). A mesh refinement in the expected fracture zone is applied, it is based on the ratio $r = l/h_e$ between the mesh size h_e and the fracture length scale l, as sketched in Fig. 10.32.

Virtual elements with a Voronoi mesh are denoted by VE-VO; virtual elements with a mesh consisting of 6 noded triangles are described by VE-T2, representing a

Table 10.3 Material parameters used in the numerical examples

Name	Symbol	Value	Unit
Young's modulus	E	206.9	GPa
Poisson's ratio	ν	0.29	–
hardening parameter	H	0.13	GPa
initial yield stress	Y_0	0.45	GPa
infinite yield stress	Y_∞	0.45/1.165	GPa
saturation parameter	δ	16.93	–
critical fracture energy	ψ_c	0.025/2.0	GPa
plastic viscosity	η_p	10^{-8}	GPa.s
fracture viscosity	η_f	10^{-8}	GPa.s
fracture length scale	l	0.008/0.02	mm
fracture parameter	ζ	8.0/1.0	–

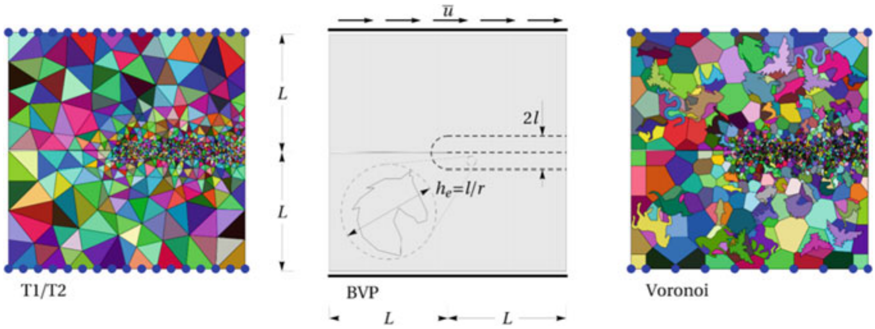


Fig. 10.32 Single-edge notched shear test—Geometry and boundary conditions along with the VEM Voronoi mesh and the triangular finite element mesh

first order virtual element.[6] Triangular finite elements with linear ansatz are denoted by FE-T1 while finite elements with a quadratic ansatz are described by FE-T2. These different discretizations are employed to compare virtual and finite elements and to test the robustness of the virtual element formulation.

The evolution of the crack phase field d in comparison with the evolution of the equivalent plastic strain α is depicted in Fig. 10.33 for three different deformation stages up to final rupture. The numerical simulation was performed by using the virtual element formulation with various animals-shaped Voronoi cells, for the fracture length scale $l = 0.008$ mm and the length/mesh ratio $r = 4$. The crack phase-field initiates at the notch-tip, where the maximum equivalent plastic strain α is concentrated. Thereafter, the crack propagates horizontally until separation.

[6] In this case T2 implies that the used triangle mesh is the same as for the second order finite element. But VE-T2 does not mean a second order virtual element with quadratic ansatz. VE-T2 is a linear element with just 6 nodes.

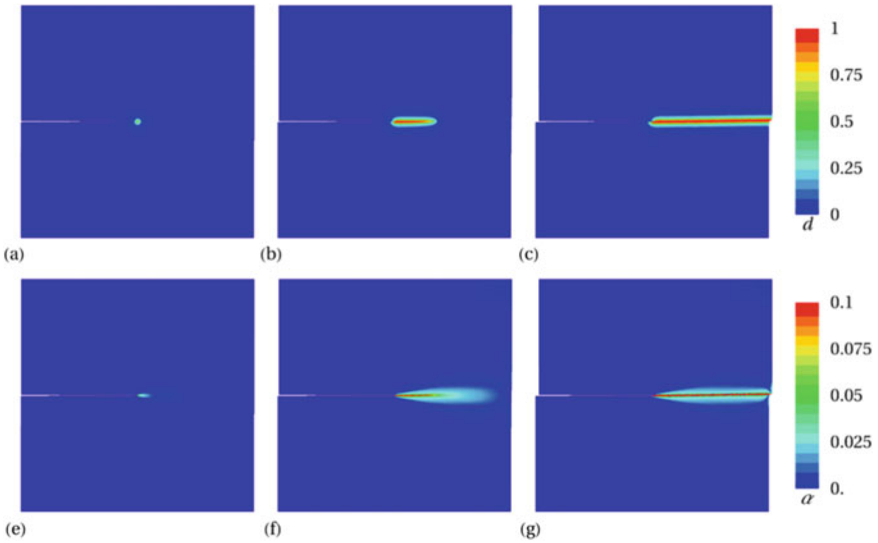


Fig. 10.33 Single-edge notched shear test—Contour plots of the fracture phase-field d in (**a**)–(**c**) and the equivalent plastic strain α in (**e**)–(**g**) for three different deformation states up to final rupture

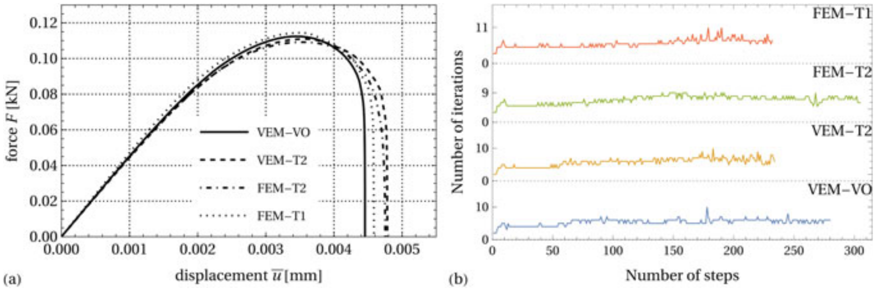


Fig. 10.34 Single-edge notched shear test—**a** Load–displacement curves, **b** comparison between the total number of iterations in each time step required to achieve convergence for different discretization. The fracture length scale is set to $l = 4h_e = 0.008$ mm

Load-displacement curves of the overall structural response are plotted for the different finite and virtual elements in Fig. 10.34a. The virtual element results are in a good agreement with the reference simulations using finite elements. Table 10.4 compares the different finite and virtual element discretization, related to the F-$\bar{u}$ curves in Fig. 10.34a, with respect to robustness and efficiency. Figure 10.34b illustrates the convergence properties for the different element formulations plotted in Fig. 10.34a at the final deformation state $\bar{u} = 0.0048$ mm. We observe that virtual elements require fewer steps and less iterations for final convergence, even when compared with finite element method of higher order. Thus within this application

Table 10.4 A comparison between different FEM and VEM discretizations, related to F-U curves in Fig. 10.34a

	VE-VO	VE-T2	FE-T2	FE-T1
Number of elements	12369	11109	11109	11109
Number of nodes	24744	22291	22290	5591
No. of equations	74195	66826	66826	16749
Number of steps	280	234	305	232
Total number of iterations	2372	1504	2610	1599
Average iterations/step	6.1134	5.76245	6.97861	6.07985

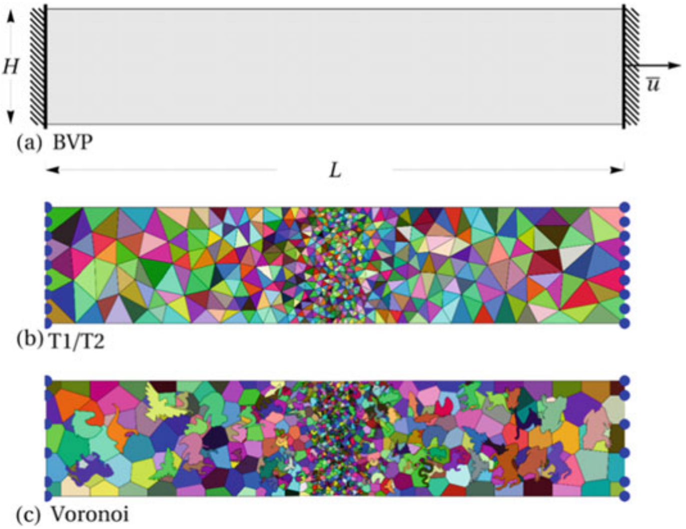


Fig. 10.35 Axial stretch of a bar—**a** Geometry and boundary conditions, **b** triangular element mesh and **c** Voronoi mesh with non-convex elements

the virtual element scheme is more robust than the finite element method, however, this comes with extra computational costs, due to the energy stabilization.

Axial stretch of a bar is discussed next. It is concerned with analyzing the ductile failure behaviour of a bar due to a prescribed displacement $\bar{u}$ in axial direction at the right side. Experimental observation shows that necking takes place before final ductile rupture. The localized plastic strains in the necking area and the subsequent ductile failure response will be used to assess the robustness of the virtual element formulation.

The geometrical setup and the boundary conditions of the bar with height $H = 2$ mm and length $L = 10$ mm are illustrated in Fig. 10.35a. A geometrical imperfection is introduced in the central zone to trigger localization and necking in the center of the bar. For that, the net cross section of the specimen is reduced by choosing the height at the center to be $H_c = 0.99\ H$. A Dirichlet boundary condition

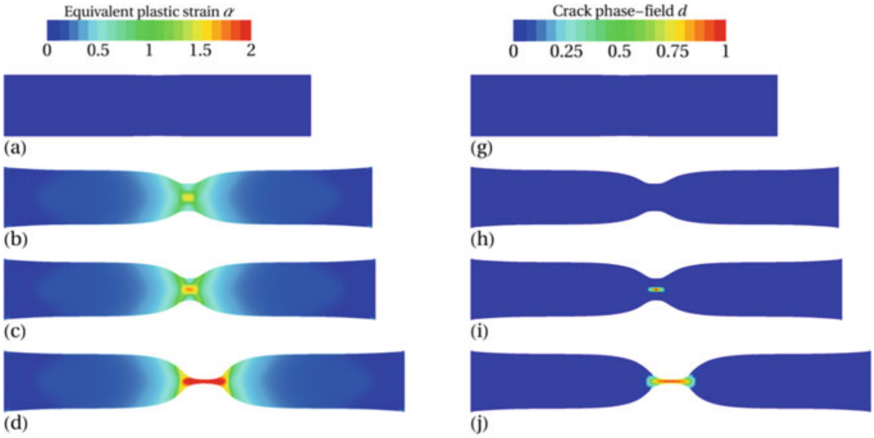


Fig. 10.36 Axial stretch of a bar—Contour plots of the equivalent plastic strain α in (**a**)–(**d**) and the fracture phase-field d in (**g**)–(**j**) for four different deformation states up to final rupture

$\bar{\boldsymbol{u}} = \mathbf{0}$ is introduced at the left edge of the bar. A horizontal displacement is applied at the right edge which has the magnitude of 20% of the bar length e.g. $\bar{u} = 0.2L$. The mesh is refined in the expected fracture zone for all virtual and finite element discretizations, see Fig. 10.35b, c.

Figure 10.36 depicts the contour plots of the equivalent plastic strain α and the fracture phase field d which stem from the simulation with the virtual element method for a fracture length scale $l = 0.02$ mm and different deformation stages up to final failure. Various animal-shaped Voronoi cells were considered in the discretization. We observe large plastic deformations in the necking zone with a concentration of hardening in Fig. 10.36b, c at the specimen center. Severe necking initiates the crack at the center zone as demonstrated in Fig. 10.36i. Thereafter, the crack phase-field propagates outward following the equivalent plastic strain path until complete failure as shown in Fig. 10.36j.

Load-displacement curves are displayed in Fig. 10.37 for different element types of the virtual and finite element formulation. All simulations show similar behaviour before crack initiation as the results are indistinguishable up to $\bar{u} = 0.7$ mm. Thereafter, during the necking process, the curves of virtual and finite elements almost coincide, except the FE-T1 solution which exhibits a stiffer response. The latter is related to the well known locking of linear finite elements in applications regarding J_2-plasticity. The capability of the virtual element with linear ansatz to compute accurate results—even without special treatment of the plastic incompressibility—is a clear advantage. Here, the linear virtual element is comparable to using finite elements of higher order, even if a mild locking occurs at very large strains starting at a displacement of $\bar{u} = 1.2$ mm which is associated with a strain of 12%. It is possible to avoid locking by using an element formulation as discussed in Sect. 6.3.3.

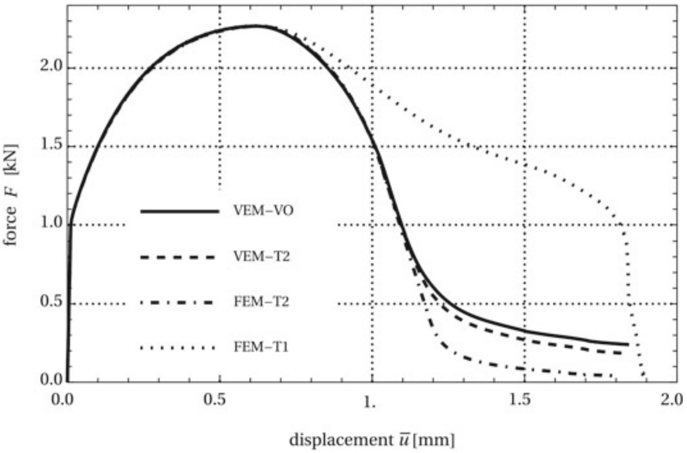


Fig. 10.37 Axial stretch of a bar—Load–displacement curves for different VEM and FEM discretization. The fracture length scale is set to $l = 4h_e = 0.02$ mm

10.5 Adaptive VEM for Phase-Field Fracture

In the previous sections of this chapter, we demonstrated the efficiency of the virtual element method for different fracture processes. Now, a robust and efficient adaptive virtual element method is proposed which is based on the phase-field formulation of fracture, see Sect. 10.3. Since a fine mesh is only needed at the fracture zone in order to reproduce the physical behaviour correctly a mesh adaptivity reduces the computation cost remarkably. The advantage of the virtual element method is its flexibility to have as many nodes along an element edge, still fulfilling the patch test, and thus meshes can be generated that include hanging-nodes at edges. Such nodes can easily be included and do not destroy the convergence behaviour of the virtual element approximation.

10.5.1 Governing Equations

The phase field formulation for brittle materials is here extended to finite elastic strains for elastic solids in two dimensions ($\Omega \in \mathcal{R}^2$). The fracture response of the solid at material points $\boldsymbol{X}$ and time t is described by the displacement field $\boldsymbol{u}(\boldsymbol{X}, t)$ and the crack phase-field $d(\boldsymbol{X}, t)$ with $\dot{d} \geq 0$. Hereby, $d(\boldsymbol{X}, t) = 0$ and $d(\boldsymbol{X}, t) = 1$ depict the unbroken and fully broken state of the material, respectively, see also Sect. 10.3.2. The deformation gradient $\boldsymbol{F}$ is defined with the displacement gradient $\nabla \boldsymbol{u}$ as, see (2.2),

$$\boldsymbol{F} = \boldsymbol{I} + \nabla \boldsymbol{u}. \tag{10.106}$$

For the phase-field problem, the sharp-crack surface topology $\Gamma \rightarrow \Gamma_l$ is regularized by the crack surface functional, as in the geometrically linear case,

$$\Gamma_l(d) = \int_\Omega \gamma_l(d, \nabla d)\, \mathrm{d}\Omega \quad \text{with} \quad \gamma_l(d, \nabla d) = \frac{d^2}{2l} + \frac{l}{2}|\nabla d|^2 \tag{10.107}$$

based on the crack surface density function γ_l per unit volume of the solid and the fracture length scale parameter l that governs the regularization. Hence, the combination of elasticity with the first-order gradient damage modeling focuses on the set constitutive state variables

$$\mathfrak{C} = \{\boldsymbol{F}, d, \nabla d\}. \tag{10.108}$$

The constitutive work density function W is assumed to depend on these constitutive state variables $\mathfrak{C}$. It consists of the sum

$$W(\mathfrak{C}) = W_{\text{bulk}}(\boldsymbol{F}, d) + W_{\text{frac}}(d, \nabla d) \tag{10.109}$$

of a degrading elastic bulk energy W_{bulk} and a contribution due to fracture W_{frac}, which contains the accumulated dissipative energy. The hyper-elastic bulk contribution in Eq. (10.109) is assumed to be a non-linear function, given as

$$W_{\text{bulk}}(\boldsymbol{F}, d) = g(d)\ \left(\psi_{\text{iso}}(\boldsymbol{F}) + \psi_{\text{vol}}(J_{F+})\right) + \psi_{\text{vol}}(J_{F-}) \tag{10.110}$$

with the isochoric $\psi_{\text{iso}}(\boldsymbol{F})$ and the volumetric part $\psi_{\text{vol}}(J_F)$

$$\psi_{\text{iso}}(\boldsymbol{F}) = \frac{\mu}{2}\left(J_F^{-\frac{2}{3}}\text{tr}(\boldsymbol{F}\boldsymbol{F}^{\text{T}}) - 3\right) \tag{10.111}$$

$$\psi_{\text{vol}}(J_F) = \frac{\kappa}{4}\left(J_F - 1 - 2\log(J_F)\right) \tag{10.112}$$

where $J_{F\pm}$ is given by $J_{F\pm} = \langle J_F \rangle_\pm = (J_F \pm |J_F|)/2$ and $\langle \square \rangle_\pm$ is the Macaulay bracket. The constitutive parameters κ and μ represent the material bulk and shear modulus, respectively, and $J_F = \det(\boldsymbol{F})$ is the determinant of deformation gradient $\boldsymbol{F}$. The function $g(d) = (1-d)^2$ models the degradation of the stored elastic energy of the solid due to fracture and it degrades the positive volumetric part and full shear part. The fracture contribution in (10.109) is defined as

$$W_{\text{frac}}(d, \nabla d) = \mathcal{G}_c\ \gamma_l(d, \nabla d) \tag{10.113}$$

where, $\mathcal{G}_c > 0$ is the Griffith's critical energy release rate.

The regularized crack surface functional evolution (10.107) can be driven by the constitutive functions as outlined in Aldakheel et al. (2018a), postulating a global evolution equation of the regularized crack surface as

$$\frac{d}{dt}\Gamma_l(d) = \frac{1}{l}\int_\Omega [\,(1-d)\mathcal{H} - \eta\dot{d}\,]\,\dot{d}\,\mathrm{dV} \geq 0\,, \tag{10.114}$$

where $\eta \geq 0$ is a material parameter that characterizes the artificial/numerical viscosity of the crack propagation. Various criteria can be chosen, see Aldakheel et al. (2018b, 2019, 2021a). In this case the crack driving force was selected to be either only the volumetric part $\psi_{\text{vol}}(J_{F+})$ or the sum of volumetric and isochoric part which leads to the introduction of

$$\mathcal{H} = \max_{s\in[0,t]} D(\boldsymbol{x}, s) \geq 0 \quad \text{with} \quad D = \psi_{\text{vol}}(J_{F+}) + \psi_{\text{iso}}(\boldsymbol{F}) \tag{10.115}$$

as a local history variable that accounts for the irreversibility of the phase-field evolution by filtering out a maximum value of what is known as the crack driving state function D. Then (10.114) provides the local equation for the evolution of the crack phase-field in the domain Ω along with its homogeneous Neumann boundary condition $\nabla d \cdot \boldsymbol{n} = 0$ on Γ as

$$(d - l^2\Delta d\,) + \eta\dot{d} + (d-1)\mathcal{H} = 0. \tag{10.116}$$

Here, $\boldsymbol{n}$ represents the outward normal on Γ. Finally, the fracture pseudo potential can be defined as

$$\widetilde{W}_{\text{frac}}(d, \nabla d) = d\dot{d}\eta - g(d)\mathcal{H} + \mathcal{G}_c\gamma_l(d, \nabla d) \tag{10.117}$$

which leads to specific potential

$$W(\mathfrak{C}) = W_{\text{bulk}}(\boldsymbol{F}, d) + \widetilde{W}_{\text{frac}}(d, \nabla d). \tag{10.118}$$

This formulation is very efficient when using automatic differentiation, like in the software tool *AceGen,* see Korelc and Wriggers (2016).

10.5.2 Mesh Refinement with Virtual Elements

For the construction of a standard low order virtual element method which can be used in phase field approaches, we follow the projection scheme introduced in Sect. 10.3.4. The resulting ansatz functions are then used in the same manner for large strains as outlined in Sect. 10.3 related to phase-field ductile fracture, however here restricted to pure elastic deformations.

In this section, the phase field method is used to predict the crack growth direction. For the accuracy of the phase field solution, the expected damage zone is locally refined by mean of an adaptive refinement strategy. Here different error measures

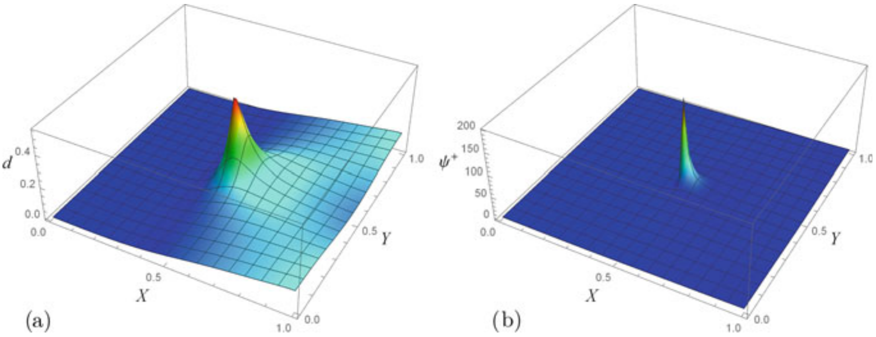


Fig. 10.38 Different refinement criteria that increase locally before the onset of crack growth. Phase field variable d in (**a**) and positive strain energy ψ^+ in (**b**)

or error indicators can be used, for an overview see e.g. Ramm et al. (2003). Two different error indicators are depicted in Fig. 10.38. The first computes simply the magnitude of the phase field d, see Fig. 10.38a, while the second error indicator uses the magnitude of the positive part of the elastic strain energy ψ^+ that contributes to the crack opening, see Fig. 10.38b. Note that the error indicator using ψ^+ concentrates very much around the crack tip with a very steep gradient. This is due to the singularity of the solution at this point. A mesh refinement scheme that is based on such localized measure will lead to a very small area of mesh refinement around the tip. Due to that the error indicator using just the phase field solution itself was selected. on one hand it results in a larger area of mesh refinement and on the other hand it is very easy to compute since d is a primary variable of the solution. Thus the criterion for mesh refinement is provided by

$$d \geq d_c \tag{10.119}$$

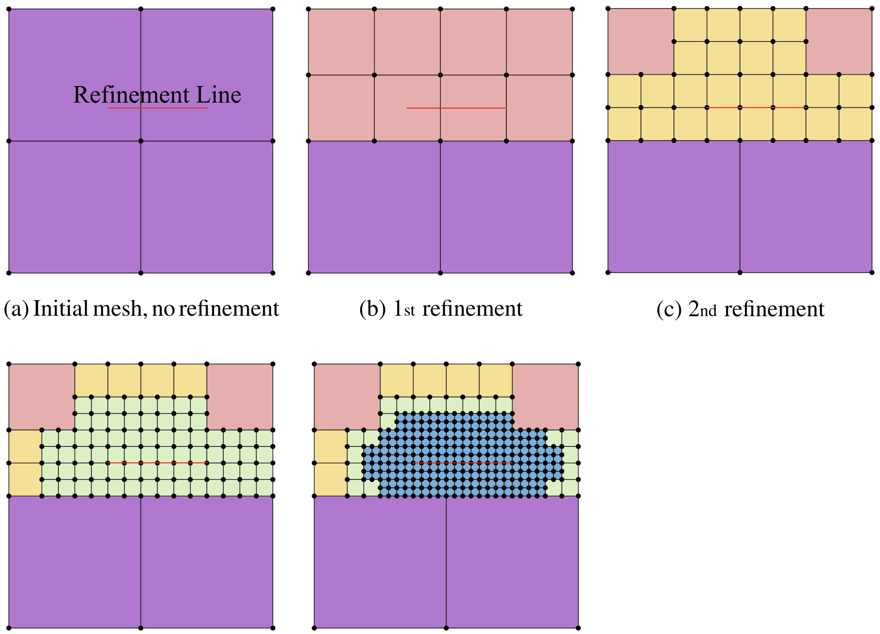
where d_c is a given threshold. Depending on the considered problem, the threshold value can be chosen between $0 < d_c < 1$. Each element that has at least one node K with phase-field value $d_K \geq d_c$, will be selected for the refinement.

A simple refining strategy is based on a refinement indicator function $f(v)$. With a loop over all virtual elements Ω_v it provides a check whether the phase field variable d_K is beyond the threshold in (10.119). The indicator function is provided by

$$f(v) = \begin{cases} 1\,, & \text{for } d_K \geq d_c \in \Omega_v \\ 0\,, & \text{for } d_K < d_c \in \Omega_v \end{cases} \tag{10.120}$$

If the indicator function $f(v) = 1$, refinement is triggered at element Ω_v. In that case, a new node is inserted on the position of the element centroid

$$\boldsymbol{X}_C|_v = \frac{1}{|\Omega_v|} \int_{\Omega_e} \boldsymbol{X} d\Omega \tag{10.121}$$



(a) Initial mesh, no refinement

(b) 1st refinement

(c) 2nd refinement

(d) 3rd refinement

(e) 4th refinement

Fig. 10.39 Mesh refinement algorithm

and additional new nodes are inserted at each element edge e of the virtual element Ω_v as

$$\boldsymbol{X}_C|_e = (\boldsymbol{X}|_e + \boldsymbol{X}|_{e+1})/2. \tag{10.122}$$

This step is only performed for new insertions. The algorithm checks if an edge has already been refined previously from a neighboring element. Thus new insertion are avoided.

Figure 10.39 illustrates the mesh refinement algorithm. Starting with a 2 × 2 mesh in Fig. 10.39a and refining around a refinement line plotted in Fig. 10.39a for different refinement depths in Fig. 10.39b–e.

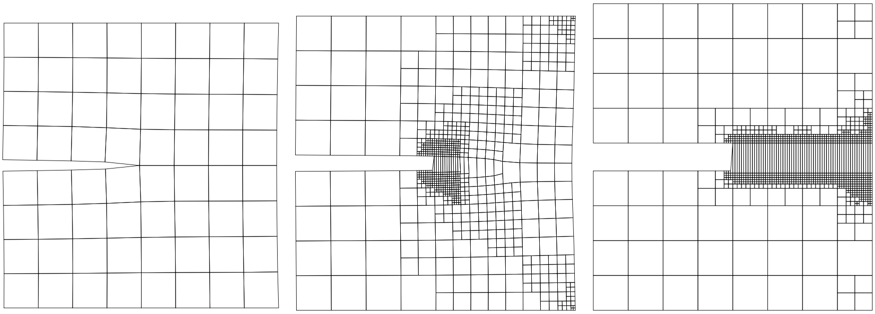
10.5.3 Adaptive Numerical Simulations for Phase-Field Fracture

The performance of the proposed refinement strategy will be demonstrated using two boundary value problems that exhibit a fracturing process.

On the computational side, a robust and efficient staggered scheme is employed using the software tool *AceFEM* in the numerical implementation to compute the

Table 10.5 Material parameters

Description	Symbol	Value	Unit
Young's modulus	E	2.1×10^4	$\frac{\text{kN}}{\text{m}^2}$
Poisson ratio	ν	0.2	
Fracture length scale	l	0.015	mm
Griffith's critical energy release rate	$\mathcal{G}_c$	10	$\frac{\text{kN}}{\text{mm}}$
Crack viscosity	η	0.01	$\frac{\text{kNs}}{\text{mm}^2}$



(a) Initiation (b) Propagation (c) Close to final state

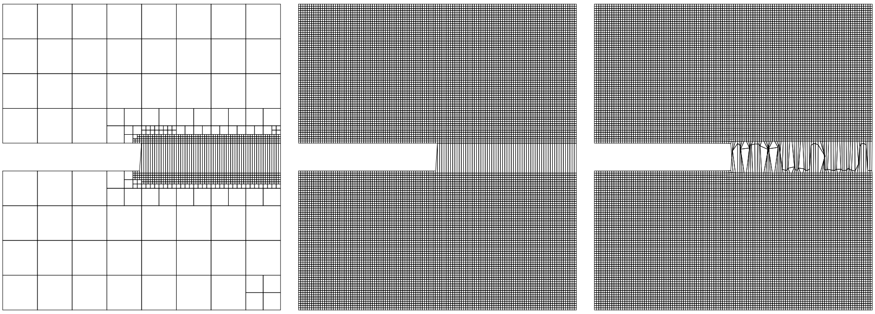
Fig. 10.40 Adaptive mesh refinements for crack phase-field evolution shown at different deformation stages until final failure

unknowns (displacement $\boldsymbol{u}$ and crack phase-field d). Within each load step a quadratic convergence is achieved, due to the fact that all formulations are linearized in a consistent manner using *AceGen*. The benchmark tests consider a square block ($L = H = 1$ mm) with a horizontal notch placed at the middle height. The material parameters used in the simulation are shown in Table 10.5.

The single-edge notched tension test is a standard example in fracture mechanics. The bottom is fully clamped. At the top a vertical displacement is described with a rate $\bar{\boldsymbol{u}}_X = 1\text{mm/s}$ until final failure at time $t = 0.1$ s which is depicted in Fig. 10.40.

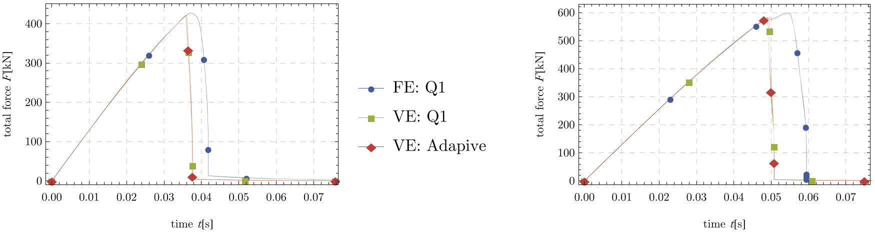
The results show a good match between a fine ($2^7 \times 2^7$ four noded Q1 elements) and adaptively refined mesh in Fig. 10.41b and c as well as a good match between VEM and FEM when looking at the peak load in Fig. 10.42. In fact, the FE-Q1 result exhibits some locking behaviour. For VE-Q1 and VE-Adaptive, the adaptive and fine meshes produce almost the same results when the virtual element method is applied as outlined in Fig. 10.42. This holds for a driving force which depends on purely on the volumetric part, see Fig. 10.42b as well as for a driving force that consists of the sum of the volumetric and isochoric part, see Fig. 10.42a.

The single-edge notched shear test is using the same block as before which is fixed at the bottom. Horizontal displacements are prescribed at the top at a rate



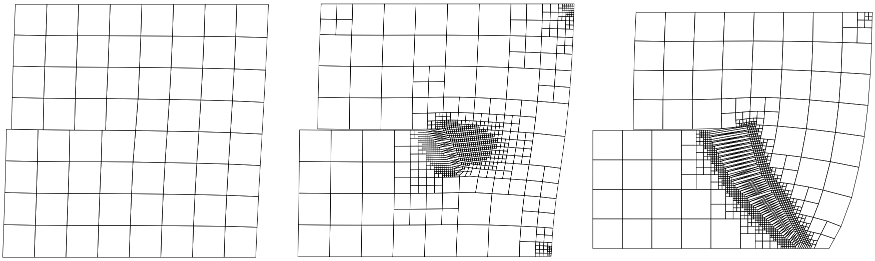
(a) VEM refined (b) VEM fine Q1 (c) FEM fine Q1

Fig. 10.41 Final state: **a** adaptive refined mesh using VEM; **b** regular fine mesh using VEM and **c** regular fine mesh using FEM



(a) $\mathcal{H} = \max_{s\in[0,t]}(\psi_{\text{vol}}(J_{F+}) + \psi_{\text{iso}}(\boldsymbol{F}))$ (b) $\mathcal{H} = \max_{s\in[0,t]}(\psi_{\text{vol}}(J_{F+}))$

Fig. 10.42 Load displacement curve for fine FE-Q1 and VE-Q1 meshes and VEM with adaptive re-meshing

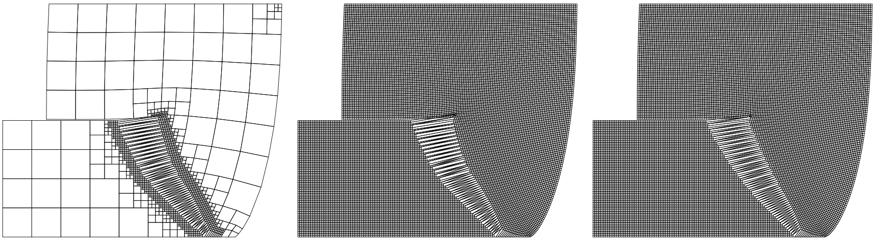


(a) Initial (b) 3rd ref. step (c) 12th ref. step

Fig. 10.43 Different deformation stages until final failure using the adaptive VEM scheme

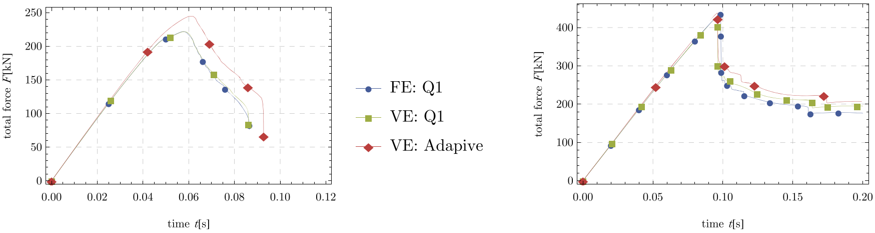
$\bar{\boldsymbol{u}}_X = 1\text{mm/s}$ until final failure at time $t = 0.2$ s. A sequence of deformed meshes until failure is outlined in Fig. 10.43.

Unlike in the tension problem, the deformation behaviour and damage states of the shear example are much more unpredictable. The solution path depends highly



(a) VEM refined (b) VEM fine Q1 (c) FEM fine Q1

Fig. 10.44 Final state: **a** adaptive refined mesh; **b** fine mesh using VEM and **c** fine mesh using FEM



(a) $\mathcal{H} = \max_{s\in[0,t]}(\psi_{\text{vol}}(J_{F^+}) + \psi_{\text{iso}}(\boldsymbol{F}))$ (b) $\mathcal{H} = \max_{s\in[0,t]}(\psi_{\text{vol}}(J_{F^+}))$

Fig. 10.45 The load displacement curve of FEM, VEM on Q1 fine mesh and VEM with adaptive re-meshing for Shear example

on the chosen split of the driving force and fracture criteria, see for details (Aldakheel et al. 2018a; Miehe et al. 2010b).

The results show a good match between the fine and adaptively refined meshes as illustrated in Fig. 10.44b and c. Furthermore, a good match is observed between virtual and finite element predictions. Similar to the first example, FE-Q1 exhibits locking behaviour which however could be avoided by a higher order ansatz.

The key point here is to illustrate the efficiency of the adaptive scheme for virtual elements in comparison with the standard uniform mesh refinements. This results in a remarkable reduction of the computation time, while producing similar results, as depicted in the load-displacement curves in Fig. 10.45. Furthermore, the refinement procedure is very simple, as shown above, due to the specific feature of the virtual element method to allow additional nodes in a consistent matter within an element,

Prediction results are completely different when employing the two different fracture driving forces $\mathcal{H}$, namely the one with only the positive volumetric part and the one with the sum of deviatoric and volumetric part. This is clearly visible when comparing the results depicted in Fig. 10.45a and b. In both cases, the developed adaptive meshing scheme yields excellent results in comparison with the uniformly refined meshes using the virtual or finite element method.

10.6 An Adaptive Scheme to Follow Crack Paths Combining Phase Field and Cutting Methods

An efficient strategy to model crack propagation can be developed by using a combination of different methods. The main idea is to employ a combination of the phase field methodology introduced in Sects. 10.3 and 10.4 with the cutting technology proposed in Sect. 10.2 and the mesh refinement in Sect. 10.5. The advantages are:

- A crack path can be easily predicted using the phase field method.
- Due to adaptivity, the necessary very fine mesh, resolving the crack, will only be present locally.
- The robust cutting technique replaces the refined mesh by a discrete crack.

In total, this combination of different methods allows a very efficient and robust solution of crack growth in solids.

10.6.1 General Idea

Different approaches to model crack propagation have been discussed in the last sections using the virtual element method. All methods have advantages and disadvantages. In the cutting scheme (Sect. 10.2) one has to compute the stress intensity factors and the direction of the crack. This is based on the linear elastic fracture mechanics and thus cannot be easily extended to nonlinear material behaviour. The phase field method (Sects. 10.3–10.4) is more general, but needs the regularization of the sharp crack discontinuities which results in higher computational effort.

Regularized modeling of Griffith-type brittle fracture in elastic solids relies on very fine meshes which are required to accurately capture the crack path. However, a global refinement strategy leads an inefficient and time-consuming analysis. If the crack path is known a priori, a local mesh pre-refinement can be used to overcome the aforementioned computational costs. However, the expected direction of the crack growth can be predicted only for some cases in advance. In this regard, various approaches were proposed to use an adaptive mesh refinement technique in which no a prior knowledge for the direction of the crack path is required, see e.g. Heister et al. (2015), Badnava et al. (2018), Nagaraja et al. (2019) and Noii et al. (2020). Most of these investigations use the phase field parameter to guide the adaptive procedure for refining only elements which are affected by the state of damage. On the other hand, Klinsmann et al. (2015) suggested to use a state variable as an indicator for the refinement procedure, which depends on the maximum positive strain energy, see also Sect. 10.5.2 for details. Both quantities are good candidates to be applied as an indicator for the refinement since at the crack tip they increase significantly. A predictor-corrector mesh refinement procedure for phase field modeling of brittle and ductile fracture can be found in Aldakheel et al. (2021b) using the phase field d as an indicator variable.

As has been shown in Sect. 10.2, treatment of crack propagation has some advantages when using the cutting technique together with virtual element formulations. They allow exploration of features such as the flexibility in dealing with complex element shapes and an arbitrary number of nodes within an element which can be changed or nodes can be added easily during the computation. Here the treatment of crack propagation is based on the phase field method to model brittle fractures in isotropic elastic solids, see Sect. 10.3. Furthermore, the adaptive refinement strategy developed in Sect. 10.5.2 is employed to obtain an accurate solution for the crack path. Finally, the crack is introduced geometrically as a real crack by using the robust cutting techniques, described in Sect. 10.2. The advantages of the approach followed in this section can be summarized as

- Refinement is performed without a priori acknowledge of the crack growth direction,
- hanging nodes are easily treated without any additional constraints and
- replacing the refined zone by a discrete crack leads to a decrease of computational simulation time.

10.6.2 Modeling Crack Propagation Using VEM

The modeling technique for crack propagation is based on three ingredients: the phase field method, adaptive mesh refinement and a scheme that is able to cut virtual elements when a crack was detected by the phase field method. These ingredients were discussed in the last sections. In detail, the phase field approach, discussed in Sect. 10.3, is coupled with an adaptive mesh refinement technique proposed in Sect. 10.5 in order to predict the direction of the crack growth. Furthermore a robust splitting algorithms to construct the crack path during the propagation is presented in Sect. 10.2.2.

10.6.3 Discontinuous Crack Propagation Using Phase Field

The same error indicator for mesh refinements, as discussed in Sect. 10.5.2, is selected for the refinement around the crack tip. Then, based on the numerical analysis with the refined mesh, the direction of the crack propagation can be estimated. VEM has the advantage that it is simple to use hanging nodes in the mesh refinement, without defining any additional constraints like in the finite element method. Once crack length and direction are known, the crack is introduced geometrically in the original, unrefined mesh by using the cutting technique, described in Sect. 10.2. Thus, after cutting, the mesh along the real crack is de-refined and does not contain the refined elements anymore, which leads to a decrease of the computation time.

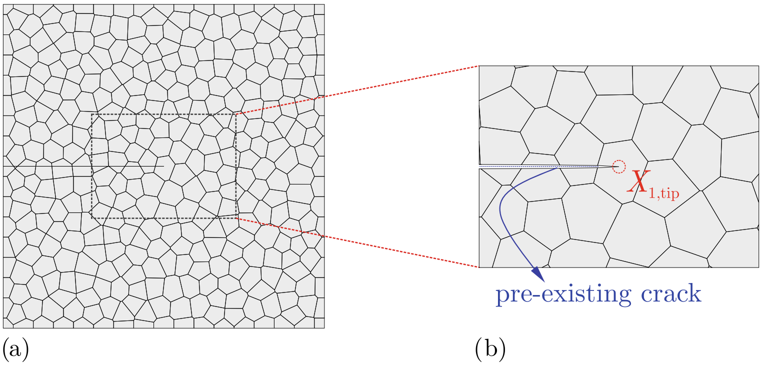


Fig. 10.46 Geometry and discretization of the specimen using Voronoi mesh

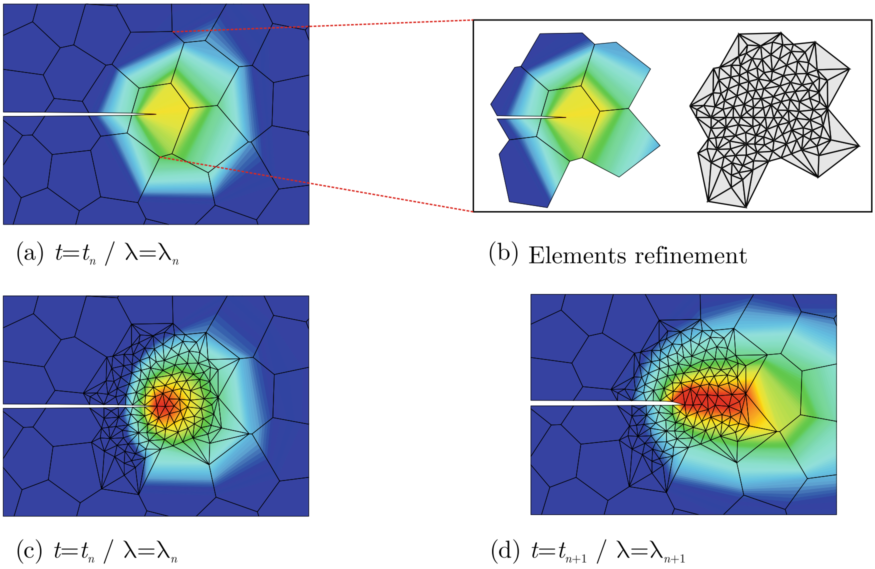

(a) t=t_n / λ=λ_n
(b) Elements refinement
(c) t=t_n / λ=λ_n
(d) t=t_{n+1} / λ=λ_{n+1}

Fig. 10.47 A schematic visualization of the refinement procedure

We will represent the general technique visually by considering the domain, depicted in Fig. 10.46. The domain is discretized by a coarse VEM-Voronoi mesh. This mesh is used for the first phase field solution which is depicted in Fig. 10.47a at time $t = t_n$ at the load level $\lambda = \lambda_n$. Let us assume that the phase field parameter d has reached a predefined threshold value at time t_n. In the next step a refinement is performed by subdividing every selected element into n_T triangular elements Ω^T, as depicted in Fig. 10.47b.[7]

[7] Here a mesh refinement can be employed that uses elements of any shape, see e.g. Fig. 10.39, to refine the mesh around the crack tip adaptively.

As a minimum requirement for the subdivision, the element size h_e of the largest triangular element Ω_e^T has to be less than half of the length scale parameter l. The subdivision can be based on different algorithms. One possibility are the refinement algorithms provided by *Mathematica* using the command `ToElementMesh` with the option `MeshRefinementFunction`, which is very robust. The density of the created mesh can be controlled by the user. After the refinement is finished, a generalized staggered algorithms is applied, as proposed by Hudobivnik (2016), to recover the equilibrium state of the last load step $\lambda = \lambda_n$, see Fig. 10.47c. Once equilibrium is established, the load is increased until the threshold value $d = d_c$ outside the already refined area has been reached. In that case, the refinement procedure (Fig. 10.47a–c is repeated again for every new selected element. Beside the threshold value, the onset of crack growth within already refined elements is another stopping criterion for the load at $\lambda = \lambda_{n+1}$, as depicted in Fig. 10.47d.

The area where the crack has already propagated is defined as the cracked zone. The goal now is to construct a discrete crack path from the cracked zone computed with the phase field approach. To this end, all N nodes with the phase field parameter $d > 0.9$ will be employed to find the new crack tip. The coordinates of the new crack tip are obtained by minimizing the distance problem

$$\boldsymbol{X}_{2,\text{tip}} = \sum_{i=1}^{N} s_i^2 \quad \longrightarrow \quad \min \tag{10.123}$$

where s_i denotes the distance between the corresponding node $\boldsymbol{X}_i$ and its projection on the vector $\boldsymbol{b} = \boldsymbol{X}_{\text{Start}} - \boldsymbol{X}_{2,\text{tip}}$, see Fig. 10.48a. The node with the coordinates being farthest from the previous crack tip $\boldsymbol{X}_{1,\text{tip}}$ will be used as starting vector $\boldsymbol{X}_{\text{Start}}$. The minimization problem (10.123) is solved by the local optimization algorithm `FindMinimum`, provided by *Mathematica*. Once the coordinates of the new crack tip $\boldsymbol{X}_{2,\text{tip}}$ is known, the refined elements from Fig. 10.47c are coarsened to restore the original coarse mesh from Fig. 10.47a. As shown in Fig. 10.49b and c, the new path is introduced as a discrete crack on the coarse mesh Fig. 10.49a by using the cutting algorithms, described in Sect. 10.2. Once again, we have to employ the staggered scheme to recover the equilibrium of the last load step $\lambda = \lambda_{n+1}$ on the coarse mesh. Further crack propagation steps can be performed by repeating the procedure depicted in Fig. 10.47.

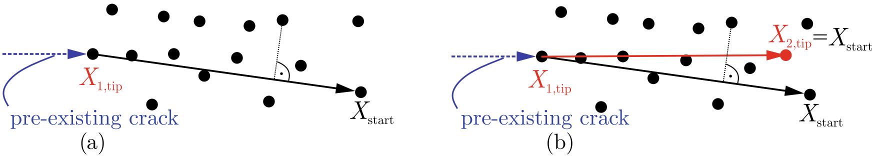
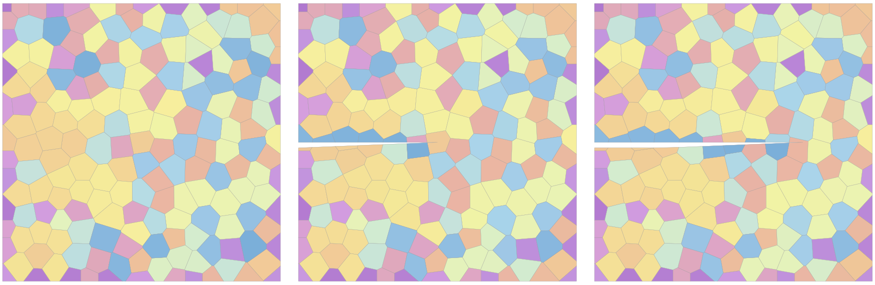


Fig. 10.48 A schematic representation for finding the new crack path by solving the minimization problem in Eq. (10.123)



(a) initial mesh (b) load step $\lambda = \lambda_1$ (c) load step $\lambda = \lambda_2$

Fig. 10.49 The crack evolution for different load steps

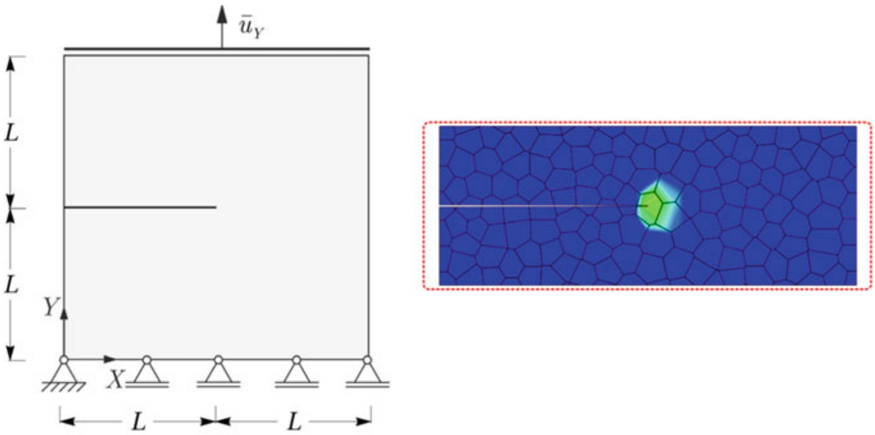
10.6.4 Numerical Examples

Some representative examples illustrate the combination of the phase field method, adaptivity and cutting techniques for the prediction of crack paths. As outlined before, when the phase field parameter d has reached a predefined threshold value, the expected damaged zone is locally refined to improve the accuracy of the solution, see Sect. 10.6.3. Once the direction of the propagation is known, the refined cracked zone is replaced by a discrete crack on the coarse mesh by using the cutting techniques, described in Sect. 10.2.

Investigation of single-edge notched tension test provides the first benchmark example. It considers a square plate with a horizontal notch placed in the middle of the specimen from the left outer surface to the centre. The geometrical setup and the loading conditions of the specimen are depicted in Fig. 10.50a. The size of the square specimen is chosen to be $L = 0.5$ mm. We fixed the bottom edge of the plate in Y-direction and applied a vertical displacement at the top edge until the plate is fully broken.

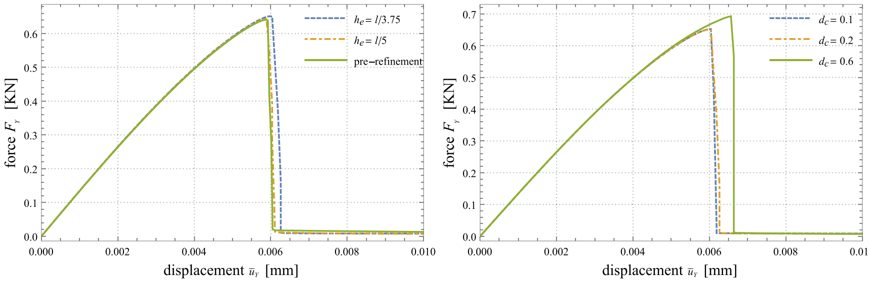
This problem was already discussed in Miehe et al. (2010b) and thus, material parameters used in the simulation are the same as in this reference work. The Lamé constants are $\lambda = 121.1538$ kN/mm^2 and $\mu = 80.7692$ kN/mm^2, the viscosity of the crack propagation is $\eta = 10^{-6}$ kNs/mm^2 and the critical energy release rate is selected as $G_c = 2.7 \times 10^{-3}$ kN/mm. All simulations are performed with the length scale parameter $l = 0.0375$ mm. The domain is discretized by using 364 VEM-Voronoi elements with the mesh size $h_e \approx 2l$. A zoom around the crack is depicted in Fig. 10.50b where elements meet the threshold value in the initial state.

The effect of the element size on the overall structural response is illustrated in Fig. 10.51a by the load-displacement curves in Y-direction for two different mesh sizes with and element size of $h_e \approx L/3.75$ and $h_e \approx L/5$. The results are compared with the load-displacement curve, obtained by a local mesh pre-refinement in the expected fracture zone. The refinement procedure is based on the threshold value $d_c = 0.2$. We observe that the result converges to the pre-refinement curve for decreasing element size h_e.



(a) System (b) Zoom around the crack

Fig. 10.50 Single-edge notched tension test



(a) $h_e \approx l/3.75\,, l/5$ ($d_c = 0.2$) (b) $d_c = 0.1\,, 0.2\,, 0.6$ ($h_e \approx l/3.75$)

Fig. 10.51 Single-edge notched tension test—Load-displacement curves for different parameters

Next, the influence of the threshold value d_c is illustrated in Fig. 10.51b. The values chosen in this analysis are $d_c = 0.1$, $d_c = 0.2$ and $d_c = 0.6$. The refinement is performed with an effective element size of $h_e \approx l/3.75$. The best results are achieved with the threshold values $d_c = 0.1$ and $d_c = 0.2$. The small threshold value ensures that more elements, next to the crack tip, will be refined. Thus, the crack will propagate in the a priori refined zone. It is possible to use a high threshold value, but in such case the predictor-corrector scheme proposed by Heister et al. (2015) should be applied.

Figure 10.52 depicts the contour plots of the fracture phase field d and the crack trajectory for different deformation stages until final failure. The crack phase field initiates at the notch-tip until reaching the threshold value $d_c = 0.2$, as sketched in Fig. 10.50b. Elements that have reached the threshold value were chosen for the refinement. Then, the staggered scheme, proposed by Hudobivnik (2016), is employed to recover the equilibrium state of the last load step $\lambda = \lambda_{n-1}$, see

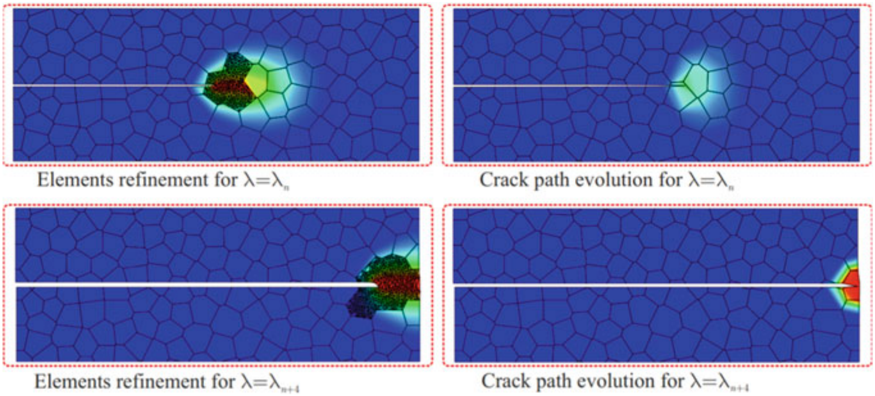

Elements refinement for $\lambda=\lambda_n$
Crack path evolution for $\lambda=\lambda_n$
Elements refinement for $\lambda=\lambda_{n+4}$
Crack path evolution for $\lambda=\lambda_{n+4}$

Fig. 10.52 Single-edge notched tension test—Contour plots of the fracture phase field d and the crack trajectory for different deformation stages until final failure

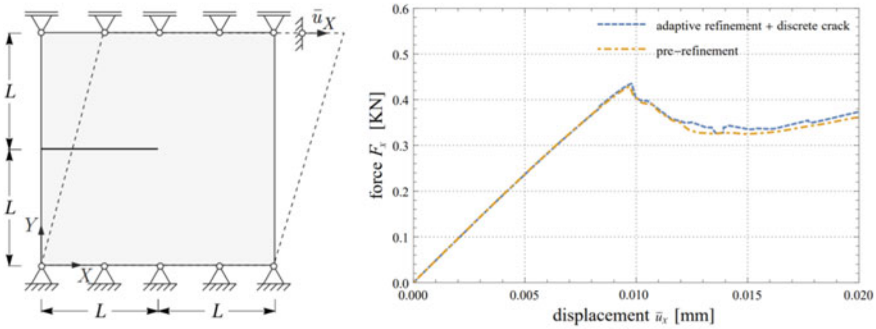
Sect. 10.6.3. By solving the minimization problem (10.123) the discrete crack path is constructed from the cracked zone. As shown in Fig. 10.52, the new path will be introduced as a discrete crack on the original coarse mesh by employing the cutting algorithms, described in Sect. 10.2.

The single-edge notched shear test deals with a single edge cracked plate, fixed at the bottom. The plate is subjected to a pure shear loading, applied at the top edge, using a prescribed displacement $\bar{u}_X$, see Fig. 10.53a. The specimen contains a crack, which is located at the middle height extending from the left outer surface to the centre of the specimen. The size of the square plate is chosen to be $L = 0.5$. The material parameters are the same as in the tension test, see the previous example.

The single-edge notched shear test is a popular benchmark test to investigate the behaviour of the phase field method. For example, Ambati et al. (2015b) used this test to examine the influence of isotropic and the anisotropic formulations of the strain energy function. Furthermore, Ambati et al. (2015b) proposed a hybrid formulation that reduces the computational cost of the phase field method. In this example, the anisotropic model introduced by Miehe et al. (2010a, b) will be applied.

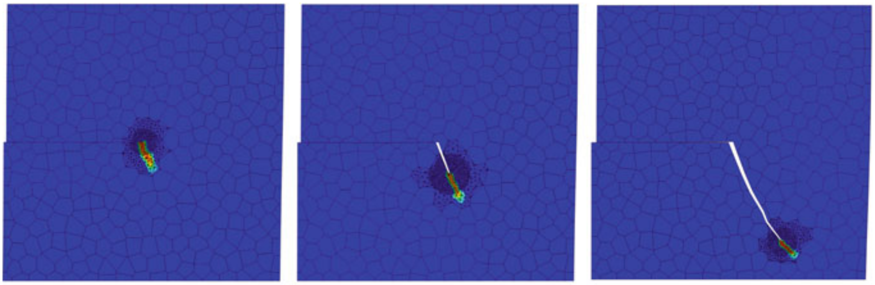
The domain is initially discretized with 364 Voronoi type virtual elements. Local adaptivity around the crack tip is here performed on the basis of a threshold value $d_c = 0.35$ of the phase field parameter. This leads to results which are compared with results computed by a local mesh pre-refinement in the expected fracture zone. The simulations were performed with a length scale parameter $l = 0.015$ mm, and a mesh size of $h_e \approx l/3$, respectively.

Figure 10.53b depicts the load-displacement curves F_X-$\bar{u}_X$ which were simulated using the virtual element formulations with a Voronoi mesh. Since the functional of the potential in Eq. (10.75) is non-convex with respect to the phase field d and the displacement $\boldsymbol{u}$, the curves depict an oscillating behaviour. This can be clearly observed once the prescribed displacement passes the value $\bar{u}_X = 0.01$ mm.



(a) System (b) Load-displacement response

Fig. 10.53 Single-edge notched shear test



(a) Refinement for $\lambda = \lambda_{n+1}$ (b) Refinement for $\lambda = \lambda_{n+3}$ (c) Refinement for $\lambda = \lambda_{n+9}$

Fig. 10.54 Single-edge notched shear test—Contour plots of the fracture phase field d and the crack path evolution at different loading steps

Within the adaptive method one has to transport the internal variables after every refinement and cutting step to the new mesh. Furthermore, equilibrium has to be established for the new mesh. This leads to a small deviation of the calculated residual (force F_X) from the solution with the pre-refined fixed mesh, see the blue dashed line in Fig. 10.53b.

One of the goals when using the adaptive procedure is to capture correctly the direction of the crack propagation together with the crack growth. The capability of the proposed formulation to capture the direction of growth along a curved path can be demonstrated with this example. Figure 10.54 depicts the contour plots of the fracture phase field d and the crack path evolution at different loading steps. To increase the accuracy of the results, the elements adjacent to the threshold elements were included in the refinement procedure. The coarsening technique discussed above was applied for the adjacent elements where the refinement indicator was beyond the d_c due to the mesh cutting. Thus the refinement follows the position of the crack tip during the loading procedure and yields an efficient method that only refines the mesh where $d > d_c$. In total, 12 refinement and cutting steps were needed to predict and to construct the crack path from the centre of the specimen to the bottom edge, as depicted in Fig. 10.54. This path is in good agreement with solutions reported in the literature.

In this section we observed that the coupling of the phase field approach with the capability of the virtual element methodology to cut elements and to use a "hanging" node refinement techniques yields an efficient and accurate discretization technique for crack growth.

References

Aldakheel, F., B. Hudobivnik, A. Hussein, and P. Wriggers. 2018a. Phase-field modeling of brittle fracture using an efficient virtual element scheme. *Computer Methods in Applied Mechanics and Engineering* 341: 443–466.

Aldakheel, F., P. Wriggers, and C. Miehe. 2018b. A modified Gurson-type plasticity model at finite strains: Formulation, numerical analysis and phase-field coupling. *Computational Mechanics* 62 (4): 815–833.

Aldakheel, F., B. Hudobivnik, and P. Wriggers. 2019. Virtual element formulation for phase-field modeling of ductile fracture. *International Journal for Multiscale Computational Engineering* 17: 181–200.

Aldakheel, F., R. Satari, and P. Wriggers. 2021a. Feed-forward neural networks for failure mechanics problems. *Applied Sciences* 11: 6483.

Aldakheel, F., N. Noii, T. Wick, O. Allix, and P. Wriggers. 2021b. Multilevel global-local techniques for adaptive ductile phase-field fracture. *Computer Methods in Applied Mechanics and Engineering* 387: 114175.

Ambati, M., T. Gerasimov, and L. De Lorenzis. 2015a. Phase-field modeling of ductile fracture. *Computational Mechanics* 55 (5): 1017–1040.

Ambati, M., T. Gerasimov, and L. De Lorenzis. 2015b. A review on phase-field models of brittle fracture and a new fast hybrid formulation. *Computational Mechanics* 55 (2): 383–405.

Ambrosio, L., and V.M. Tortorelli. 1990. Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence. *Communications on Pure and Applied Mathematics* 43: 999–1036.

Badnava, H., M.A. Msekh, E. Etemadi, and T. Rabczuk. 2018. An h-adaptive thermo-mechanical phase field model for fracture. *Finite Elements in Analysis and Design* 138: 31–47.

Barsoum, R.S. 1974. Application of quadratic isoparametric finite elements in linear fracture mechanics. *International Journal of Fracture* 10 (4): 603–605.

Bažant, Z.P., and M. Jirasek. 2002. Nonlocal integral formulations of plasticity and damage: Survey of progress. *Journal of Engineering Mechanics* 128: 1119–1149.

Bažant, Z.P., and G. Pijaudier-Cabot. 1989. Measurement of characteristic length of nonlocal continuum. *Journal of Engineering Mechanics* 115: 755–767.

Beirão da Veiga, L., F. Brezzi, and L. Marini. 2013. Virtual elements for linear elasticity problems. *SIAM Journal of Numerical Analysis* 51: 794–812.

Belytschko, T., and T. Black. 1999. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering* 45 (5): 601–620.

Belytschko, T., and M. Tabbara. 1996. Dynamic fracture using element-free Galerkin methods. *International Journal for Numerical Methods in Engineering* 39 (6): 923–938.

Belytschko, T., Y.Y. Lu, and L. Gu. 1994. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering* 37 (2): 229–256.

Belytschko, T., Y.Y. Lu, and L. Gu. 1995. Crack propagation by element-free Galerkin methods. *Engineering Fracture Mechanics* 51 (2): 295–315.

Benedetto, M.F., A. Caggiano, and G. Etse. 2018. Virtual elements and zero thickness interface-based approach for fracture analysis of heterogeneous materials. *Computer Methods in Applied Mechanics and Engineering* 338: 41–67.

Benvenuti, E., A. Chiozzi, G. Manzini, and N. Sukumar. 2022. Extended virtual element method for two-dimensional linear elastic fracture. *Computer Methods in Applied Mechanics and Engineering* 390: 114352.

Biabanaki, S.O.R., A.R. Khoei, and P. Wriggers. 2014. Polygonal finite element methods for contact-impact problems on non-conformal meshes. *Computer Methods in Applied Mechanics and Engineering* 269: 198–221.

Bittencourt, T.N., P.A. Wawrzynek, A.R. Ingraffea, and J.L. Sousa. 1996. Quasi-automatic simulation of crack propagation for 2d lefm problems. *Engineering Fracture Mechanics* 55 (2): 321–334.

Borden, M.J., T.J.R. Hughes, C.M. Landis, and C.V. Verhoosel. 2014. A higher-order phase-field model for brittle fracture: Formulation and analysis within the isogeometric analysis framework. *Computer Methods in Applied Mechanics and Engineering* 273: 100–118.

Bouchard, P., F. Bay, Y. Chastel, and I. Tovena. 2000. Crack propagation modelling using an advanced remeshing technique. *Computer Methods in Applied Mechanics and Engineering* 189 (3): 723–742.

Bourdin, B., G. Francfort, and J.J. Marigo. 2000. Numerical experiments in revisited brittle fracture. *Journal of the Mechanics and Physics of Solids* 48 (4): 797–826.

Chan, S.K., I.S. Tuba, and W.K. Wilson. 1970. On the finite element method in linear fracture mechanics. *Engineering Fracture Mechanics* 2 (1): 1–17.

Coleman, B.D., and W. Noll. 1963. The thermodynamics of elastic materials with heat conduction and viscosity. *Archives for Rational Mechanics and Analysis* 13: 167–178.

Dai, S., C. Augarde, C. Du, and D. Chen. 2015. A fully automatic polygon scaled boundary finite element method for modelling crack propagation. *Engineering Fracture Mechanics* 133: 163–178.

De Bellis, M., P. Wriggers, B. Hudobivnik, and G. Zavarise. 2018. Virtual element formulation for isotropic damage. *Finite Element Analysis and Design* 144: 38–48.

de Borst, R. 1991. Simulation of strain localization: A reappraisal of the cosserat continuum. *Engineering Computations* 8: 317–332.

de Borst, R., L.J. Sluys, H.B. Muhlhaus, and J. Pamin. 1993. Fundamental issues in finite element analyses of localization of deformation. *Engineering Computations* 10: 99–121.

Erdogan, F., and G.C. Sih. 1963. On the crack extension in plates under plane loading and transverse shear. *Journal of Basic Engineering* 85 (4): 519–525.

Eshelby, J. 1974. The calculation of energy release rates. *Prospects of Fracture Mechanics*, 69–84.

Fleming, M., Y.A. Chu, B. Moran, and T. Belytschko. 1998. Enriched element-free Galerkin methods for crack tip fields. *International Journal for Numerical Methods in Engineering* 40 (8): 1483–1504.

Francfort, G., and J.J. Marigo. 1998. Revisiting brittle fracture as an energy minimization problem. *Journal of the Mechanics and Physics of Solids* 46 (8): 1319–1342.

Govindjee, S., G.J. Kay, and J.C. Simo. 1995. Anisotropic modelling and numerical simulation of brittle damage in concrete. *International Journal for Numerical Methods in Engineering* 38: 3611–3633.

Gross, D., and T. Seelig. 2017. *Fracture mechanics: With an introduction to micromechanics*, 3rd ed. Heidelberg: Springer.

Heister, T., M.F. Wheeler, and T. Wick. 2015. A primal-dual active set method and predictor-corrector mesh adaptivity for computing fracture propagation using a phase-field approach. *Computer Methods in Applied Mechanics and Engineering* 290: 466–495.

Hellen, T.K. 1975. On the method of virtual crack extensions. *International Journal for Numerical Methods in Engineering* 9 (1): 187–207.

Henshell, R.D., and K.G. Shaw. 1975. Crack tip finite elements are unnecessary. *International Journal for Numerical Methods in Engineering* 9 (3): 495–507.

Hudobivnik, B. 2016. Automatic differentiation based solution of strongly coupled problems in engineering. Ph.D. thesis, University of Ljubljana, Faculty of Civil and Geodetic Engineering.

Hussein, A., F. Aldakheel, B. Hudobivnik, P. Wriggers, P.A. Guidault, and O. Allix. 2019. A computational framework for brittle crack propagation based on an efficient virtual element method. *Finite Elements in Analysis and Design* 159: 15–32.

Hussein, A., B. Hudobivnik, and P. Wriggers. 2020. A combined adaptive phase field and discrete cutting method for the prediction of crack paths. *Computer Methods in Applied Mechanics and Engineering* 372: 113329.

Ingraffea A.R., G.E. Blandford, and J.A. Liggett. 1983. Automatic modelling of mixed-mode fatigue and quasi-static crack propagation using the boundary element method. In *Fracture mechanics: Fourteenth symposium-volume I: Theory and analysis*. ASTM International.

Irwin, G.R. 1956. Onset of fast crack propagation in high strength steel and aluminum alloys. *Sagamore Research Conference Proceedings* 2: 289–305.

Kachanov, L.M. 1958. Time of the rupture process under creep conditions. *Izvestia Akademii Nauk. S.S.R. Otd. Tech. Nauk* 8: 26–31.

Karihaloo, B.L., and Q.Z. Xiao. 2001. Accurate determination of the coefficients of elastic crack tip asymptotic field by a hybrid crack element with p-adaptivity. *Engineering Fracture Mechanics* 68 (15): 1609–1630.

Khoei, A.R., R. Yasbolaghi, and S.O.R. Biabanaki. 2015. A polygonal finite element method for modeling crack propagation with minimum remeshing. *International Journal of Fracture* 194 (2): 123–148.

Klinsmann, M., D. Rosato, M. Kamlah, and R.M. McMeeking. 2015. An assessment of the phase field formulation for crack growth. *Computer Methods in Applied Mechanics and Engineering* 294: 313–330.

Korelc, J., and P. Wriggers. 2016. *Automation of finite element methods*. Berlin: Springer.

Kuna, M. 2013. *Finite elements in fracture mechanics*. Springer.

Kuna, M., and M. Zwicke. 1990. A mixed hybrid finite element for three-dimensional elastic crack analysis. *International Journal of Fracture* 45 (1): 65–79.

Lemaitre, J. 1996. *A course on damage mechanics*. Berlin, New York: Springer.

Lemaitre, J., R. Desmorat, and M. Sauzay. 2001. Anisotropic damage law of evolution. *European Journal of Mechanics - A/Solids* 19: 187–208.

Lubliner, J., J. Oliver, S. Oller, and E.O. Nate. 1989. A plastic-damage model for concrete. *International Journal of Solids and Structures* 25: 299–326.

Malvern, L.E. 1969. *Introduction to the mechanics of a continuous medium*. Englewood Cliffs, New Jersey: Prentice-Hall.

Miehe, C., and E. Gürses. 2010. A robust algorithm for configurational-force-driven brittle crack propagation with r-adaptive mesh alignment. *International Journal for Numerical Methods in Engineering* 72 (2): 127–155.

Miehe, C., E. Gürses, and M. Birkle. 2007. A computational framework of configurational-force-driven brittle fracture based on incremental energy minimization. *International Journal of Fracture* 145 (4): 245–259.

Miehe, C., M. Hofacker, and F. Welschinger. 2010a. A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits. *Computer Methods in Applied Mechanics and Engineering* 199 (45–48): 2765–2778.

Miehe, C., F. Welschinger, and M. Hofacker. 2010b. Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field fe implementations. *International Journal for Numerical Methods in Engineering* 83 (10): 1273–1311.

Moës, N., J. Dolbow, and T. Belytschko. 1999. A finite element method for crack growth without remeshing. *International journal for Numerical Methods in Engineering* 46 (1): 131–150.

Mumford, D., and J. Shah. 1989. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics* 42: 577–685.

Nagaraja, S., M. Elhaddad, M. Ambati, S. Kollmannsberger, L. De Lorenzis, and E. Rank. 2019. Phase-field modeling of brittle fracture with multi-level hp-fem and the finite cell method. *Computational Mechanics* 63 (6): 1283–1300.

Needleman, A. 1988. Material rate dependence and mesh sensitivity in localization problems. *Computer Methods in Applied Mechanics and Engineering* 67: 69–85.

Nguyen-Thanh, V.M., X. Zhuang, H. Nguyen-Xuan, T. Rabczuk, and P. Wriggers. 2018. A virtual element method for 2d linear elastic fracture analysis. *Computer Methods in Applied Mechanics and Engineering* 340: 366–395.

Noii, N., F. Aldakheel, T. Wick, and P. Wriggers. 2020. An adaptive global-local approach for phase-field modeling of anisotropic brittle fracture. *Computer Methods in Applied Mechanics and Engineering* 361: 112744.

Nuismer, R.J. 1975. An energy release rate criterion for mixed mode fracture. *International Journal of Fracture* 11: 245–250.

Oliver, J., M. Cervera, S. Oller, and J. Lubliner. 1990. Isotropic damage models and smeared crack analysis of concrete. In *Proceedings SCI-C Computer Aided Analysis and Design of Concrete Structures*, ed. N. Bizanic and H. Mang. Pineridge Press.

Oliver, J. 1989. A consistent characteristic length for smeared cracking models. *International Journal for Numerical Methods in Engineering* 28: 461–474.

Ooi, E.T., and Z.J. Yang. 2009. Modelling multiple cohesive crack propagation using a finite element, a scaled boundary finite element coupled method. *Engineering Analysis with Boundary Elements* 33 (7): 915–929.

Ooi, E.T., C. Song, F. Tin-Loi, and Z.J. Yang. 2012. Automatic modelling of cohesive crack propagation in concrete using polygon scaled boundary finite elements. *Engineering Fracture Mechanics* 93: 13–33.

Paris, P.C., and G.C. Sih. 1965. Stress analysis of cracks. In *Fracture toughness testing and its applications*. ASTM International.

Pezeshki, M., S. Loehnert, P. Wriggers, P.A. Guidault, and E. Baranger. 2018. 3d dynamic crack propagation by the extended finite element method and a gradient-enhanced damage model. In *Multiscale modeling of heterogeneous structures*, ed. J. Sorić, P. Wriggers, and O. Allix, 277–299.

Pijaudier-Cabot, G., and Z.P. Bažant. 1987. Nonlocal damage theory. *Journal of Engineering Mechanics* 113: 1512–1533.

Portela, A., M.H. Aliabadi, and D.P. Rooke. 1992. The dual boundary element method: Effective implementation for crack problems. *International Journal for Numerical Methods in Engineering* 33 (6): 1269–1287.

Ramm, E., E. Rank, R. Rannacher, K. Schweizerhof, E. Stein, W. Wendland, G. Wittum, P. Wriggers, and W. Wunderlich. 2003. *Error-controlled adaptive finite elements in solid mechanics*. Chichester: Wiley.

Rice, J.R. 1968. A path independent integral and the approximate analysis of strain concentration by notches and cracks. *Journal of Applied Mechanics* 35 (2): 379–386.

Rice, J.R., and G.F. Rosengren. 1968. Plane strain deformation near a crack tip in a power-law hardening material. *Journal of the Mechanics and Physics of Solids* 16 (1): 1–12.

Rocco, C., G.V. Guinea, J. Planas, and M. Elices. 1999a. Mechanism of rupture in splitting test. *Materials Journal* 96 (1): 52–60.

Rocco, C., G.V. Guinea, J. Planas, and M. Elices. 1999b. Size effect and boundary conditions in the brazilian test: Experimental verification. *Materials and Structures* 32 (3): 210.

Rocco, C., G.V. Guinea, J. Planas, and M. Elices. 1999c. Size effect and boundary conditions in the brazilian test: Theoretical analysis. *Materials and Structures* 32 (6): 437.

Scotta, R., R. Vitaliani, A. Saetta, E. Oñate, and A. Hanganu. 2001. A scalar damage model with a shear retention factor for the analysis of reinforced concrete structures: Theory and validation. *Computers & Structures* 79: 737–755.

Sih, G.C., P.C. Paris, and F. Erdogan. 1962. Crack-tip, stress-intensity factors for plane extension and plate bending problems. *Journal of Applied Mechanics* 29 (2): 306–312.

Simo, J.C., and J.W. Ju. 1987. Strain- and stress-based continuum damage models–I. Formulation. *International Journal of Solids and Structures* 23: 821–840.

Song, C., E.T. Ooi, and S. Natarajan. 2018. A review of the scaled boundary finite element method for two-dimensional linear elastic fracture mechanics. *Engineering Fracture Mechanics* 187: 45–73.

Stolarska, M., D.L. Chopp, N. Moës, and T. Belytschko. 2001. Modelling crack growth by level sets in the extended finite element method. *International Journal for Numerical Methods in Engineering* 51 (8): 943–960.

Sukumar, N. 2004. Construction of polygonal interpolants: A maximum entropy approach. *International Journal for Numerical Methods in Engineering* 61 (12): 2159–2181.

Sukumar, N., and E.A. Malsch. 2006. Recent advances in the construction of polygonal finite element interpolants. *Archives of Computational Methods in Engineering* 13 (1): 129.

Sukumar, N., and J.H. Prévost. 2003. Modeling quasi-static crack growth with the extended finite element method part i: Computer implementation. *International Journal of Solids and Structures* 40 (26): 7513–7537.

Tong, P., T.H.H. Pian, and S.J. Lasry. 1973. A hybrid-element approach to crack problems in plane elasticity. *International Journal for Numerical Methods in Engineering* 7 (3): 297–308.

Williams, M.L. 1957. On the stress distribution at the base of a stationary crack. *Journal of Applied Mechanics* 24: 109–114.

Wriggers, P., B. Reddy, W. Rust, and B. Hudobivnik. 2017. Efficient virtual element formulations for compressible and incompressible finite deformations. *Computational Mechanics* 60: 253–268.

Yang, Z.J., and A.J. Deeks. 2007. Fully-automatic modelling of cohesive crack growth using a finite element, a scaled boundary finite element coupled method. *Engineering Fracture Mechanics* 74 (16): 2547–2573.

Yau, J.F., S.S. Wang, and H.T. Corten. 1980. A mixed-mode crack analysis of isotropic solids using conservation laws of elasticity. *Journal of Applied Mechanics* 47 (2): 335–341.

Chapter 11
Virtual Element Formulation for Contact

Almost all engineering applications include structural parts that act on other elements through an area of contact. In many cases the behaviour at the contact interface has an influence on the performance of machines, looking for example at friction and wear, or on production processes like metal forming. Therefore it is necessary to capture these effects in predictive numerical simulations during the design. Thus, starting with the seminal work of Coulomb (1785) on frictional contact and the work of Hertz (1882) and Mindlin (1949) on the analytical solution of contact, scientists have developed numerous constitutive relations and analytical approaches for contact problems, see e.g. Barber (2018).

In solid mechanics the finite element method (FEM) is mainly employed for numerical simulations of contact problems. It has a history of about 50 years. Different approaches and discretization techniques where developed in this period by many scientists. An overview can be found in the textbooks of Laursen (2002) and Wriggers (2006).

Generally, contact falls in the category of unilateral problems, meaning that no penetration can take place and thus the deformation is restricted. The related mathematical formulations have to be treated as constraint optimization problems since the contact area is a priori unknown. Within this methodology one has to enforce contact constraints. This means that the surfaces in contact have to be coupled and contact tractions need to be transmitted at the interface.

In this chapter several formulations for the treatment of contact problems will be developed in the light of the virtual element method and some specific features of VEM will be explored that lead to simplification and enhancement of discretization schemes for contact. The schemes related to the virtual element method are based on the treatments in Wriggers et al. (2016), Wriggers and Rust (2019), Aldakheel et al. (2020) and Cihan et al. (2022).

P. Wriggers et al., *Virtual Element Methods in Engineering Sciences*,
https://doi.org/10.1007/978-3-031-39255-9_11

11.1 Introduction

Many different approaches were developed over the last five decades to formulate contact constraints for numerical simulation methods. In the finite element approach discretizations schemes relied on so called node-to-node contact, node-to-segment contact and segment-to-segment contact, see Fig. 11.1 which describe different possibilities to discretize contact constraints:

- **Node-to-node contact.** The first papers on contact discretizations were for problems with small elastic strains and deformations. Thus a contact formulation was possible that introduced a pair of nodes at two contacting bodies which during deformation could come in contact, see e.g. Wilson and Parsons (1970), Chan and Tuba (1971) and Hughes et al. (1976).
- **Node-to-segment contact.** In order to discretize contact constraints for finite deformations a node-to-segment contact was developed which allows large sliding over several elements, see e.g. Hallquist (1984), Wriggers and Simo (1985) and Hallquist et al. (1985). These formulations were enhanced by smooth interpolations of contact surfaces that introduce an increased continuity, e.g. C^1-continuity, at the contacting surfaces, see e.g. Pietrzak and Curnier (1997), Padmanabhan and Laursen (2001), Wriggers et al. (2001) and Krstulovic-Opara et al. (2002). C^1-continuous methods perform better during the solution of the contact problem, especially for sliding, but do not increase the convergence rate that still is related to the underlying finite element mesh.
- **Segment-to-segment contact.** Based on the early work by Simo et al. (1985) segment-to-segment discretization schemes were developed for two-and three-dimensional contact problems. These are now called mortar methods, see Belgacem et al. (1997), who coined the name. Mortar methods have been favourised for the discretization of contact constraints and can be found in e.g. Wohlmuth (2000), McDevitt and Laursen (2000), Wohlmuth and Krause (2004), Puso (2004), Puso and Laursen (2004), Flemisch et al. (2005), Fischer and Wriggers (2005, 2006), Tur et al. (2009), Popp et al. (2010) and Laursen et al. (2012). Mortar methods discretize the contact constraints in a weak sense and thus provide a stable interpolation scheme for contact constraints.
- **Higher order contact.** Another approach to model contact is provided within the isogeometric analysis (IGA), see Cottrell et al. (2009). Contact formulations using IGA yield highly robust schemes but need considerable effort when implementing these discretizations, see e.g. Temizer et al. (2011), De Lorenzis et al. (2011, 2012)

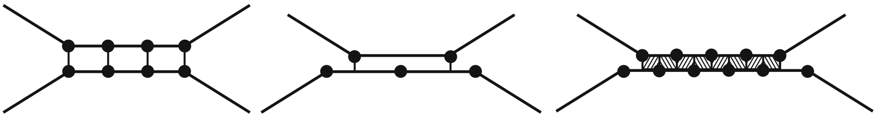


Fig. 11.1 Node-to-node, node-to-segment and segment-to-segment contact discretizations

> and Temizer et al. (2012). Collocation methods were developed to overcome related shortcomings, see e.g. De Lorenzis et al. (2015) and Kruse et al. (2015).

The virtual element method (VEM), described with its basic features in Chaps. 3 and 5, is a flexible polygonal method which allows arbitrary number of nodes at an element, independent on the ansatz space. This feature is a key for a simple contact formulation in which nodes are inserted at points that are necessary to enforce the contact constraints correctly. In this way a node-to-node contact formulation can be generated, even for non-matching meshes and large sliding situations. A rather simple algorithm transforms the contacting VEM meshes in matching meshes where non-matching interfaces occur. Note that this scheme can be employed for virtual elements with higher order ansatz spaces resulting in a higher order approximation.The associated contact formulations for the virtual element method will be discussed in the following sections.

11.2 Theoretical Background for Contact of Solids

The contacting continua are chosen in a first step as elastic bodies. The domains Ω^δ ($\delta = 1, 2$) of the contacting bodies are bounded by Γ^δ in the initial configuration. In addition to the balance of momentum, see (2.19) and the Dirichlet, see (2.20), and Neumann boundary conditions, see (2.21), constraint conditions have to be defined within the contact interface Γ_c in the current configuration.

11.2.1 Local Contact Kinematics

All contact conditions are derived for the case of finite deformations to have the most general description. Simplifications for small deformations are introduced when necessary.

To set up local contact conditions we describe the surface of the bodies by convective coordinates and define the surface of the contacting body, Ω^1, as master or reference surface. The surface of the other body is called slave surface.[1] Figure 11.2 shows the parameterization of the contact surface. The coordinates of the current configuration $\boldsymbol{x}^\delta = \boldsymbol{\varphi}^\delta(\boldsymbol{X}^\delta)$ are written for body Ω^δ ($\delta = 1, 2$) in terms of the displacements $\boldsymbol{x}^\delta = \boldsymbol{X}^\delta + \boldsymbol{u}^\delta$ where the coordinates $\boldsymbol{X}^\delta$ are related to the reference configuration and $\boldsymbol{u}^\delta$ is the displacement field. The normal vector $\boldsymbol{n}^1$ is associated with the master surface which is located within the possible contact surface $\varphi^1(\Gamma^1)$.

[1] This choice is arbitrary and completely interchangeable since in the final solution of the contact problem all geometrical quantities of both surfaces (e.g., the normal and tangent vectors) coincide.

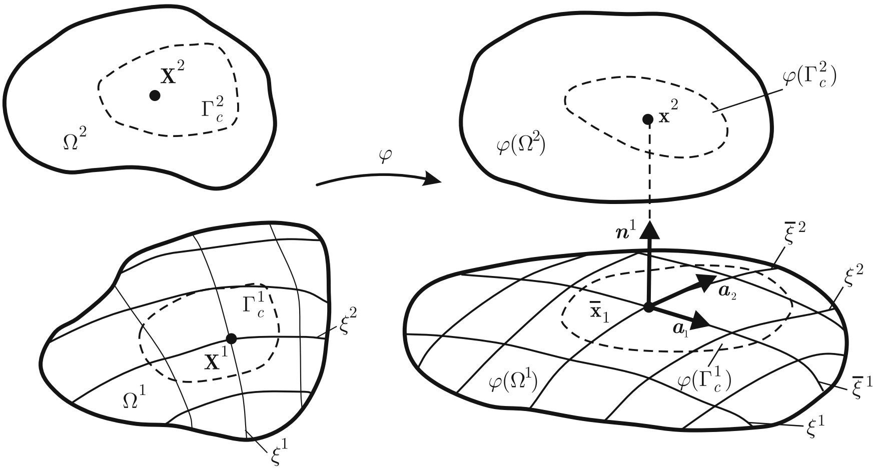


Fig. 11.2 Definition of the gap for finite deformations

Normal Contact. Assuming that the contact boundary describes, at least locally, a convex region we can relate to every point $\boldsymbol{x}^2$ on $\varphi^2(\Gamma^2)$ a point $\bar{\boldsymbol{x}}^1 = \boldsymbol{x}^1(\bar{\boldsymbol{\xi}})$ on $\varphi^1(\Gamma^1)$ via the minimal distance problem, see Fig. 11.2,

$$\|\boldsymbol{x}^2 - \bar{\boldsymbol{x}}^1\| = \min_{\boldsymbol{x}^1 \subseteq \varphi^1(\Gamma^1)} \|\boldsymbol{x}^2 - \boldsymbol{x}^1(\boldsymbol{\xi})\| \,, \tag{11.1}$$

where $\boldsymbol{\xi} = (\xi^1, \xi^2)$ are the convective coordinates that parameterize the boundary Γ^1. The minimization process yields the—generally nonlinear—conditions ($\alpha \in \{1, 2\}$)

$$\left[\boldsymbol{x}^2 - \boldsymbol{x}^1(\bar{\boldsymbol{\xi}})\right] \cdot \bar{\boldsymbol{a}}_\alpha^1 = 0 \tag{11.2}$$

from which $\bar{\boldsymbol{\xi}}$ and the associated normal $\bar{\boldsymbol{n}}^1$ can be obtained.[2] The bar denotes the solution point for the minimum distance problem. The normal vector $\bar{\boldsymbol{n}}^1$ can either be defined by

$$\bar{\boldsymbol{n}}^1 = \frac{\boldsymbol{x}^2 - \boldsymbol{x}^1(\bar{\boldsymbol{\xi}})}{\|\boldsymbol{x}^2 - \boldsymbol{x}^1(\bar{\boldsymbol{\xi}})\|} \quad \text{or } \bar{\boldsymbol{n}}^1 = \frac{\bar{\boldsymbol{a}}_1^1 \times \bar{\boldsymbol{a}}_2^1}{\|\bar{\boldsymbol{a}}_1^1 \times \bar{\boldsymbol{a}}_2^1\|}$$

which is obvious from Fig. 11.2. However for computational purposes the second definition has the advantage to be more stable since for a vanishing gap the nominator and denominator in the first definition approach zero.

Once the point $\bar{\boldsymbol{x}}^1$ is known, we can write the geometrical constraint inequality which prevents penetration of one body into the other

[2] The normal and the coordinate $\bar{\boldsymbol{\xi}}$ change when a point $\boldsymbol{x}^2$ is sliding along the surface $\varphi^1(\Gamma_c^1)$. This has to be taken into account for finite deformations when computing residuals and tangent matrices of the dicretization scheme.

$$g_N = (\boldsymbol{x}^2 - \bar{\boldsymbol{x}}^1) \cdot \bar{\boldsymbol{n}}^1 \geq 0 \tag{11.3}$$

where g_N is the gap at the minimum distance point

For small deformations the projection (11.1) can be evaluated at the initial configuration and the normal vector in the deformed state $\bar{\boldsymbol{n}}^1$ can be expressed by the normal vector in the undeformed state: $\bar{\boldsymbol{N}}^1 = \bar{\boldsymbol{n}}^1$ evaluated at the a fixed projection point $\bar{\xi}$. Furthermore, relations $\boldsymbol{x}^\delta = \boldsymbol{X}^\delta + \boldsymbol{u}^\delta$ can be inserted in (11.3) which yields

$$g_N = (\boldsymbol{u}^2 - \bar{\boldsymbol{u}}^1) \cdot \bar{\boldsymbol{N}}^1 + g_{No} \geq 0 \tag{11.4}$$

with the initial gap $g_{No} = (\boldsymbol{X}^2 - \bar{\boldsymbol{X}}^1) \cdot \bar{\boldsymbol{N}}^1$. Thus, once the projection is computed using (11.2) for the coordinates in the initial configuration, the constraint inequality can be expressed in terms of the displacements for a fixed normal $\bar{\boldsymbol{N}}^1$.

Tangential Contact. In case of frictional sliding it is necessary to compute the relative tangential displacement between the two contacting solids at the contact interface. Starting from the orthogonality condition (11.2) time differentiation yields for $g_N = 0$ the change of the projection point $\bar{\boldsymbol{\xi}}$ in time due to the relative velocities $\boldsymbol{v}^2$ and $\bar{\boldsymbol{v}}^1$ at that point, see e.g. Wriggers (2006) and De Lorenzis et al. (2017),

$$\dot{\bar{\xi}}^\beta = \bar{a}^{\alpha\beta} \, (\boldsymbol{v}^2 - \bar{\boldsymbol{v}}^1) \cdot \bar{\boldsymbol{a}}^1_\alpha \tag{11.5}$$

where $\bar{a}^{\alpha\beta}$ is the contravariant metric tensor of the contact surface $\varphi^1(\Gamma^1_c)$ at the projection point.This tensor is related to the covariant metric tensor $\bar{a}_{\alpha\beta} = \bar{\boldsymbol{a}}^1_\alpha \cdot \bar{\boldsymbol{a}}^1_\beta$ by $\bar{a}^{\alpha\beta} = (\bar{a}_{\alpha\beta})^{-1}$.

Based on this preliminary definition it is possible to distinguish the stick and slip case which can occur at the contact interface.

- **Stick case.** During stick there is no relative tangential motion at the contact interface and hence the change of the convective coordinate of a projection point is zero since both velocities $\boldsymbol{v}^2$ and $\bar{\boldsymbol{v}}^1$ are the same. Thus the stick condition can be simply stated as
$$\dot{\xi}^\beta = 0\,. \tag{11.6}$$

- **Slip case.** Slip describes the relative motion of the two contacting bodies within the contact interface and the projection point changes with time, see (11.5). The tangential relative displacement, needed to compute the sliding path of point $\boldsymbol{x}^2$ on the contact surface $\varphi^1(\Gamma^1_c)$ can be defined by the increment
$$\mathrm{d}\boldsymbol{g}_T = \bar{\boldsymbol{a}}^1_\alpha \, \mathrm{d}\xi^\alpha = \bar{\boldsymbol{x}}^1_{,\alpha} \, \mathrm{d}\xi^\alpha = \bar{\boldsymbol{x}}^1_{,\alpha} \, \dot{\bar{\xi}}^\alpha \, \mathrm{d}t \tag{11.7}$$
With the incremental path length $\mathrm{d}g_T = \|\mathrm{d}\boldsymbol{g}_T\|$ the complete sliding path of a point $\boldsymbol{x}^2$ in a given time interval $(t_0\,, t)$ can be determined by

$$g_T = \int_{t_0}^{t} \|\bar{\boldsymbol{x}}^1_{,\alpha}\, \dot{\bar{\xi}}^\alpha\| \, \mathrm{d}\tau \tag{11.8}$$

Note that $\dot{\bar{\xi}}^\alpha$ is already given in (11.5) in terms of the velocities $\boldsymbol{v}^2 = \dot{\boldsymbol{x}}^2$ and $\bar{\boldsymbol{v}}^1 = \dot{\bar{\boldsymbol{x}}}^1$. When inserting this expression in (11.8) it follows for the term in the norm $\bar{\boldsymbol{x}}^1_{,\alpha}\, \bar{a}^{\alpha\beta}\, (\boldsymbol{v}^2 - \bar{\boldsymbol{v}}^1) \cdot \bar{\boldsymbol{a}}^1_\beta = \bar{\boldsymbol{a}}^1_\alpha\, \bar{a}^{\alpha\beta}\, (\boldsymbol{v}^2 - \bar{\boldsymbol{v}}^1) \cdot \bar{\boldsymbol{a}}^1_\beta$ which can be rewritten as

$$\bar{\boldsymbol{x}}^1_{,\alpha}\, \dot{\bar{\xi}}^\alpha = \mathbb{P}_T\, (\boldsymbol{v}^2 - \bar{\boldsymbol{v}}^1)$$

with the projector onto the tangent plane: $\mathbb{P}_T = \bar{a}^{\alpha\beta}\, \bar{\boldsymbol{a}}^1_\alpha \otimes \bar{\boldsymbol{a}}^1_\beta$. Thus (11.8) can also be expressed by

$$g_T = \int_{t_0}^{t} \|\, \mathbb{P}_T\, (\boldsymbol{v}^2 - \bar{\boldsymbol{v}}^1)\| \, \mathrm{d}\tau \tag{11.9}$$

which states that the norm of the relative tangential velocity at the projection point $\bar{\boldsymbol{\xi}}$ in the contact interface has to be integrated to obtain the sliding path.
The tangential relative velocity follows from (11.7)

$$\dot{\boldsymbol{g}}_T = \dot{\bar{\xi}}^\alpha\, \bar{\boldsymbol{x}}^1_{,\alpha} = \dot{\bar{\xi}}^\alpha\, \bar{\boldsymbol{a}}^1_\alpha \tag{11.10}$$

which yields with the identities above the equivalent form $\dot{\boldsymbol{g}}_T = \boldsymbol{v}^2 - \bar{\boldsymbol{v}}^1$.
For small deformation (11.9) can be used without changes, only the projection point has to be computed with respect to the initial configuration.

11.2.2 Constitutive Relations for Contact

Many constitutive equations exist at the contact interface. For normal contact these span the arch from non-smooth unilateral conditions via micromechanically motivated equations to approximate interface laws that are introduced for algorithmic purposes, see e.g. Kikuchi and Oden (1988) and Wriggers (2006). Here we will restrict ourselves to non-smooth contact and associated regularization methods.

For the tangential contact the number of constitutive relations is even broader due to many possible material pairings in the contact area. Non-smooth and regularized formulations are available to describe frictional behaviour as well as constitutive relations that derive from micromechanical observations. Lately, multi-scale and homogenization approaches were pursued in numerical schemes, see e.g. Bandeira et al. (2004), Temizer and Wriggers (2008), Wriggers and Reinelt (2009) and Wagner et al. (2015) for different material pairings.

Normal contact. The inequality contact constraint (11.3) has to be complemented by a condition related to the contact pressure since classically adhesion is not considered at the contact interface. Hence only compression occurs at the contact area. This yields an inequality for the normal traction

$$t_N \leq 0\,. \tag{11.11}$$

Equations (11.3) and (11.11) describe the following states:

- **contact:** $t_N \leq 0$ and $g_N = 0$ or
- **gap:** $g_N \geq 0$ and $t_N = 0$.

Note, that the relation $t_N\, g_N = 0$ is fulfilled for both states. This leads to the so called Karush–Kuhn–Tucker conditions[3]

$$g_N \geq 0\,, \quad t_N \leq 0 \quad \text{and} \quad t_N\, g_N = 0 \tag{11.12}$$

that are the basis for the unilateral contact formulation. The contact case $g_N = 0$ can be incorporated in the potential of a contact problem by a Lagrangian multiplier. Other possibilities will be discussed in Sect. 11.2.3.

Tangential contact. In case of tangential contact one has to distinguish between stick and slip. Stick is classically modelled by the constraint provided in (11.6). This means that the projection $\bar{\boldsymbol{\xi}}$ does not change in time and hence the positions of the contacting points $\boldsymbol{x}^2$ and $\bar{\boldsymbol{x}}^1$ assume the same location at the contact interface leading to the alternative constraint $\boldsymbol{x}^2 - \bar{\boldsymbol{x}}^1 = \mathbf{0}$

$$\dot{\bar{\boldsymbol{\xi}}} = \mathbf{0} \Longleftrightarrow \boldsymbol{x}^2 - \bar{\boldsymbol{x}}^1 = \mathbf{0} \tag{11.13}$$

Such a constraint is added to the potential energy of a contact problem via Lagrange multipliers. Additional formulations that relax the above constraint can be found in Sect. 11.2.3.

Sliding occurs at the interface once the tangential force bypasses a limit. The law of Coulomb is mostly applied. It introduces the limit $\|\boldsymbol{t}_T\| > \mu\, |t_N|$ for sliding where the friction coefficient μ is a constitutive parameter which depends on the material pairing. Based on this assumption the stick-slip behaviour at the contact interface can be described

- **stick:**

$$\|\boldsymbol{t}_T\| \leq \mu\, |t_N| \quad \text{and} \quad \dot{\bar{\boldsymbol{\xi}}} = \mathbf{0} \tag{11.14}$$

- **slip:**

$$\|\boldsymbol{t}_T\| > \mu\, |t_N|\,, \quad \dot{\bar{\boldsymbol{\xi}}} \neq \mathbf{0} \quad \text{and} \quad \boldsymbol{t}_T = \mu\, |t_N|\, \frac{\dot{\boldsymbol{g}}_T}{\|\dot{\boldsymbol{g}}_T\|}\,. \tag{11.15}$$

Alternatively, a regularized form of the stick-slip behaviour was firstly introduced in Wriggers (1987). It is based on the analogy of the friction model with

[3] These conditions are known from the theory of optimization. In classical contact mechanics they denoted by the name Hertz–Signorini–Moreau conditions, see e.g. Franke et al. (2010).

non-associated plasticity. In this case a small elastic tangential motion is allowed, regularizing the stick. The tangential motion is then split into an stick (elastic) and sliding (plastic) part which leads to the kinematical relation

$$\boldsymbol{g}_T^e = \boldsymbol{g}_T - \boldsymbol{g}_T^s \tag{11.16}$$

where the superscript e stands for elastic motion (stick) and s for plastic sliding.

The simplest possible model for the elastic part of the tangential contact is an isotropic linear elastic relation which yields

$$\boldsymbol{t}_T = c_T\, \boldsymbol{g}_T^e \tag{11.17}$$

where c_T is the elastic constant.

The plastic, frictional tangential slip $\boldsymbol{g}_T^s$ is governed by a constitutive evolution equation, see Wriggers (1987) or Giannokopoulos (1989). This relation can be stated, like (2.69) in elasto-plasticity, as

$$\dot{\boldsymbol{g}}_T^s = \dot{\gamma}\, \frac{\partial f_s}{\partial \boldsymbol{t}_T}\,. \tag{11.18}$$

The flow rule f_s in (11.18) is for Coulomb friction given by

$$f_s(\boldsymbol{t}_T) = \parallel \boldsymbol{t}_T \parallel - \mu\, |t_N| \leq 0 \tag{11.19}$$

which defines the plastic slip criterion function. Additionally we have the loading-unloading conditions in Karush–Kuhn–Tucker form

$$\dot{\gamma} \geq 0\,, \quad f_s(\boldsymbol{t}_T) \leq 0\,, \quad \dot{\gamma}\, f_s(\boldsymbol{t}_T) = 0\,. \tag{11.20}$$

which determine the plastic parameter $\dot{\gamma}$.

11.2.3 Potential form for Solids in Contact

If W^δ is the strain energy function of the hyperelastic solid Ω^δ then the potential form follows as

$$U = \sum_{\delta=1}^{2} \left[\int\limits_{\Omega^\delta} W^\delta(\boldsymbol{C})\, \mathrm{d}V - \int\limits_{\Omega^\delta} \bar{\boldsymbol{f}}^\delta \cdot \boldsymbol{u}^\delta\, \mathrm{d}V \int\limits_{\Gamma_N^\delta} \bar{\boldsymbol{t}}^\delta \cdot \boldsymbol{u}^\delta\, \mathrm{d}A \right] \tag{11.21}$$

where the right Cauchy Green tensor $\boldsymbol{C}$ is given in (2.5), and examples of the strain energy function can be found in Sect. 2.2.2.1. The loading terms are related to the body force $\bar{\boldsymbol{f}}$ and the tractions $\bar{\boldsymbol{t}}$ at the Neumann boundary Γ_N.

Contact constraints, as described above, have now to be considered within the potential U. For this we assume for the moment that the contact surface is known.[4] Different methods can be used to enforce the contact constraints, among them are Lagrange multiplier, penalty and barrier approaches and many variants, see e.g. Wriggers (2006). They lead generally from a variational inequality to the minimization problem

$$U + U_{con} = 0 \tag{11.22}$$

where the part U_{con}, regarding the contact constraints will be formulated for the penalty and augmented Lagrangian method.

- **Penalty method.** A common approach to solve contact problems relies on the penalty method, see e.g. Kikuchi and Oden (1988), Introducing a penalty term to prevent indentations at contact in normal direction and tangential movement (stick) in tangential direction leads for a known contact surface Γ_c^1 to

$$U_{con}^{pen} = \frac{1}{2} \int\limits_{\Gamma_c} (\,\varepsilon_N g_N^2 + \varepsilon_T \parallel \boldsymbol{g}_T \|^2\,)\, \mathrm{d}\Gamma \tag{11.23}$$

with the penalty parameters ε_N and ε_T for normal and tangential directions, respectively. In case that stick occurs, the contact constraint can also be formulated with (11.13) as

$$U_{con}^{pen} = \frac{1}{2} \int\limits_{\Gamma_c} \varepsilon \parallel \boldsymbol{x}^2 - \bar{\boldsymbol{x}}^1 \|^2 \, \mathrm{d}\Gamma \tag{11.24}$$

where ε is now a penalty parameter for the constraint, for details see Wriggers and Haraldsson (2003) and Wriggers and Krstulovic-Opara (2004).
The penalty method is well suited to handle stick-slip contact based on the regularization discussed in the previous section. Once the threshold for sliding in (11.15) is exceeded the constitutive Eqs. (11.17) to (11.20) are valid. Due to their dissipative nature they have no potential. As in plasticity, a pseudo-potential can be formulated with (11.16) and (11.17)

$$U_{con}^{ps} = \frac{1}{2} \int\limits_{\Gamma_c} (\,\varepsilon_N g_N^2 + c_T \parallel \boldsymbol{g}_T^e \|^2)\, \mathrm{d}\Gamma\,. \tag{11.25}$$

where now in the first variation the tangential slip $\boldsymbol{g}_T^s$ has to be held constant.

[4] When solving variational inequalities it is common to introduce active sets that describe the contact surface. These active sets are defined within the solution algorithms.

- **Augmented Lagrangian method.** In contact mechanics, the augmented Lagrangian multiplier approach can have two meanings. The first one is related to a combination of Lagrange multiplier and penalty methods where the Lagrange multipliers are no additional unknowns, but determined within an Uzawa algorithm, see e.g. Simo and Laursen (1992), Laursen and Simo (1993) and Wriggers (2006). The downside of this simple method is the linear convergence of the algorithm which does not lead to an efficient numerical scheme.
  The second option of the augmented Lagrangian method was proposed in Alart and Curnier (1991), see also Pietrzak and Curnier (1999) and Lengiewicz et al. (2011), it is also known as primal-dual active set strategy. This methodology is closely related to a semismooth Newton method, see e.g. Christensen et al. (1998) and Popp et al. (2009). The following potential form can be constructed for the augmented Lagrangian scheme

$$U_{con}^{AL} = \int\limits_{\Gamma_c} \left[\mathcal{L}_N(g_N, \lambda_N) + \mathcal{L}_T(\boldsymbol{g}_T, \boldsymbol{\lambda}_T, \lambda_N) \right] \, \mathrm{d}\Gamma \tag{11.26}$$

  where the Lagrangian function $\mathcal{L}_N$ for the normal direction is defined as, see De Lorenzis et al. (2017),

$$\mathcal{L}_N(g_N, \lambda_N) = \begin{cases} \left(\lambda_N + \frac{\varepsilon}{2} g_N\right) g_N, & \hat{\lambda}_N \leq 0 \text{ (contact)} \\ -\frac{1}{2\varepsilon} \lambda_N^2, & \hat{\lambda}_N > 0 \text{ (separation)} \end{cases} \tag{11.27}$$

  and $\mathcal{L}_T$ related to the tangential direction is given for contact ($\hat{\lambda}_N \leq 0$) as

$$\mathcal{L}_T(\boldsymbol{g}_T, \boldsymbol{\lambda}_T, \lambda_N) = \begin{cases} \left(\boldsymbol{\lambda}_T + \frac{\varepsilon}{2} \boldsymbol{g}_T\right) \cdot \boldsymbol{g}_T, & \parallel \hat{\boldsymbol{\lambda}}_T \parallel \leq \hat{k} \text{ (stick)} \\ -\frac{1}{2\varepsilon}(\parallel \boldsymbol{\lambda}_T \parallel^2 - 2\hat{k} \parallel \hat{\boldsymbol{\lambda}}_T \parallel + \hat{k}^2), & \parallel \hat{\boldsymbol{\lambda}}_T \parallel > \hat{k} \text{ (slip)} \end{cases} \tag{11.28}$$

  and for separation ($\hat{\lambda}_N > 0$) as

$$\mathcal{L}_T(\boldsymbol{g}_T, \boldsymbol{\lambda}_T, \lambda_N) = -\frac{1}{2\varepsilon} \parallel \boldsymbol{\lambda}_T \parallel^2 \tag{11.29}$$

  with the penalty parameter $\varepsilon > 0$. The multiplier $\hat{\lambda}_N = \lambda_N + \varepsilon g_N$ in normal direction is used to distinguish between contact ($\hat{\lambda}_N \leq 0$) and separation ($\hat{\lambda}_N > 0$) states. The multiplier $\hat{\boldsymbol{\lambda}}_T = \boldsymbol{\lambda}_T + \varepsilon \boldsymbol{g}_T$ in tangent direction differentiates between stick ($\parallel \hat{\boldsymbol{\lambda}}_T \parallel \leq \hat{k}$) and slip ($\parallel \hat{\boldsymbol{\lambda}}_T \parallel > \hat{k}$) states. The quantity $\hat{k}$ denotes the radius of the Coulomb friction cone: $\hat{k} = max\left(0, -\mu\hat{\lambda}_N\right)$.
  The essential advantage of this formulation over the penalty method and classical Lagrange multiplier approaches is that U_c^{AL} is C^1-differentiable and thus provides a smooth transition from stick to slip and from contact to gap states, and vice versa.

The contact problem (11.22) can now be formulated as an unconstrained saddle-point problem.

11.3 Contact Discretization Based on Node Insertion

The virtual element method includes elements with arbitrary number of nodes. Thus it allows to add new nodes to existing discretizations and elements which can be employed to enforcement contact constraints at nodal level. Hence the simplest possible contact formulation, the node-to-node contact can be established.

The advantage over other methods is that the changes made to the element calculation are minimal. Adding a node does not change the element ansatz or integration procedure of the virtual element method, it just adds an additional term in the sum of the projection, see e.g. (3.43) for a linear ansatz. For higher order elements it only amounts to adding one step to the integration loop, see e.g. (3.69). Only the vector of the number of unknowns and with this the residual vector and the stiffness matrix have to be extended by the additional degrees of freedom. This procedure affects elements with inserted nodes but leaves the surrounding elements unchanged.

Figure 11.3 depicts the general procedure for node insertion when virtual elements are employed to discretize a contact problem. Node insertion is based on the following algorithm:

1. Check nodalwise for contact using standard search algorithms.
2. For nodes at master or slave surfaces that come into contact a matching new node will be introduced at the element boundary. Arrows in Fig. 11.3 indicate

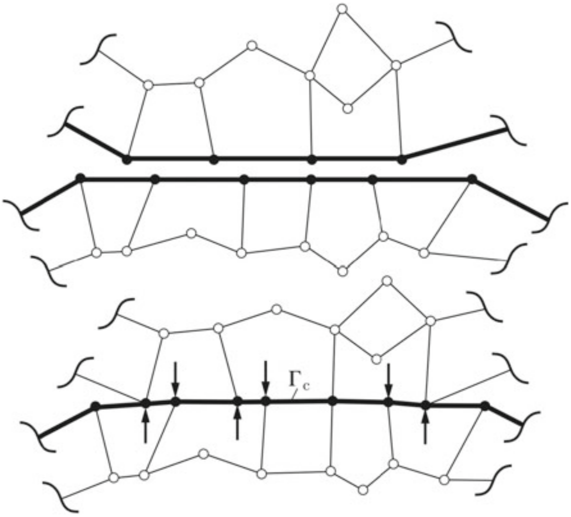


Fig. 11.3 Non-matching mesh and additional new contact nodes within a VEM discretization

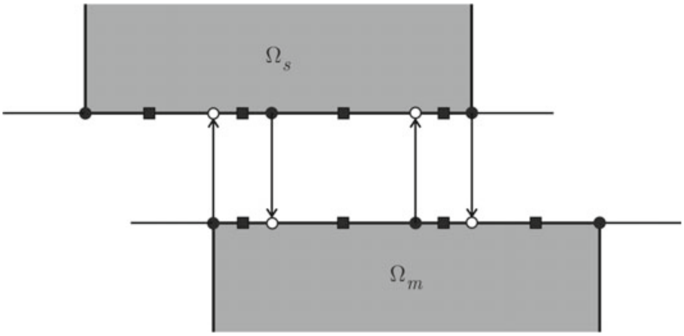


Fig. 11.4 Insertion for elements with quadratic interpolation at the edge

the insertion of new nodes at the master and slave surface where contact was established.

The inserted nodes provide the basis to formulate the contact constraints using penalty or augmented Lagrangian multiplier discretization. Ansatz functions for the gap as well as for Lagrange multipliers have to be formulated for the discretization. These interpolations can be based on the ansatz functions at the edges of the virtual elements, see e.g. (3.5) and (3.6) in Sect. 3.1.1.

Remark 11.1 In general the virtual element method leads to elements that have the same degrees of freedom as finite elements at the boundary. Thus VEM fits in the standard FEM framework and hence the virtual element method can easily be combined with standard finite elements. This fact can be explored to create a node-to-node contact approach for contact situations with non-matching meshes within a finite element discretization using a contact layer of virtual elements. ■

Remark 11.2 For virtual elements with quadratic ansatz functions the insertion algorithms has to be amended in such a way that it maintains a quadratic interpolation within the contact segments defined by the insertion procedure. Figure 11.4 depicts this special case. Starting from the coordinates of the contacting virtual elements "●" the projection yields the nodes denoted by "○". However a matching quadratic interpolation cannot be constructed from these points. Hence it is necessary to insert intermediate points "■". In this way each segment has a matching quadratic interpolation. Note that this procedure has also to be used at the end of contact zones as demonstrated in the left upper slave element Ω_s. ■

11.4 Two-Dimensional Treatment of Contact Using VEM

The above described insertion of nodes at the contact interface can be performed in an efficient way in case of two-dimensional problems. This is described in the next paragraphs in detail. Due to the fact that contact is only formulated at the boundary the ansatz $\mathbf{u}_h$ in (3.4) can be used directly in this section.

11.4.1 Inserted Node and Gap in the Two-Dimensional Case

A new node can be obtained within the node insertion algorithm by a closest point projection. Virtual elements have per definition straight edges thus a closed form solution can be derived.[5] Based on the notation introduced in Fig. 3.2 we define a projection, see Fig. 11.5, that denotes the closest distance g_n^s of a slave node s to the master edge l_e^m. It is related to the positions of the slave body Ω_n^2 at time t_n and for Ω_{n+1}^2 at t_{n+1} while the master body does not change its position. This projection defines the location of nodes P_n and P_{n+1} to be inserted and the related gaps g_n^s and g_{n+1}^S, respectively. The orthogonal projection is computed from (11.2) which yields in case of a straight edge for both positions[6]

$$\mathbf{a}_e^m \cdot [\mathbf{x}^s - \hat{\boldsymbol{x}}_e^m(\bar{\xi})] = 0 \tag{11.30}$$

with the tangent vector $\mathbf{a}_e^m = (\mathbf{x}_{k+1} - \mathbf{x}_k) \,/\, l_e^m$ and the length of the master edge $l_e^m = \|\mathbf{x}_{k+1} - \mathbf{x}_k\|$. Note that all vectors are in the current configuration and thus depend on the deformations. The position of the edge is given by $\mathbf{x}_{k+1}$ and $\mathbf{x}_k$. Since $\hat{\boldsymbol{x}}_e^m(\xi) = \mathbf{x}_k + (\mathbf{x}_{k+1} - \mathbf{x}_k)\,\xi$ with $\xi \in (0, 1)$ we obtain from (11.30)

$$\bar{\xi} = \frac{1}{l_e^m}\mathbf{a}_e^m \cdot (\mathbf{x}^s - \mathbf{x}_k) \tag{11.31}$$

which defines the position P of a new node (insertion) at the edge of the virtual element given by $\hat{\boldsymbol{x}}_e^m(\bar{\xi}) = \mathbf{x}_k + (\mathbf{x}_{k+1} - \mathbf{x}_k)\,\bar{\xi}$, see Fig. 11.5. The equations above hold for both time instances t_n and t_{n+1} and thus yield $\bar{\xi}_n$ and $\bar{\xi}_{n+1}$ when the associated coordinates are inserted.

[5] Lately virtual elements with curved edges were constructed, see e.g. Beirão da Veiga et al. (2019) and Wriggers et al. (2020). In this case the projection has to be based on an iterative procedure, see Aldakheel et al. (2020).

[6] In the following, all quantities that are related to nodal values are denoted by "bold roman" letters while quantities that are functions are provided as "bold italic" letters. Thus the slave node $\mathbf{x}^s$ is in "bold roman", while the coordinates describing the master surface $\hat{\boldsymbol{x}}_e^m(\xi)$ are in "italics". Furthermore the tangent $\mathbf{a}_e$ is defined by two nodes and thus "bold roman". The same is true for the normal $\mathbf{n}_e$.

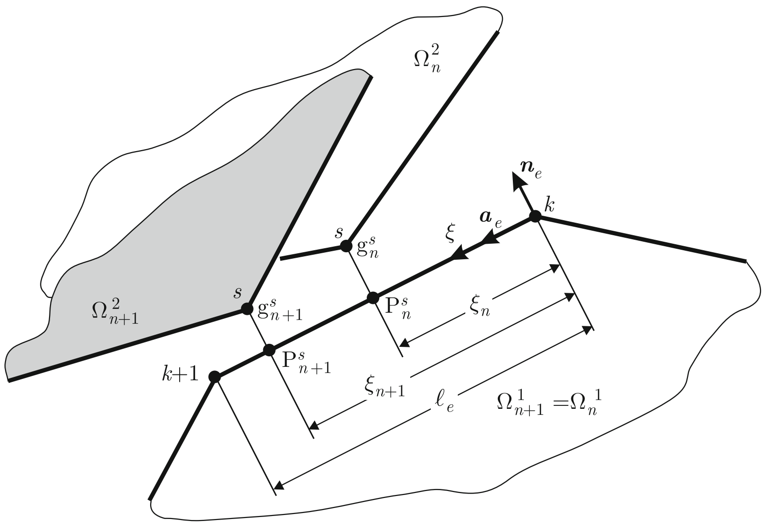


Fig. 11.5 Kinematics for projection slave to master

Once the projection is known the gap between the projected node $\hat{\mathbf{x}}_m(\bar{\xi})$ and the slave node g_s can be computed

$$g^s = [\mathbf{x}^s - \hat{\boldsymbol{x}}^m_e(\bar{\xi})] \cdot \mathbf{n}_e = [\mathbf{x}^s - \mathbf{x}_k] \cdot \mathbf{n}^m_e \tag{11.32}$$

and used to check for contact. The vector $\mathbf{n}^m_e$ is normal to the master surface. In the two-dimensional case it follows from $\mathbf{n}^m_e = \mathbf{a}^m_e \times \mathbf{e}_3$ where $\mathbf{e}_3$ is a unit vector perpendicular to the two-dimensional plane.

If the projection is extended to nodes of the master surface, being projected onto the slave surface, then the local coordinate - here we use η as surface coordinate on the slave surface—is given by

$$\bar{\eta} = \frac{1}{l_s} \mathbf{a}^s_e \cdot (\mathbf{x}_k - \mathbf{x}_s) \tag{11.33}$$

with the tangent vector $\mathbf{a}^s_e = (\mathbf{x}_{s+1} - \mathbf{x}_s) \,/\, l_s$ and the length of the slave segment $l^s_e = \|\mathbf{x}_{s+1} - \mathbf{x}_s\|$ at the interface of the virtual element of body Ω^2. The projection point leading to a new surface node of the virtual element is then defined by

$$\hat{\mathbf{x}}^s_e(\bar{\eta}) = (1 - \bar{\eta})\, \mathbf{x}_s + \bar{\eta}\, \mathbf{x}_{s+1} \,. \tag{11.34}$$

11.4.2 Discretization of the Contact Interface in 2d

Normal gap. Once the contact segments are established and the projections of slave nodes onto the master segments have created new nodes the gap function within a segment c (also called contact element) can be formulated using the interpolation for the displacements within the segment. Based on the definitions of nodal gaps $g_{N\,I\,n+1}$ at time t_{n+1}

$$g_{N\,I\,n+1} = \left[\mathbf{x}^s_{I\,n+1} - \hat{\mathbf{x}}^m_{I\,e}(\bar{\xi}_{n+1})\right] \cdot \mathbf{n}^m_{e\,n+1} \tag{11.35}$$

at each side of the contact segment c an interpolation can be introduced where n_a is the number of nodal pairs in the segment depending on the interpolation order. For linear virtual elements ($n_a = 2$) the ansatz function N_I is given by (3.5) and for a quadratic virtual element ($n_a = 3$) by (3.6). This ansatz leads to a continuous definition of the gap

$$g_{N\,c}(\eta) = \sum_{I=1}^{n_a} M_I(\eta)\, g_{N\,I\,n+1}\,. \tag{11.36}$$

where $\eta \in (0, 1)$ is a non-dimensional coordinate describing the gap function within the contact segment, see Fig. 11.6. Note that the current coordinates of the nodal pair associated with the contact segment c in (11.35) can be expressed by the coordinates in the initial configuration $\mathbf{X}$ and the displacements $\mathbf{u}$:

$$\mathbf{x}^s_{I\,n+1} = \mathbf{X}^s_I + \mathbf{u}^s_{I\,n+1} \quad \text{and } \hat{\mathbf{x}}^m_{I\,e}(\bar{\xi}_{n+1}) = \mathbf{X}^m_{I\,e} + \hat{\mathbf{u}}^m_{I\,e}(\bar{\xi}_{n+1})\,.$$

In case of small deformations the projection is based on the initial coordinates, leading to

$$\bar{\xi} = \frac{1}{l^m_e}\mathbf{A}^m_e \cdot (\mathbf{X}^s - \mathbf{X}_k) \tag{11.37}$$

which then is kept constant during the simulation as well as the tangent vector $\mathbf{A}^m_e$ and the normal vector $\mathbf{N}^m_e$. Hence the nodal normal gap in (11.35) can be simplified

$$g^{lin}_{I\,n+1} = \left[\mathbf{u}^s_{I\,n+1} - \hat{\mathbf{u}}^m_{I\,e}(\bar{\xi}_{n+1})\right] \cdot \mathbf{N}^m_e + g^0_I \tag{11.38}$$

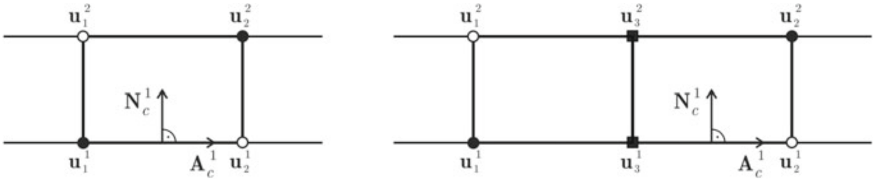


Fig. 11.6 Linear and quadratic interpolation at a contact segment

with the initial gap $g_I^0 = \left[\mathbf{X}^s_{I\,n+1} - \hat{\mathbf{X}}^m_{Ie}(\bar{\xi}_{n+1})\right] \cdot \mathbf{N}^m_e$. This leads to the interpolations

$$g_{N\,c}(\eta) = \sum_{I=1}^{n_a} M_I(\eta)\,(g^{lin}_{I\,n+1} + g^0_I)\,. \tag{11.39}$$

Tangential gap. Based on Eq. (11.10) for the relative tangential velocity it is possible to approximate the change of the convective coordinate within a time increment $\Delta t = t_{n+1} - t_n$ by the difference of the projection of a point of the slave surface onto the master surface: $\dot{\bar{\xi}} \approx (\bar{\xi}_{n+1} - \bar{\xi}_n)\,/\,\Delta t$, see Sect. 11.2.1. This yields the relative tangential displacement, see Fig. 11.5,

$$\Delta\boldsymbol{g}_{T\,n+1} = (\bar{\xi}_{n+1} - \bar{\xi}_n)\,\mathbf{a}^m_{e\,n+1} \quad \text{with } \Delta\boldsymbol{g}_{T\,n+1} = \boldsymbol{g}_{T\,n+1} - \boldsymbol{g}_{T\,n} \tag{11.40}$$

when an implicit Euler integration is used.[7] Note that the slip path can be computed with these approximations as

$$g_{T\,n+1} = g_{T\,n} + \int_{t_n}^{t_{n+1}} \|\dot{\bar{\xi}}\,\mathbf{a}^m_{e\,n+1}\|\mathrm{d}\tau = g_{T\,n} + (\bar{\xi}_{n+1} - \bar{\xi}_n)l^m_{e\,n+1} \tag{11.41}$$

where $\mathbf{a}^m_{e\,n+1}$ is defined as a unit vector in the current configuration as well as $\bar{\xi}$, see (11.31).

The projection leading to the insertion of a new node is performed at the end of the previous converged state t_n. The coordinates of the new node in the current configuration are given by $\hat{\mathbf{x}}^m_n = (1 - \bar{\xi}_n)\,\mathbf{x}^m_{k\,n} + \bar{\xi}_n\,\mathbf{x}^m_{k+1\,n}$. The relative tangential displacement is zero at the beginning of the new time increment ($\bar{\xi}_n = 0$) for the new nodal pair $(\mathbf{x}^s_I\,,\,\hat{\mathbf{x}}^m_I)$. This construction allows to compute the relative tangential gap at node I as

$$\bar{\xi}_{I\,n+1} = \left[\mathbf{x}^s_{I\,n+1} - \hat{\mathbf{x}}^m_{I\,n+1}\right] \cdot \mathbf{a}^m_{e\,n+1}\,\frac{1}{l^m_{e\,n+1}} \tag{11.42}$$

Note that the nodal pair inherits the history of previous tangential deformations related to stick and slip. Hence for such pair the tangential traction $t_{T\,I\,n}$ is known as well as the tangential motion $g_{T\,I\,n}$. Based on these nodal values the interpolation of the relative tangential motion

$$\Delta g_{T\,c}(\eta) = \sum_{I=1}^{n_a} M_I(\eta)\,\bar{\xi}_{I\,n+1} \tag{11.43}$$

is obtained for the contact segment c.

[7] Other integration rules—like the trapezoidal or Gauss rule—can be applied to integrate (11.10) which results in a more complex formulation. Since small time steps (load increments) have to be employed for many engineering problems the approximation in (11.40) is usually sufficient.

Equation (11.5) can be employed for small deformations to compute the change in the projection point $\bar{\xi}$

$$\dot{\bar{\xi}} = (\boldsymbol{v}^2 - \bar{\boldsymbol{v}}^1) \cdot \mathbf{A}_e^m \frac{1}{l_{e\,n+1}^m} \tag{11.44}$$

where $\mathbf{A}_e^m$ is the tangent vector to the edge of a virtual element in the initial configuration. By approximating the velocities

$$\boldsymbol{v}^2 = \frac{\boldsymbol{u}_{n+1}^2 - \boldsymbol{u}_n^2}{\Delta t} \quad \text{and } \bar{\boldsymbol{v}}^1 = \frac{\bar{\boldsymbol{u}}_{n+1}^1 - \bar{\boldsymbol{u}}_n^1}{\Delta t}$$

the slip increment within a time step Δt is given by

$$\Delta g_{T n+1} = (\bar{\xi}_{n+1} - \bar{\xi}_n) l_{e\,n+1}^m = (\boldsymbol{u}_{n+1}^2 - \bar{\boldsymbol{u}}_{n+1}^1) \cdot \mathbf{A}_e^m - \Delta u_{T n} \tag{11.45}$$

with the initial relative gap at time t_n: $\Delta u_{T\,n} = (\boldsymbol{u}_n^2 - \bar{\boldsymbol{u}}_n^1) \cdot \mathbf{A}_e^m$.

Since the projection points $\bar{\xi}$ define the boundary of the segment, see Fig. 11.3 in Sect. 11.3, the discretization is given by

$$\Delta g_{Tc}(\eta) = \sum_{I=1}^{n_a} M_I(\eta)\,(\Delta g_{TI} - \Delta u_{TIn}) \tag{11.46}$$

where the subscript $n+1$ in (11.45) has been omitted for the slip increment $\Delta g_{TI} = (\mathbf{u}_I^2 - \bar{\mathbf{u}}_I^1) \cdot \mathbf{A}_e^m$ and the initial gap increment Δu_{TIn} at the node I, see Fig. 11.6.

Discretization of the friction law in time. The constitutive modeling of frictional phenomena at the contact interface leads to a local differential equation in time, see Sect. 11.2.2 which will be solved for a given time interval $0 \le t \le T$. A time increment is denoted by $\Delta t = t_{n+1} - t_n$. The underlying stick-slip behaviour can be solved by a classical return mapping algorithm, see Wriggers (1987, 2006), which is well known from elasto-plasticity. The background for the algorithm is depicted in Fig. 11.7 and described in Algorithm 11.1. Here it is designed for a two-dimensional setting, details for the associated Euler backward integration within a return mapping scheme can be found in Wriggers (2006). The algorithm has to be performed at each integration point used to evaluate the contribution of the contact segment c.

The algorithm starts from a configuration in which a known tangential traction t_{Tn} is given.[8] This state is given by the relative slip at time t_n and the deformation related to the stick (elastic) part which yields the tangential traction $t_{Tn} = c_T\, g_{Tn}^e = c_T\,(g_{Tn} - g_{Tn}^s)$. According to Fig. 11.7 the trial tangential traction follows from $t_{Tn+1}^{tr} = t_{Tn} + c_T\,\Delta g_{Tn+1}$ where c_T is the constitutive constant (spring) that describes the magnitude of the relative stick motion. The associated Algorithm 11.1 is given below. Note that

[8] Here one could also start from a pure slip state related to the tangential slip g_{Tn}^s. This however is not possible when using the node insertion since the new node is obtained based on the local coordinate $\bar{\xi}$ that includes also the relative tangential motion due to stick.

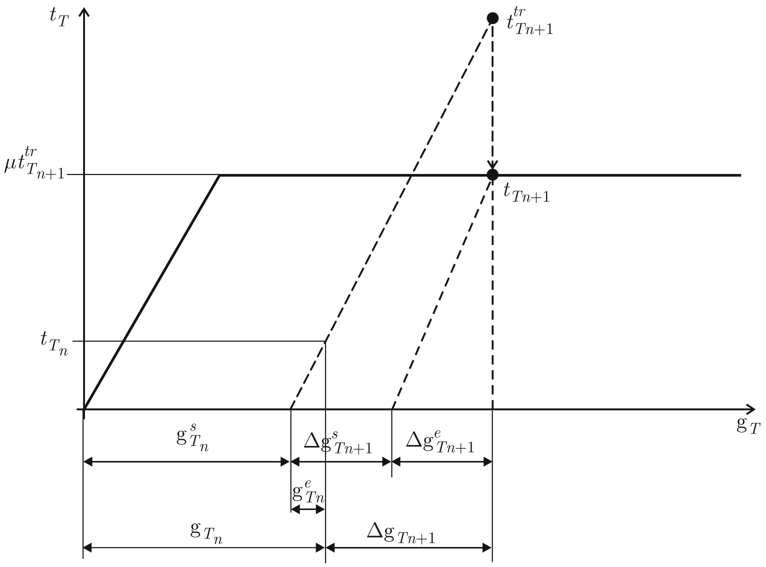


Fig. 11.7 Integration of the stick-slip motion in time

all tractions can be expressed in terms of the normal and relative tangential gap which underlines that the algorithm is displacement driven.

Algorithm 11.1: Algorithmic update of the frictional force and slip

```
Input: $\Delta g_{Tn+1}$ , $\Delta g_{Nn+1}$ , $g^s_{Tn}$
Output: $t_{Tn+1}$ , $\Delta g^s_{Tn+1}$
$t^{tr}_{Tn+1} == t_{Tn} + c_T \, \Delta g_{Tn+1}$
$t_{Nn+1} = \varepsilon_N \, g_{Nn+1}$
if Stick: $|t^{tr}_{Tn+1}| \leq \mu \, |t_{Nn+1}|$ then
    $g^s_{Tn+1} = g^s_{Tn}$
    $t_{Tn+1} = t^{tr}_{Tn+1}$
else if Slip:$|t^{tr}_{Tn+1}| > \mu \, |t_{Nn+1}|$ then
    $g^s_{Tn+1} = g^s_{Tn} + \frac{1}{c_T} \left[ t^{tr}_{Tn+1} - \mu |t_{Nn+1}| \, \text{sign}(t^{tr}_{Tn+1}) \right]$
              $= g_{Tn} + \Delta g_{Tn+1} - \frac{\mu}{c_T} |t_{Nn+1}| \, \text{sign}(t^{tr}_{Tn+1})$
    $t_{Tn+1} = \mu |t_{Nn+1}| \, \text{sign}(t^{tr}_{Tn+1})$
```

For the formulation of the different contact states the elastic relative displacement is needed. It follows from the equations in Algorithm 11.1

$$
\begin{aligned}
\text{stick:} \quad \Delta g^e_{Tn+1} &= (g_{Tn} - g^s_{Tn}) + \Delta g_{Tn+1} \, , \\
\text{slip:} \quad \Delta g^e_{Tn+1} &= \mu \frac{\varepsilon_N}{c_T} |g_{Nn+1}| \, \text{sign}(t^{tr}_{Tn+1})
\end{aligned}
\tag{11.47}
$$

where $\Delta g_{T n+1}$ is provided by (11.45) which is the only quantity that depends upon the current displacements. Interestingly, in case of slip, the elastic part of the relative tangential displacement increment depends only on the normal gap and thus will lead to a non-symmetric tangent matrix.

11.4.3 Penalty Formulation

The penalty method is based in the general case of frictional contact on the potential (11.25). In the two-dimensional case we split the potential into a part that yields the discretization of a contact interface with n_c segments for normal contact

$$U_N^{gap} = \frac{1}{2} \sum_{c=1}^{n_c} \int\limits_{(l_c)} \varepsilon_N \, g_{Nc}^2 \, \mathrm{d}\gamma \tag{11.48}$$

where the normal gap $g_{N\,c}$ is given by (11.39) for the linear and by (11.36) for the nonlinear case. The length of a segment c is l_c, see Fig. 11.6. This form describes a path independent potential that in the end will generate a symmetric contribution to the overall tangent matrix.

For the tangential contact one has to distinguish between stick and slip. For stick a path independent potential can be defined while the slip case needs the construction of a pseudo potential. The latter is based on the results provided in Algorithm 11.1 for the slip case. Both forms are combined below

$$U_T^{st-sl} = \begin{cases} \frac{1}{2} \sum_{c=1}^{n_c} \int\limits_{(l_c)} c_T \, \Delta g_{Tc}^2 \, \mathrm{d}\gamma, & |t_T| \leq \mu |t_N| \text{ (stick)} \\ \sum_{c=1}^{n_c} \int\limits_{(l_c)} \Delta g_{Tc} \, \mu \varepsilon_N |g_{Nc}| \, \mathrm{sign}(t_{Tn+1}^{tr}) \, \mathrm{d}\gamma, & |t_T| > \mu |t_N| \text{ (slip)} \end{cases} \tag{11.49}$$

where the relative tangential motion is given in (11.46) for small deformations and in (11.43) for finite deformations, respectively. In case of slip, the variation of U_T^{pen} has to be computed by holding the variables $g_{N\,c}$ and $\mathrm{sign}(t_{Tn+1}^{tr})$ constant.

The potential U_{Nc}^{pen} can now be evaluated for segment c as

$$U_{Nc}^{gap} = \frac{1}{2} \int\limits_{(l_c)} \varepsilon_N \, g_{N\,c}^2 \, \mathrm{d}\gamma = \frac{l_c \, \varepsilon_N}{2} \int\limits_0^1 \left[\sum_{I=1}^{n_a} M_I(\eta) \, g_{N\,I\,n+1} \right]^2 \mathrm{d}\eta \, . \tag{11.50}$$

This potential holds for the nonlinear case. For a linear problem g_{In+1} has to be replaced by $(g_{NIn+1}^{lin} + g_I^0)$. An analytical integration using the linear and quadratic ansatz (3.5) and (3.6) yields

$$U_N^{gap} = \begin{cases} \frac{l_c\,\varepsilon_N}{6}(g_1^2 + g_1 g_2 + g_2^2) & n_a = 2 \\ \frac{l_c\,\varepsilon_N}{30}(2g_1^2 - g_1 g_2 + 2g_2^2 + 2g_1 g_3 + 2g_2 g_3 + 8g_3^2) & n_a = 3 \end{cases} \tag{11.51}$$

where the subscript $n+1$ denoting the current time step has been omitted to simplify notation. The number of nodes of the segment are $n_a = 2$ for the linear and $n_a = 3$ for the quadratic ansatz and the nodal values g_I of the normal gap depend generally in a nonlinear way on the displacements, see (11.36).

For the stick-slip case the contact constraints in (11.49) have to be evaluated. This yields for the segment c in case of stick

$$U_{Tc}^{st} = \frac{1}{2}\int\limits_{(l_c)} c_T\,\Delta g_{T\,c}^2\,\mathrm{d}\gamma = \frac{l_c^2\,c_T}{2}\int\limits_0^1 \left[\sum_{I=1}^{n_a} M_I(\eta)\,\bar{\xi}_{In+1}\right]^2 \mathrm{d}\eta\,. \tag{11.52}$$

for nonlinear case. For a linear problem $\bar{\xi}_{In+1}$ has to be replaced by $(\Delta g_{TIn+1}^{lin} - \Delta u_{TIn})$. An analytical integration yields the same result as (11.51), only g_I has to be exchanged by $\bar{\xi}_{In+1}$ and ε_N by c_T.

In case of slip the pseudo potential

$$\begin{aligned} U_T^{sl} &= \int\limits_{(l_c)} \Delta g_{Tc}\,\mu\varepsilon_N |g_{Nc}|\,\mathrm{sign}(t_{Tn+1}^{tr})\,\mathrm{d}\gamma \\ &= \kappa\int\limits_0^1 \left[\sum_{I=1}^{n_a} M_I(\eta)\,\bar{\xi}_{In+1}\right]\left[\,|\sum_{I=1}^{n_a} M_I(\eta)\,g_{NIn+1}|\,\mathrm{sign}(t_{TIn+1}^{tr})\right]\mathrm{d}\eta \end{aligned} \tag{11.53}$$

needs to be solved which yields (omitting the subscript $n+1$ and the signum function)

$$U_T^{sl} = \begin{cases} \frac{\kappa}{6}[g_1(2\bar{\xi}_1 + \bar{\xi}_2) + g_2(\bar{\xi}_1 + 2\bar{\xi}_2)] \\ \frac{\kappa}{30}[g_1(4\bar{\xi}_1 - \bar{\xi}_2 + 2\bar{\xi}_3) + g_2(-\bar{\xi}_1 + 4\bar{\xi}_2 + 2\bar{\xi}_3) + 2g_3(\bar{\xi}_1 + \bar{\xi}_2 + 8\bar{\xi}_3)] \end{cases} \tag{11.54}$$

for $n_a = 2, 3$ with $\kappa = \mu\,\varepsilon_N\,l_c^2$.

Geometrical linear case. The residual and tangent matrix will be derived next for the case of linear virtual elements using the ansatz functions in (3.5). Let us define the vector of unknowns $\mathbf{u}_{Nc} = \langle u_{n1}^1\, u_{n2}^1\, u_{n1}^2\, u_{n2}^2\rangle^T$ in normal contact direction with $u_{nI}^A = \mathbf{u}_I^A \cdot \mathbf{N}_c^1$ and $\mathbf{u}_{Tc} = \langle u_{t1}^1\, u_{t2}^1\, u_{t1}^2\, u_{t2}^2\rangle^T$ in tangential direction at the contact interface with $u_{tI}^A = \mathbf{u}_{In+1}^A \cdot \mathbf{A}_c^1$. Furthermore, we introduce the vector $\mathbf{g}_{N0} = \langle\, g_1\, g_2\,\rangle^T$ for the initial normal gap and the vector $\Delta\mathbf{u}_{Tn} = \langle \Delta u_{T1n}\, \Delta u_{T2n}\rangle^T$ describing the initial gap increment. The analytically integration is given in (11.51) for the contact segment where $g_I = u_{nI}^2 - u_{nI}^1$ for $I = 1, n_a$ for the normal contact and $\Delta g_{TI}^{lin} = u_{tI}^2 - u_{tI}^1$ for $I = 1, n_a$ for the stick case. The result can now be differentiated with respect to the unknowns $\mathbf{u}_{Nc}$ and $\mathbf{u}_{Tc}$ which yields the residuals for the contact segment for the normal and tangential direction

$$\begin{aligned}
\mathbf{R}_{Nc} &= \frac{\partial U_c^{ps}}{\partial \mathbf{u}_{Nc}} = \frac{l_c\,\varepsilon_N}{6}\left[\mathbf{K}_c^L\,\mathbf{u}_{Nc} + \mathbf{G}^L\,\mathbf{g}_{N0}\right] \\
\mathbf{R}_{Tc} &= \frac{\partial U_c^{ps}}{\partial \mathbf{u}_{Tc}} = \frac{l_c\,c_T}{6}\left[\mathbf{K}_c^L\,\mathbf{u}_{Tc} - \mathbf{G}^L\,\Delta\mathbf{u}_{Tn}\right]
\end{aligned} \tag{11.55}$$

with the matrices

$$\mathbf{K}_c^L = \begin{bmatrix} 2 & 1 & -2 & -1 \\ 1 & 2 & -1 & -2 \\ -2 & -1 & 2 & 1 \\ -1 & -2 & 1 & 2 \end{bmatrix} \quad \text{and } \mathbf{G}^L = \begin{bmatrix} -2 & -1 \\ -1 & -2 \\ 2 & 1 \\ 1 & 2 \end{bmatrix}. \tag{11.56}$$

The derivation of the residuals with respect to the unknowns yields the tangent matrices

$$\mathbf{K}_{Nc}^L = \frac{\partial \mathbf{R}_c}{\partial \mathbf{u}_{Nc}} = \frac{l_c\,\varepsilon_N}{6}\mathbf{K}_c^L \quad \text{and } \mathbf{K}_{Tc}^L = \frac{\partial \mathbf{R}_c}{\partial \mathbf{u}_{Tc}} = \frac{l_c\,c_T}{6}\mathbf{K}_c^L\,. \tag{11.57}$$

The same derivations can be performed for quadratic virtual elements. In this case the ansatz functions (3.6) have to be employed to generated the matrix formulations for a contact segment c. This yields with the unknown vectors $\mathbf{u}_{Nc}^Q = \langle u_{n1}^1\, u_{n2}^1\, u_{n3}^1\, u_{n1}^2\, u_{n2}^2\, u_{n3}^2\rangle^T$ and $\mathbf{u}_{Tc}^Q = \langle u_{t1}^1\, u_{t2}^1\, u_{t3}^1\, u_{t1}^2\, u_{t2}^2\, u_{t3}^2\rangle^T$ and the initial normal and tangential gaps $\mathbf{g}_{N0}^Q = \langle\, g_1\, g_2\, g_3\,\rangle^T$ and $\Delta\mathbf{u}_{Tn}^Q = \langle \Delta u_{T1n}\, \Delta u_{T2n}\, \Delta u_{T3n}\rangle^T$

$$\begin{aligned}
\mathbf{R}_{Nc}^Q &= \frac{\partial U_c^{ps}}{\partial \mathbf{u}_{Nc}^Q} = \frac{l_c\,\varepsilon_N}{30}\left[\mathbf{K}_c^Q\,\mathbf{u}_{Nc}^Q + \mathbf{G}^Q\,\mathbf{g}_{N0}^Q\right] \\
\mathbf{R}_{Tc} &= \frac{\partial U_c^{ps}}{\partial \mathbf{u}_{Tc}^Q} = \frac{l_c\,c_T}{30}\left[\mathbf{K}_c^Q\,\mathbf{u}_{Tc}^Q - \mathbf{G}^Q\,\Delta\mathbf{u}_{Tn}^Q\right]
\end{aligned} \tag{11.58}$$

with the matrices

$$\mathbf{K}_c^Q = \begin{bmatrix} 4 & -1 & 2 & -4 & 1 & -2 \\ -1 & 4 & 2 & 1 & -4 & -2 \\ 2 & 2 & 16 & -2 & -2 & -16 \\ -4 & 1 & -2 & 4 & -1 & 2 \\ 1 & -4 & -2 & -1 & 4 & 2 \\ -2 & -2 & -16 & 2 & 2 & 16 \end{bmatrix} \quad \text{and } \mathbf{G}^Q = \begin{bmatrix} -4 & 1 & -2 \\ 1 & -4 & -2 \\ -2 & -2 & -16 \\ 4 & -1 & 2 \\ -1 & 4 & 2 \\ 2 & 2 & 16 \end{bmatrix}. \tag{11.59}$$

Again the derivation of the residuals with respect to the unknowns yields the tangent matrices

$$\mathbf{K}_{Nc}^Q = \frac{l_c\,\varepsilon_N}{30}\mathbf{K}_c^Q \quad \text{and } \mathbf{K}_{Tc}^Q = \frac{l_c\,c_T}{30}\mathbf{K}_c^Q\,. \tag{11.60}$$

In case of frictional contact with slip motions the results provided in (11.54) have to be employed to derive the matrix form. For the sliding case we define the vector of the nodal degrees of freedom by including the normal and tangential components

$$\mathbf{u}_{Sc} = \langle \mathbf{u}_{Nc}\, \mathbf{u}_{Tc} \rangle^T = \langle\, u_{n1}^1\, u_{n2}^1\, u_{n1}^2\, u_{n2}^2\, u_{t1}^1\, u_{t2}^1\, u_{t1}^2\, u_{t2}^2\, \rangle^T$$

for the linear ansatz. The residual $\mathbf{R}_S^L$ follows now from the integrated form (11.54) by differentiating with respect to vector $\mathbf{u}_{Sc}$ while keeping the normal components constant that govern the sliding motion

$$\mathbf{R}_{Sc}^L = \left.\frac{\partial U_c^{ps}}{\partial \mathbf{u}_{Sc}}\right|_{\Delta g_{NI}=const.} = \frac{\kappa}{6}\left(\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{K}_S^L & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_{Nc} \\ \mathbf{u}_{Tc} \end{Bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{G}^L \end{bmatrix} \mathbf{g}_{N0} \right) \tag{11.61}$$

where $\mathbf{G}^L$ and $\mathbf{K}_S^L = \mathbf{K}_c^L$ can be found in (11.56). The differentiation of the residual with respect to $\mathbf{u}_{Sc}$ leads to the tangent matrix that has the form

$$\mathbf{K}_{Sc}^L = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{K}_S^L & \mathbf{0} \end{bmatrix} . \tag{11.62}$$

Note that the differentiation of the signum function in (11.53) is zero since the latter is constant.

11.4.4 Augmented Lagrangian Multiplier Formulation

Within the augmented Lagrangian formulation, see (11.27) and (11.28) , the normal and tangential gaps as well as the Lagrangian multiplier have to be approximated along the contact interface. While the discretization of the normal gap g_N is already defined in (11.39) and the relative tangential motion can by found in (11.43) the Lagrangian multipliers λ_N and λ_T need to be interpolated within a contact segment c. Here we use the same functions (3.5) and (3.6) as for the displacement variables leading to the ansatz

$$\lambda_N = \sum_{A=1}^{n_a} L_A(\eta)\, \lambda_A \tag{11.63}$$

To enforce the contact constraints in normal direction by an augmented Lagrangian multiplier the following part, see (11.27), has to be added to the global functional describing the deformation of the body:

- **Normal contact.** In this case the amended Lagrange multiplier $\hat{\lambda}_N = \lambda_N + \varepsilon_N\, g_N$ fulfils $\hat{\lambda}_N \leq 0$ and the contact contribution is given for all n_c contact segments by

$$
\begin{aligned}
U_N^{AL} &= \sum_{c=1}^{n_c} \int\limits_{(l_c)} \left(\lambda_N + \frac{\varepsilon_N}{2} g_N \right) g_N \, \mathrm{d}\gamma \\
&= \sum_{c=1}^{n_c} l_c \int\limits_0^1 \left[\sum_{A=1}^{n_a} L_A(\eta)\, \lambda_{NA} + \frac{\varepsilon_N}{2} \sum_{I=1}^{n_a} M_I(\eta)\, g_{NI} \right] \sum_{I=1}^{n_a} M_I(\eta)\, g_{NI} \, \mathrm{d}\eta
\end{aligned} \tag{11.64}
$$

and for $\hat{\lambda}_N > 0$ by

$$
U_N^{AL} = \sum_{c=1}^{n_c} \int\limits_{(l_c)} -\frac{1}{2\varepsilon_N} \lambda_N^2 \, \mathrm{d}\gamma = -\sum_{c=1}^{n_c} \frac{l_c}{2\varepsilon_N} \int\limits_0^1 \left[\sum_{A=1}^{n_a} L_A(\eta)\, \lambda_{NA} \right]^2 \mathrm{d}\eta \tag{11.65}
$$

which depend on the nodal gaps g_{NI} (displacements) and the nodal Lagrange multiplier λ_{NA} (contact pressure). Note that the subscript $n+1$ has been omitted to simplify notation. For a linear problem g_{NI} has to be replaced by $(g_I^{lin} + g_I^0)$.

- **Stick-slip.** For tangential contact the potential form is given for contact, $\hat{\lambda}_N \leq 0$, and stick, $|\hat{\lambda}_T| \leq \hat{k}$, as

$$
\begin{aligned}
U_T^{AL} &= \sum_{c=1}^{n_c} \int\limits_{(l_c)} \left(\lambda_T + \frac{\varepsilon_T}{2} g_T \right) g_T \, \mathrm{d}\gamma \\
&= \sum_{c=1}^{n_c} l_c^2 \int\limits_0^1 \left[\sum_{A=1}^{n_a} L_A(\eta)\, \lambda_{TA} + \frac{\varepsilon_T l_c}{2} \sum_{I=1}^{n_a} M_I(\eta)\, \bar{\xi}_I \right] \sum_{I=1}^{n_a} M_I(\eta)\, \bar{\xi}_I \, \mathrm{d}\eta
\end{aligned} \tag{11.66}
$$

and for slip $|\hat{\lambda}_T| > \hat{k}$

$$
\begin{aligned}
U_T^{AL} &= \sum_{c=1}^{n_c} \int\limits_{(l_c)} -\frac{1}{2\varepsilon_T} (|\lambda_T|^2 - 2\hat{k}|\hat{\lambda}_T| + \hat{k}^2) \, \mathrm{d}\gamma \\
&= -\sum_{c=1}^{n_c} \frac{l_c}{2\varepsilon_T} \int\limits_0^1 \left[|\sum_{A=1}^{n_a} L_A(\eta)\, \lambda_{TI}|^2 - 2\hat{k}\, |\sum_{A=1}^{n_a} L_A(\eta)\, (\lambda_{TI} + \varepsilon_N l_c \bar{\xi}_I)| + \hat{k}^2 \right] \mathrm{d}\eta
\end{aligned} \tag{11.67}
$$

with $\hat{\lambda}_T = \lambda_T + \varepsilon_T g_T$. For separation, $\hat{\lambda}_N > 0$, the tangential contribution is

$$
U_T^{AL} = \sum_{c=1}^{n_c} \int\limits_{(l_c)} -\frac{1}{2\varepsilon_T} |\lambda_T|^2 \, \mathrm{d}\gamma = -\sum_{c=1}^{n_c} \frac{l_c}{2\varepsilon_T} \int\limits_0^1 \left[\sum_{A=1}^{n_a} L_A(\eta)\, \lambda_{TA} \right]^2 \mathrm{d}\eta \tag{11.68}
$$

with the penalty parameters $\varepsilon_N > 0\,, \varepsilon_T > 0$. The quantity $\hat{k}$ has to be formulated such that for no contact, see (11.68), sliding is impossible. This leads to the relation $\hat{k} = max\left(0, -\mu\hat{\lambda}_N\right)$. For a linear problem $\bar{\xi}_I l_c$ has to be replaced by $(\Delta g_{TI}^{lin} - \Delta u_{TIn})$, see (11.52).

The integration of the contact contributions within a segment c can again be performed analytically. It yields the same expressions as derived in (11.51) and (11.54). Thus a detailed description is omitted.

11.5 Three-Dimensional Treatment of Contact Using VEM

In the case of three-dimensional contact problems it is also possible to use node insertion and to establish node-to-node contact discretizations as discussed in Sect. 11.4. The related formulation is more complex as will be discussed below. In the following all displacement fields are given by the projection (3.157) $\mathbf{u}_\pi = \mathbf{N}_u^{(3,1)}(X, Y, Z)\, \mathbb{P}_u^{(3,1)}\, \mathbf{u}_v$ for three-dimensional virtual elements with linear ansatz.

11.5.1 Node Insertion for Contact of Three-Dimensional Solids

When using virtual elements for the spatial discretization, the surfaces are plane. A simple strategy for the computation of the contact projection is provided by a subdivision of a possible contact surface into non overlapping triangles, see also Sect. 3.2.3 where this approach was used for the integration over the surfaces of three-dimensional virtual elements. One of the triangles in the contact interface is depicted in Fig. 11.8. A linear triangular finite elements with the shape functions

$$N_1^\alpha = 1 - \xi^\alpha - \eta^\alpha \qquad N_2^\alpha = \xi^\alpha \qquad N_3^\alpha = \eta^\alpha\,, \tag{11.69}$$

is used where ξ^α and η^α denote the local coordinates in the parametric space for body Ω^α.

The location of the projection of a point $\boldsymbol{x}^2 = \boldsymbol{X}^2 + \boldsymbol{u}^2$ at the surface of the deformed body $\varphi(\Omega^2)$ onto the surface of deformed body $\varphi(\Omega^1)$ has now to be determined which then allows to compute the gap g_N in (11.3). This projection, known as closest point projection, is based on (11.2) which leads for the configuration shown in Fig. 11.8 to an additional set of equations to be solved for the unknown local coordinates ξ^1 and η^1

$$\left[\boldsymbol{x}^2 - \sum_{I=1}^{3} N_I^1\,(\bar{\xi}^1, \bar{\eta}^1)\,\mathbf{x}_I^1\right] \cdot \bar{\mathbf{a}}_\alpha = 0\,, \tag{11.70}$$

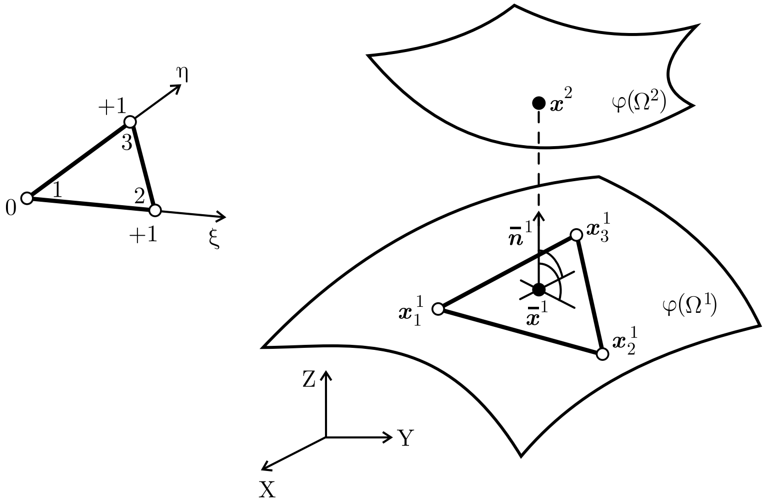


Fig. 11.8 Triangle in parametric space (left) and at contact interface in 3D Euclidian space (right)

with the nodal vectors $\mathbf{x}_I^1 = \mathbf{X}_I^1 + \mathbf{u}_I^1$ in the deformed configuration. The tangent vectors $\bar{\mathbf{a}}_\alpha$ are defined as

$$\bar{\mathbf{a}}_1 = \mathbf{x}_2^1 - \mathbf{x}_1^1 \quad \text{and} \quad \bar{\mathbf{a}}_2 = \mathbf{x}_3^1 - \mathbf{x}_1^1 \,. \tag{11.71}$$

Equation (11.70) yields in case of a planar triangular surface the linear equation system

$$\begin{Bmatrix} \bar{\xi}^1 \\ \bar{\eta}^1 \end{Bmatrix} = \begin{bmatrix} (\bar{\mathbf{a}}_1 \cdot \bar{\mathbf{a}}_1) & (\bar{\mathbf{a}}_1 \cdot \bar{\mathbf{a}}_2) \\ (\bar{\mathbf{a}}_2 \cdot \bar{\mathbf{a}}_1) & (\bar{\mathbf{a}}_2 \cdot \bar{\mathbf{a}}_2) \end{bmatrix}^{-1} \begin{Bmatrix} (\mathbf{x}^2 - \mathbf{x}_1^1) \cdot \bar{\mathbf{a}}_1 \\ (\mathbf{x}^2 - \mathbf{x}_1^1) \cdot \bar{\mathbf{a}}_2 \end{Bmatrix} \tag{11.72}$$

which can be solved directly for $\bar{\xi}^1$ and $\bar{\eta}^1$. Insertion of these points yields the gap g_N in (11.3): $g_N = (\mathbf{x}^2 - \bar{\mathbf{x}}^1) \cdot \bar{\mathbf{n}}^1$ which is the indicator for contact $g_N \leq 0$. The normal vector $\bar{\mathbf{n}}^1$ in Fig. 11.8 is given by

$$\bar{\mathbf{n}}^1 = \frac{\bar{\mathbf{a}}_1 \times \bar{\mathbf{a}}_2}{\|\bar{\mathbf{a}}_1 \times \bar{\mathbf{a}}_2\|} \tag{11.73}$$

The projection point $\bar{\mathbf{x}}^1$ in Fig. 11.8 is finally given by

$$\bar{\mathbf{x}}^1 = \sum_{I=1}^{3} N_I^1 \, (\bar{\xi}^1, \bar{\eta}^1) \, \mathbf{x}_I^1 \tag{11.74}$$

where $\bar{\xi}^1$ and $\bar{\eta}^1$ follow from (11.72).

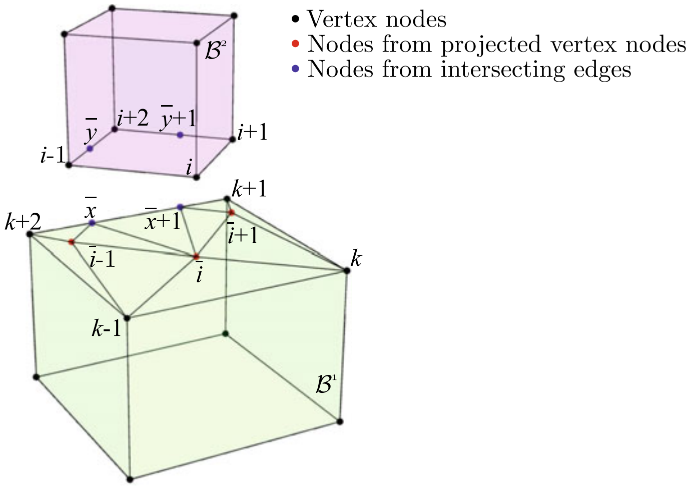


Fig. 11.9 Possible contact pairs between two virtual elements

Remark 11.3 In the case of small deformations, the computation of the projection is based on the initial configuration. Hence the tangent vectors are given by $\mathbf{A}_1 = \mathbf{X}_2^1 - \mathbf{X}_1^1$ and $\mathbf{A}_2 = \mathbf{X}_3^1 - \mathbf{X}_1^1$ and the normal is defined as $\bar{\mathbf{N}}^1 = \mathbf{A}_1 \times \mathbf{A}_2 \,/\, \|\mathbf{A}_1 \times \mathbf{A}_2\|$. The gap is then computed as $g_N = (\mathbf{x}^2 - \bar{\mathbf{x}}^1) \cdot \bar{\mathbf{N}}^1$. ■

Once node insertion is performed for all possible contact nodes at the intersection of two virtual elements Ω^1 and Ω^2 the gap can be evaluated for each matching nodal pair at the contact interface, see Fig. 11.9. The set of contact pairs in one interface includes

- nodes stemming from existing vertex nodes (black and red color in Fig. 11.9) and
- nodes related to edge-to-edge intersections of two bodies (blue color in Fig. 11.9).

For the node-to-face projection (like $\bar{i}$ to i in Fig. 11.9), one can compute the projection with (11.72).

For an edge-to-edge intersection the projection of e.g. $\bar{y}$ onto $\bar{x}$, see Fig. 11.9, is computed via closest point projection of the two lines $\boldsymbol{x}^1(\xi) = (1-\xi)\,\mathbf{x}^1_{k+2} + \xi\,\mathbf{x}^1_{k+1}$ and $\boldsymbol{x}^2(\eta) = (1-\eta)\,\mathbf{x}^2_{i+2} + \eta\,\mathbf{x}^2_{i-1}$ that define the edges under the condition that closest point has to be normal to the master surface. The computation of the minimum distance $d = \min \|\boldsymbol{x}^2(\eta) - \boldsymbol{x}^1(\xi) - \zeta\,\bar{\mathbf{n}}^1\|$ between the edge $\boldsymbol{x}^2(\eta)$ and the plane $\boldsymbol{x}^1(\xi) - \zeta\,\bar{\mathbf{n}}^1$ leads for straight edges with $\mathbf{t}^1 = \mathbf{x}^1_{k+1} - \mathbf{x}^1_{k+2}$ and $\mathbf{t}^2 = \mathbf{x}^2_{i-1} - \mathbf{x}^2_{i+1}$ to the orthogonality conditions

$$\begin{aligned} -\left[\boldsymbol{x}^2(\eta) - \boldsymbol{x}^1(\xi)\right] \cdot \mathbf{t}^1 &= 0 \\ \left[\boldsymbol{x}^2(\eta) - \boldsymbol{x}^1(\xi)\right] \cdot \mathbf{t}^2 &= 0 \\ -\left[\boldsymbol{x}^2(\eta) - \boldsymbol{x}^1(\xi)\right] \cdot \bar{\mathbf{n}}^1 &= 0\,. \end{aligned} \tag{11.75}$$

which yield an equation system for the local coordinates $(\bar{\xi}\,, \bar{\eta}\,, \bar{\zeta})$ that define the intersection point

$$\begin{bmatrix} (\mathbf{t}^1 \cdot \mathbf{t}^1) & -(\mathbf{t}^1 \cdot \mathbf{t}^2) & 0 \\ -(\mathbf{t}^1 \cdot \mathbf{t}^2) & (\mathbf{t}^2 \cdot \mathbf{t}^2) & -(\bar{\mathbf{n}}^1 \cdot \mathbf{t}^2) \\ 0 & -(\bar{\mathbf{n}}^1 \cdot \mathbf{t}^2) & (\bar{\mathbf{n}}^1 \cdot \bar{\mathbf{n}}^1) \end{bmatrix} \begin{Bmatrix} \bar{\xi} \\ \bar{\eta} \\ \bar{\zeta} \end{Bmatrix} = \begin{Bmatrix} (\mathbf{x}^2_{i+1} - \mathbf{x}^1_{k+2}) \cdot \mathbf{t}^1 \\ -(\mathbf{x}^2_{i+1} - \mathbf{x}^1_{k+2}) \cdot \mathbf{t}^2 \\ (\mathbf{x}^2_{i+1} - \mathbf{x}^1_{k+2}) \cdot \bar{\mathbf{n}}^1 \end{Bmatrix} . \tag{11.76}$$

Now the minimum distance at this point is an indicator for contact. When the gap g_N fulfills $g_N = \left[\mathbf{x}^2(\bar{\eta}) - \mathbf{x}^1(\bar{\xi})\right] \cdot \bar{\mathbf{n}}^1 = \bar{\zeta} \leq 0$ contact is established. For the line interaction the projection points are defined for each of the intersecting lines of body Ω^1 and Ω^2. They are given by

$$\bar{\mathbf{x}}^1 = (1 - \bar{\xi})\, \mathbf{x}^1_{k+2} + \bar{\xi}\, \mathbf{x}^1_{k+1} \quad \text{and} \quad \bar{\mathbf{x}}^2 = (1 - \bar{\eta})\, \mathbf{x}^2_{i+2} + \bar{\eta}\, \mathbf{x}^2_{i-1} \tag{11.77}$$

where $\bar{\xi}$ and $\bar{\eta}$ follow from (11.76).

Note, in case of small deformations the deformed coordinates of the nodal points have to be replaced adequately by the coordinates related to the undeformed configuration.

11.5.2 Algorithmic Treatment of Node-to-Node Intersection

Generally a two step approach has to be used for the treatment of contact problems. The first step is related to a global search to find contacting bodies, elements and surfaces. The global search algorithm is based on a description of the solids and elements by specific geometrical objects that contain the solids or elements but have a simple geometrical description. Then sorting algorithms are applied to provide a list for the next step, the local search. Details related to global search can be found in e.g. Laursen (2002) and Wriggers (2006).

Here we restrict ourselves to the second step, the local contact search, which is related to the element level. The following algorithm describes the methodology which is used for the local contact search using virtual elements. The focus lies on node-to-node contact. The related algorithm is formulated for one specific edge and surface, but can be applied for all intersecting edges and projections on surfaces in the same way. The local search has to be performed at each simulation step. We denote Ω^1 as master body and its surface Γ^1 as master surface which defines the contact normals $\bar{\mathbf{n}}^1$, see (11.73).

Algorithm:

1. Find all faces $\mathcal{F}_i^1$, of the surface of body Ω^1, i.e. $\bigcup_{i=1}^{n_s} \mathcal{F}_i^1 = \Gamma^1$ where n_s is the number of faces being in possible contact.
2. For each face $\mathcal{F}_i^1$ find the closest subset of faces $\tilde{\mathcal{F}}_1^2$ from the other body Ω^2 in the deformed configuration which have an orientation towards $\mathcal{F}_i^1$ and are located in the proximity of $\mathcal{F}_i^1$ within a given gap tolerance.
3. Project face $\mathcal{F}_i^1$ to $\tilde{\mathcal{F}}_1^2$ via the normal $\bar{\mathbf{n}}^1$ to find a projection point $\bar{\mathbf{x}}^1$ under the assumption of flat faces. This point is either (a) an intersection of an edge of the subset, see (11.77), or (b) a node within the interior of face $\tilde{\mathcal{F}}_1^2$, see (11.74).
4. Since $\bar{\mathbf{x}}^1$ is a point in the current configuration, with its location described by the convective coordinates $(\bar{\xi}\ , \bar{\eta})$, it has to be transformed back to the initial configuration $\bar{\mathbf{X}}^1$ for node insertion;

    a. For the edge-to-edge intersection the position in the initial configuration $\bar{\mathbf{X}}^1$ is defined by (11.77) with the local coordinate of the edge. Only the current coordinates $\mathbf{x}$ have to be exchanged by the initial coordinates $\mathbf{X}$. This yields $\bar{\mathbf{X}}^1 = (1-\bar{\xi})\,\mathbf{X}_{k+2}^1 + \bar{\xi}\,\mathbf{X}_{k+1}^1$ in the element Ω^1 and to $\bar{\mathbf{X}}^2 = (1-\bar{\eta})\,\mathbf{X}_{i+2}^2 + \bar{\eta}\,\mathbf{X}_{i-1}^2$ in the intersecting element Ω^2.
    b. Node $\bar{\mathbf{X}}^1$ in the node-to-face projection is found by evaluating (11.74) using the coordinates in the initial configuration: $\bar{\mathbf{X}}^1 = \sum_{I=1}^{3} N_I\,(\bar{\xi}^1, \bar{\eta}^1)\,\mathbf{X}_I^1$. Insertion of the point $\mathbf{X}^2$ on Γ^2 is not required since this pairing node is already known.

5. A check is made for each projection point if a node already exists at the same coordinate (intersection on edge end, or projecting on triangle corner). If not $\bar{\mathbf{X}}^1$ and for intersecting edges also $\bar{\mathbf{X}}^2$ have to be inserted within all elements that contain the edge intersection or the triangle.
6. All node-to-node pairs are gathered containing indices of projection $\bar{\mathbf{X}}^1$ and its origin $\mathbf{X}^2$. The number of contact elements is modified to match the number of pairs, furthermore the related node indices are updated. Thus the size of the global tangent changes in every load/time step.[9]
7. In case of node to face projection, a face has to be triangulated based on new included nodes, see Fig. 11.10. Nodes stemming from edge-to-edge insertion are blue while nodes related to node-to-face insertion are colored red.

This algorithm was described for Ω^1 being defined as master body. The same algorithm can be performed for Ω^2 being the master body, just by exchanging the indices 1 and 2.

Once contact is checked using the algorithm above and all nodes are inserted in the updated mesh and the contact contributions have to be added to the potential. Here the augmented Lagrangian from (11.26) will be used. Since only normal contact is considered, the part $\mathcal{L}_N(g_N\ , \lambda_N)$ is introduced, as defined in (11.27). The contact contributions of a segment Γ_s can be written as a sum over all triangles A_i with nodes $\mathbf{X}_i$, $\mathbf{X}_{i+1}$ and $\mathbf{X}_{i+2}$ which are defined by the above algorithm in each intersection of

[9] Nodes which are not active in the current step are removed from the current mesh and thus are deactivated. As a consequence, they have no contribution to the local/global tangent.

two virtual elements $\Omega_k^1$and Ω_m^2, see Fig. 11.10, leading to a set of n_s segments. Each segment Γ_s consists of a n_{km} triangles A_i. The gap function is a linear function in these triangles A_i when a virtual element with linear ansatz functions is used. The gap function within a triangle A_i is defined as

$$g_N^i(\xi,\eta) = \sum_{I=1}^{3} N_I^1\,(\xi^1,\eta^1)\,g_I \tag{11.78}$$

with

$$\begin{aligned} g_I &= \left[\mathbf{x}_I^2 - \sum_{K=1}^{3} N_K^1(\bar{\xi}_I\,,\bar{\eta}_I)\,\mathbf{x}_K^1\right]\cdot\bar{\mathbf{n}}_I^1 \quad \text{or} \\ g_I &= \left[\mathbf{x}^2(\bar{\eta}_I) - \mathbf{x}^1(\bar{\xi}_I)\right]\cdot\bar{\mathbf{n}}_I^1 \end{aligned} \tag{11.79}$$

where the index K defines a corner node of the triangle A_i. The computation of the nodal gaps g_I depends on the type of intersection: $(11.79)_1$ for node-to-triangle and $(11.79)_2$ for edge-to-edge contact. These define the vertices of the contact triangle A_i in Fig. 11.10.

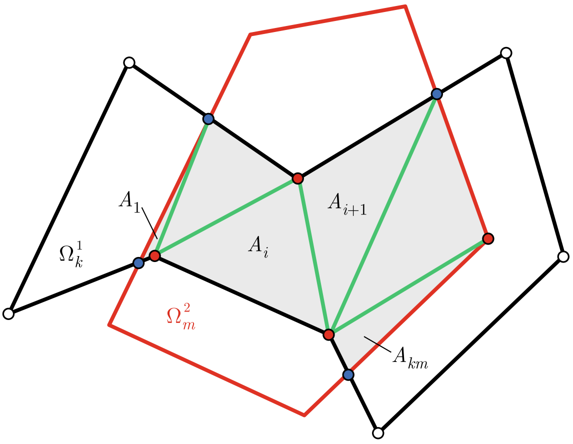
All contact contributions stem from the sum over all segments Γ_s defined by

$$U_{con}^{AL} = \int\limits_{\Gamma_c} \mathcal{L}_N(g_N\,,\lambda_N)\,\mathrm{d}\Gamma = \mathop{\mathbf{A}}_{s=1}^{n_s} \int\limits_{\Gamma_s} \mathcal{L}_N(g_N\,,\lambda_N)\,\mathrm{d}\Gamma \tag{11.80}$$

where the integral is a sum over all triangles of a segment Γ_s, see Fig. 11.10

$$\int\limits_{\Gamma_s} \mathcal{L}_N(g_N\,,\lambda_N)\,\mathrm{d}\Gamma = \mathop{\mathbf{A}}_{i=1}^{n_{km}} \int\limits_{A_i} \mathcal{L}_N(g_{N_i}\,,\lambda_N^i)\,\mathrm{d}A \tag{11.81}$$

Fig. 11.10 Triangularization of a contact segment based on the inserted points



Note that the Lagangian multiplier λ_N^i in a triangle A_i can also be approximated by a linear function in each triangle as $\lambda_N^i(\xi, \eta) = \sum_{I=1}^{3} L_A\,(\xi^1, \eta^1)\,\lambda_A$ with the nodal values λ_A.

In case of a penalty approach only the gap function enters the contact contributions. Based on (11.23) it follows

$$U_{con}^{pen} = \frac{1}{2}\int\limits_{\Gamma_c} \varepsilon_N g_N^2 \,\mathrm{d}\Gamma = \frac{1}{2}\,\overset{n_s}{\underset{s=1}{\mathbf{A}}}\int\limits_{\Gamma_s} \varepsilon_N g_N^2 \,\mathrm{d}\Gamma \tag{11.82}$$

with the penalty parameter ε_N. Again the integral on the right hand side is a sum over all triangles of a segment Γ_s, see Fig. 11.10,

$$\int\limits_{\Gamma_s} \varepsilon_N g_N^2 \,\mathrm{d}\Gamma = \overset{n_{km}}{\underset{i=1}{\mathbf{A}}}\int\limits_{A_i} \varepsilon_N\,[g_N^i]^2\,\mathrm{d}A \tag{11.83}$$

where the integral over the triangles A_i can be integrated with (11.78) using a Gauss point integration with n_g integration points

$$\int\limits_{A_i} \varepsilon_N\,[g_N^i]^2\,\mathrm{d}A = \varepsilon_N \sum_{g=1}^{n_g} [g_N^i(\xi_g\,, \eta_g)]^2\, w_g\, J_i \tag{11.84}$$

where w_g is the weighting parameter and $(\xi_g\,, \eta_g)$ are the integration points. Furthermore J_i is the determinant of the mapping between the reference and the initial configuration, see Fig. 11.9. The integration in (11.84) is based on a quadratic function since the gap function in (11.78) is linear. Furthermore the triangles A_i, defined in Fig. 11.10, have straight edges and a constant normal $\bar{\mathbf{n}}^1$ which allows an exact integration with three Gauss points, leading to

$$\int\limits_{A_i} \varepsilon_N\,[g_N^i]^2\,\mathrm{d}A = \frac{\varepsilon_N\, A_i}{6}\,\left[g_1(g_1 + g_2 + g_3) + g_2(g_2 + g_3) + g_3^2\right] \tag{11.85}$$

where g_1, g_2 and g_3 are the gap values at the nodes of the triangle A_i, see (11.79). This result can now be inserted in (11.82) and (11.83) together with the definitions (11.79). Now the contact potential of the penalty formulation is a function of the unknown displacements and thus the associated residuals and tangent matrices are directly computable.

The same integration can be performed for the augmented Lagrangian formulation. In that case, the Lagrangian multiplier is introduced as a linear function within each triangle A_i which again allows an exact integration of the contact contributions.

Another possibility to evaluate the contact contribution in (11.80) and (11.82) is based on the introduction of nodal forces at the vertices of the triangle being in contact. This leads for the penalty formulation to

$$U_{con}^{pen} = \frac{1}{2} \int\limits_{\Gamma_c} \varepsilon_N \, g_N^2 \, \mathrm{d}\Gamma = \frac{1}{2} \sum_{k=1}^{n_c} \varepsilon_N \, g_{N_k}^2 \tag{11.86}$$

where g_{N_k} is the gap and $\varepsilon_N \, g_{N_k}$ the nodal force associated with the contact node k. The total number of nodes that are in contact is given by n_c. Each node is treated in the same way in this formulation. Thus the area of the triangles, e.g. A_i, being in contact is not considered which might lead to numerical difficulties and erroneous local stress distributions. Remedial action can be taken by weighting the different contributions by the area of triangles $\bar{A}_k = \sum_{l=1}^{m} A_l \,/\, 3$ around each of the nodes k that are in contact. The area $\bar{A}_k$ can be computed as one third of the area of the m triangles that have the same vertex k.

11.6 Stabilization of VEM in Case of Contact

Generally all virtual elements derived so far can be used to discretize the solid part of a contact problem. Stabilization was discussed for virtual solid elements in Sect. 6.1.3 for the case of elasticity. Two possibilities exist which are stabilizations using bilinear forms, see Beirão da Veiga et al. (2013), and using am additional energy term, introduced in Wriggers et al. (2017). Only the additions to the interface discretization, derived above for contact, have to be added.

In case of the stabilization with a bilinear form, see e.g. Sect. 6.1.3, the formulation has to be changed for elements that share a contact interface. This is due to the fact that nodes can come very close to each other during the node insertion process and due to large sliding, To ensure the smoothness of the displacement field near these boundary points, the error $\boldsymbol{u}_h - \boldsymbol{u}_\pi$ is minimized at the boundary by introducing a special stabilization term

$$S^E = \gamma \sum_{i=1}^{n_V} \frac{|d(\mathbf{X}_{i+1}) - d(\mathbf{X}_i)|^2}{|\mathbf{X}_{i+1} - \mathbf{X}_i|} = \gamma \; S_*^E \, . \tag{11.87}$$

This special form of the stabilization S^E is a sum over the vertices of all nodes being in contact. It penalizes nodes that are close to each other stronger than other nodes. The penalty or stabilization parameter γ has to be selected according to the problem. The distance d can be defined at every vertex $\mathbf{X}_i$ as a local interpolation error computed from the difference between the nodal value and the interpolated value using (3.58) for the linear ansatz

$$d(\mathbf{X}_i) = [\mathbf{u}_i - \boldsymbol{u}_\pi(\mathbf{X}_i)] = \left(\mathbf{I} - \mathbf{H}_u^{(2,1)}(\mathbf{X}_i)\, \mathbb{P}_u^{(2,1)}\right) \mathbf{u}_v \, . \tag{11.88}$$

For the quadratic ansatz a similar relation holds, see (3.84) and (6.25).

The energy stabilization, described in Sect. 5.2.2, can be applied without any changes.

11.7 Numerical Examples

Numerical studies are presented in this section to demonstrate the robustness, convergence and accuracy of the contact formulation for virtual elements. This includes geometrical linear problems as well as contact of solids undergoing large deformations. All examples demonstrate that the node-insertion algorithms can be effectively applied for contact problems.

11.7.1 Behaviour of Different Stabilization Methods

The different stabilization methods, defined in Sect. 5.2, are assessed by using the test depicted in Fig. 11.11.

A regular mesh is connected to a Voronoi mesh in a way that creates irregular spaced nodes at the interface. The displacements of both meshes are fixed at the sides and the bottom. The system is loaded at the top as shown in Fig. 11.11. The

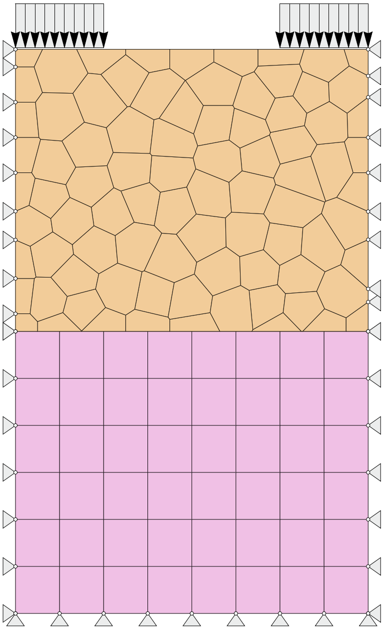


Fig. 11.11 Setup for testing different stabilizations with a quadrilateral and Voronoi mesh

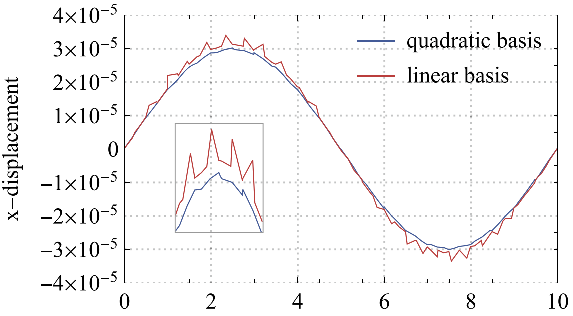


Fig. 11.12 Plot of x-displacement along the interface of both meshes for the DOFI stabilization, see Sect. 5.2.1

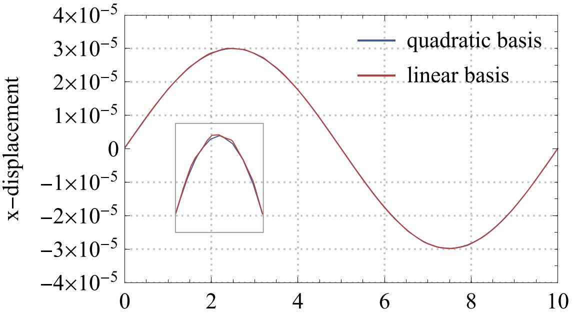


Fig. 11.13 Plot of x-displacement along the interface of both meshes for the edge stabilization in (11.87)

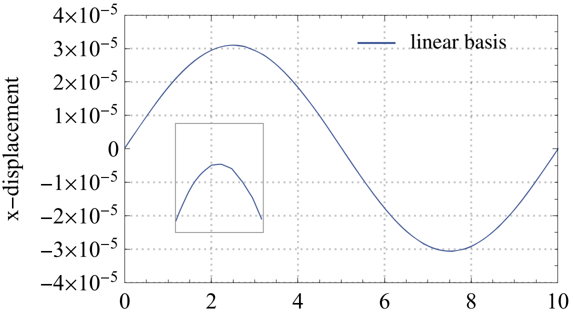


Fig. 11.14 Plot of x-displacement along the interface of both meshes for the energy stabilization, see Sect. 5.2.2

displacements at the interface of both meshes provide a measure for the quality of the stabilization of the virtual element formulation when using a non-matching mesh at the contact interface.

In Figs. 11.12, 11.13 and 11.14 the displacements in x-direction are plotted which show the influence of the used stabilizations. Starting with the bi-linear (DOFI) stabilization, see Sect. 5.2.1, it can well be observed that the displacements at the interface of both meshes oscillate locally, while the overall result is correct. This effect can be smoothed but not totally avoided by increasing the order of the polynomial basis. Hence the stabilization is not sufficient as the oscillations will lead to wrong gaps and thus incorrect contact stresses which yield erroneous contact or sliding states.

The edge stabilization in (11.87) exhibits a much smoother result for the surface displacements as depicted in Fig. 11.12. Because of this it is well suited for the virtual element formulation in contact computations. However the method still requires a parameter fitting to perform well, making its application to non-linear contact problems more difficult.

This is overcome by the energy stabilization, see Sect. 5.2.2. The results are shown in Fig. 11.14 which demonstrate a smooth surface displacement already for the linear basis.

11.7.2 Two-Dimensional Patch Test

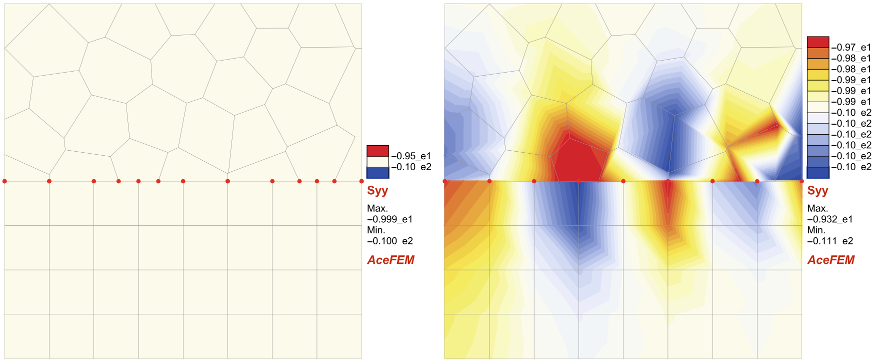
A contact patch test, see Fig. 11.15, is introduced in order to show that the virtual element method is able to produce constant stress states even for non-matching meshes at the contact interface. In detail, two blocks with the same dimensions of 2×1, but non-matching meshes are considered. The bottom block is discretized with a uniform mesh while for the block on the top a Voronoi mesh is chosen. Both blocks have the same material parameters $E = 1000$ and $\nu = 0.3$ plain strain is assumed and linear elastic material behaviour. The upper block is loaded with a pressure of $p_n = 2$. The bottom is fixed in vertical direction while the displacements at left and right side are constraint in horizontal direction.

By looking at the results in Fig. 11.15a, it can be seen that the VEM contact passes the patch test when using the node insertion methodology introduced in the previous sections. This is an advantage over the classical node-to-segment method,[10] either using FEM or VEM, used here as a comparison in Fig. 11.15b. For the non-uniform mesh the standard node-to-segment contact, see e.g. Wriggers (2006), does not guarantee accurate transfer of the pressure, showing small deviations.

11.7.3 Three-Dimensional Patch Test

The three dimensional contact patch test is illustrated in Fig. 11.16a. Both solid blocks Ω^1 and Ω^2 are fixed in x-direction at $x = 0$ and in y-direction at $y = 0$. The lower block Ω^2 is fixed in z-direction at $z = 0$. The solid Ω^1 is loaded at the upper face by the surface load $p_0 = 1 \cdot 10^7$. The Young's modulus is set to $E_1 = E_2 = 4 \cdot 10^8$ and the Poisson ratio is given by $\nu_1 = \nu_2 = 0.3$. The two blocks are discretized using

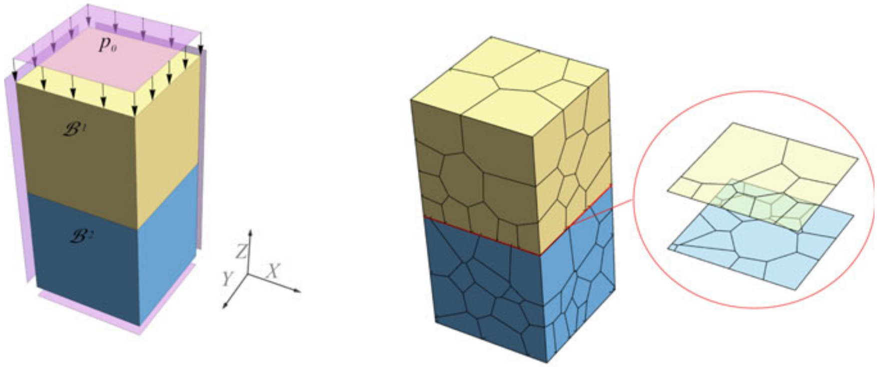
[10] For a finite element analysis there exist several formulations that also lead to a discretization which passes the patch test. Here we can mention the mortar method, see e.g. McDevitt and Laursen (2000), and special treatments of the node-to-segment schemes as discussed in Zavarise and De Lorenzis (2009).



(a) VEM: node-to-node contact

(b) VEM: node-to-segment

Fig. 11.15 Contact patch test—comparison of the stress distribution



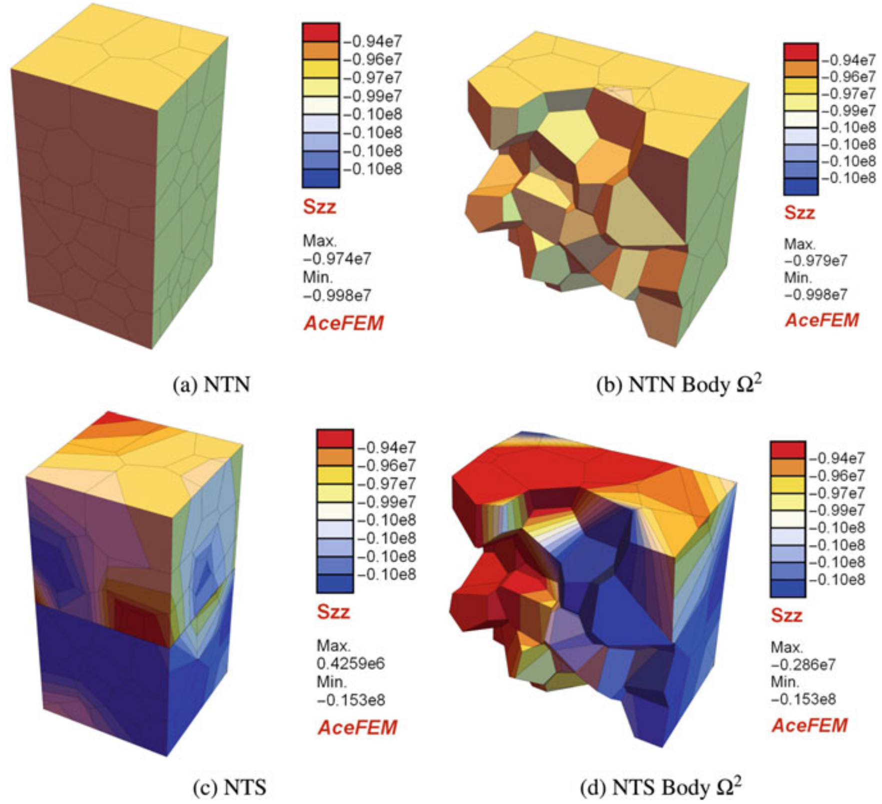
(a) System and loading

(b) Voronoi mesh

Fig. 11.16 Three-dimensional contact patch-test with mesh at the contact interface on the right

an irregular shaped Voronoi mesh which does not match at the contact interface, see the mesh of the contact interface for the contacting blocks on the right side of Fig. 11.16b.

Figure 11.17 depicts the stress σ_{zz} in the direction of the load. As in the two-dimensional case in the previous section, the node insertion scheme, leading to a node-to-node (NTN) discretization, reproduces a homogeneous stress state (Fig. 11.17a and b). A node-to-surface contact (NTS) enforcement results in incorrect contact forces and thus yields a non homogeneous stress state both bodies, as can be seen in Fig. 11.17c and d. This illustrates the advantage of the node insertion strategy which can be applied easily in the virtual element method.



(a) NTN

(b) NTN Body Ω^2

(c) NTS

(d) NTS Body Ω^2

Fig. 11.17 Patch-Test using node-to-node and node-to-surface contact

11.7.4 Hertzian Contact Problem, Two-Dimensional

This example was selected in order to test the accuracy of the solution for nonhomogeneous stress states. The Hertzian contact problem has an analytical solution for linear elastic material, thus the results obtained by the virtual element method can be compared to the analytical solution. Figure 11.18 shows the geometry and different discretizations. The disc Ω_1 of radius $R = 10$ is in contact with a stiff block Ω_2 of length $L = 20$ and height $H = 10$ which is fixed at the bottom. The disc is loaded by a force $F = 500$ that is applied as a pressure $\bar{t} = 25$ on one half of the disc. Young's moduli and Poisson's ratio are chosen as $E_1 = 7000$, $\nu_1 = 0.3$, $E_2 = 700000$ and $\nu_2 = 0.3$.

Following Johnson (1985), the analytical solution of the contact pressure p_n in the contact area is given by a function of the size of the contact area b and the effective material parameter E^*

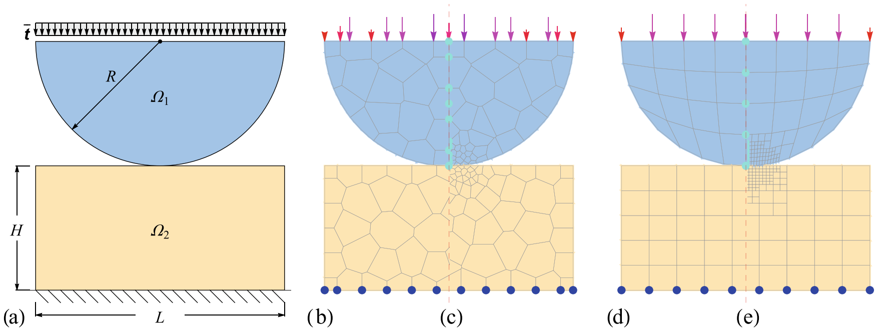
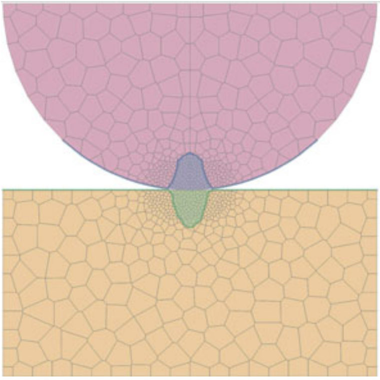


Fig. 11.18 Hertzian contact problem: **a** geometry and boundary conditions, **b** Voronoi cell mesh (VOU), **c** refined Voronoi cell mesh (VOU-ref), **d** structured 4 node quadrilateral discretization (Q1) and **e** refined structured mesh (Q1-ref)

Fig. 11.19 Distribution of the normal stress p_n in the contact area



$$p_n = \frac{4R\,\bar{t}}{\pi\,b^2}\sqrt{b^2 - x^2} \quad \text{and} \quad b = 2\sqrt{\frac{2R^2\,\bar{t}}{\pi\,E^*}} \quad \text{with} \quad \frac{1}{E^*} = \frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2} \tag{11.89}$$

The contact pressure p_n is used to compare with the numerical solution. For the given parameters its maximum is $p_{n,max} = 333.61$ and the contact area is $b = 0.954$.

The following possibilities to discretize the contacting surfaces will be discussed, see also Sect. 11.4.2,

- N-S: Standard node-to-segment contact, both edges are assumed to be linear,
- N-N: Node-to node contact, resulting from the node insertion used within the virtual element framework.

Figure 11.19 depicts the normal stress p_n at the contact interface for a discretization with a refined Voronoi mesh (VE-VOU-ref). This demonstrates that the VE-VOU formulation correctly reproduces the equivalence of the normal contact stress p_n in the disc and the block.

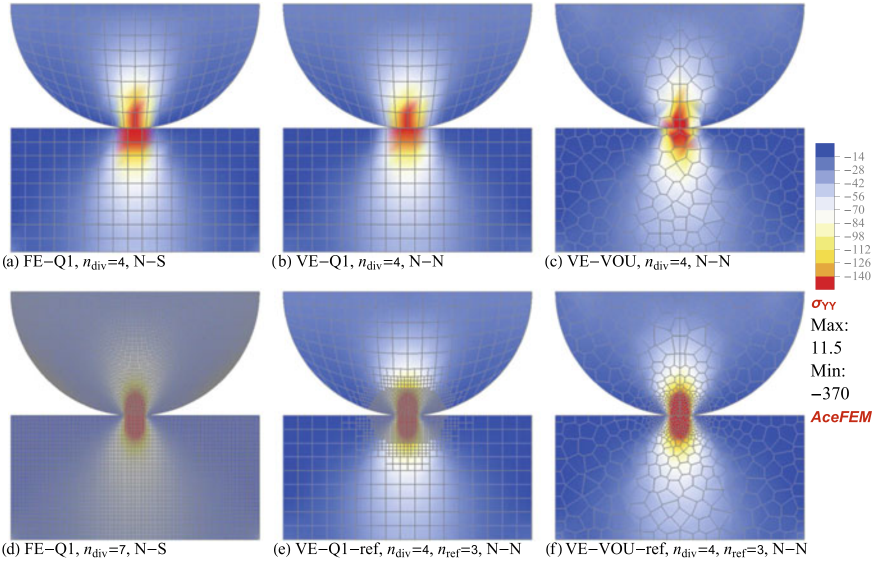


Fig. 11.20 Hertzian contact problem—stress distribution σ_{yy} for **a** standard FE-Q1, **b** standard VE-Q1, **c** coarse VE-VOU mesh, **d** fine FE-Q1 (overkill solution), **e** locally refined VE-Q1-ref and **f** locally refined VE-VOU-ref mesh

Figure 11.20 shows the distribution of the vertical stress σ_{yy} for different discretizations. These stresses were computed using the following mesh types:

- FE-Q1: finite element mesh with 4 noded quadrilateral elements,
- VE-Q1: virtual element mesh with 4 noded quadrilateral elements,
- VE-Q1-ref: locally refined VE-Q1 mesh,
- VE-VOU: uniform Voronoi cell mesh with arbitrary number of element nodes,
- VE-VOU-ref: locally refined VE-VOU mesh.

The structured meshes are refined locally within the predefined contact area $n_{ref} \in \{1, 2, 3, 4\}$ times. Whereas the locally-refined Voronoi mesh is based on the refinement until an element has an area smaller than $\bar{\Omega}_v \,/\, n_{ref}^2$ near the contact zone, where $\bar{\Omega}_v$ is the average element size of the non-refined Voronoi mesh. As a reference model, a very fine discretization is used which is based on the FE-Q1 mesh in Fig. 11.18d and shown in Fig. 11.20d.

In order to investigate the performance of the virtual element method, the results are compared with the analytical solution in (11.89). The relation between the contact forces $F_c \,/\, F_{c,ref} = \int_{\Gamma_c} \sigma_{yy} \, d\Gamma \,/\, \int_{\Gamma_c} p_n \, d\Gamma$ is introduced as an error measure. The convergence behaviour is depicted in Fig. 11.21 where various meshes and contact schemes are compared. As expected, linear convergence is achieved in the simulations when the meshes in Fig. 11.18b and c are employed. However, for the locally refined meshes, see Fig. 11.18e and f, we observe an increase of the convergence

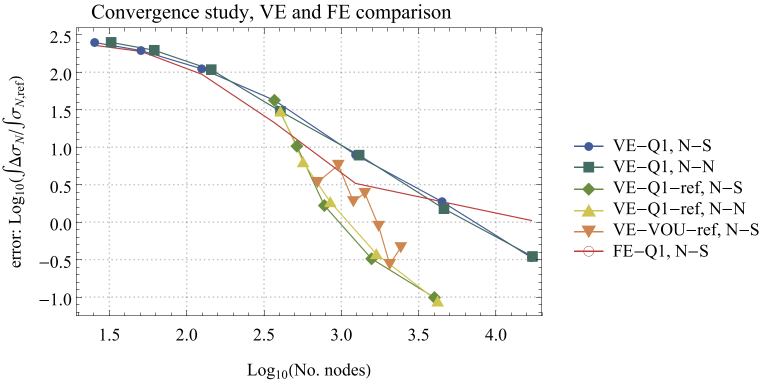


Fig. 11.21 Hertzian contact problem—convergence study of the total contact force

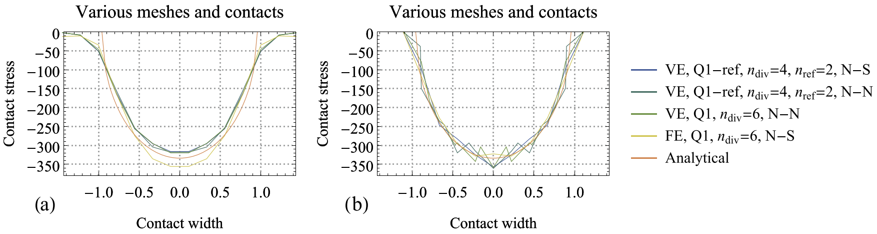


Fig. 11.22 Hertzian contact problem—contact stresses

rate, see e.g. results of VE-Q1-ref, N-S and N-N in Fig. 11.21. Moreover, we observe that all type of contact discretization schemes yield similar results as demonstrated in Fig. 11.20d–f.

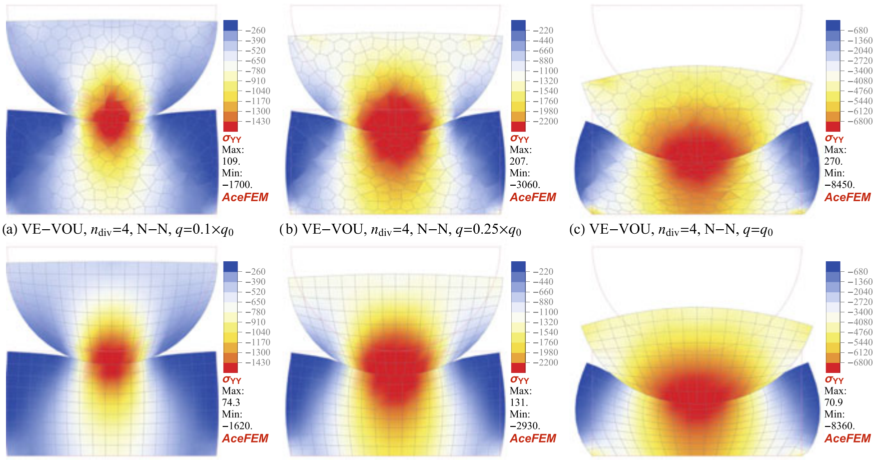
The distribution of the normal contact stresses at the contact zone is compared for various virtual and finite element discretizations with the analytical solution in (11.89). The stresses obtained from the actual stress at the surface as $p_n = (\boldsymbol{\sigma}\, \boldsymbol{n}) \cdot \boldsymbol{n}$ are shown in Fig. 11.22a. The stresses obtained from the contact element residuals at the nodes, normalized by the area related to each node as $p_n = (\boldsymbol{R}_c \,/\, A_c) \cdot \boldsymbol{n}$, are shown in Fig. 11.22b. The results demonstrate a good agreement with the analytical solutions for finer meshes. For finest discretization (ndiv = 6) the graphs coincide, except for the finite element solution which yields for the same mesh division slightly overestimated stresses. Furthermore it can be observed that the computation from the actual stresses at the surface, see Fig. 11.22a, yields a smoother stress distribution at the contact interface.

11.7.5 Hertz Contact for Large Deformations, Two-Dimensional

The Hertzian contact problem, sketched in Fig. 11.18, is computed allowing large deformations. In this example, the node adjustment algorithm is employed for the node-to-node treatment with virtual elements. To achieve large deformations, the Young's modulus of the upper disc Ω_1 is changed to $E_1 = 70000$ and the Young's modulus of the block Ω_2 is reduced to $E_2 = 7000$. The Poisson ratio is the same for both solids, $\nu_1 = \nu_2 = 0.3$. The top surface is loaded in 10 load steps with a distributed pressure of $p_n = 5000$. Frictionless contact is considered with using as regularization parameter $\varepsilon_N = 20000$.

The numerical simulation is based on virtual elements with linear ansatz using the node-to-node technique described in Sect. 11.3–11.4.2 together with a penalty approach. The two solids are discretized with a mesh based on regular Voronoi cells and a regular mesh. Both discretizations can be seen in Fig. 11.23a–c for the Voronoi (VOU) and Fig. 11.23d–f for the regular (Q1) meshes. Figure 11.23a and d depict the deformation for 10% of the total applied load $p_{a)} = 0.1 p_n$ while Fig. 11.23b and e show the deformation state at $p_{b)} = 0.25 p_n$. The final deformation under the total load is then provided in Fig. 11.23c and f. The contour plots demonstrate a correct transmission of the stresses σ_{yy}.

The load-displacement curve of the applied load $F = 2R\, p_n$ versus the mid-displacement u_y of the disc in the contact zone illustrates the nonlinear behaviour for this finite deformation contact problem. We observe that Voronoi as well as regular discretizations yield the same global response curve (Fig. 11.24).



(a) VE−VOU, n_{div}=4, N−N, q=0.1×q_0 (b) VE−VOU, n_{div}=4, N−N, q=0.25×q_0 (c) VE−VOU, n_{div}=4, N−N, q=q_0

(d) VE−Q1, n_{div}=4, N−N, q=0.1×q_0 (e) VE−Q1, n_{div}=4, N−N, q=0.25×q_0 (f) VE−Q1, n_{div}=4, N−N, q=q_0

Fig. 11.23 Large deformation Hertz contact—deformed configurations for a sequence of loading steps from 10% over 25% to 100% of the applied load p_n

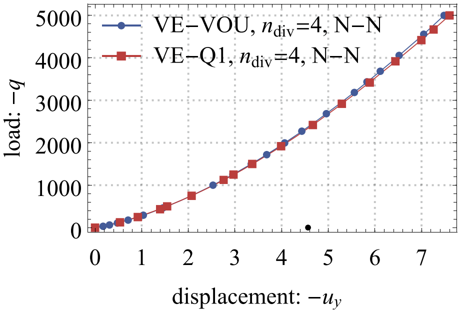


Fig. 11.24 Large deformation Hertz contact—load-displacement curve

11.7.6 Hertzian Contact, Three-Dimensional

The Hertzian contact of a half sphere and a block is analyzed. The analytical result, see e.g. Hertz (1882) and Johnson (1985), yields the normal contact pressure as a function of the radius r

$$p_n(r) = \frac{2E^*}{\pi R}\sqrt{a^2 - r^2}\,, \tag{11.90}$$

where R is the radius of the sphere, a the width of the contact area and E^* the effective modulus. The width of the contact area a and the effective Young's modulus E^* are given as

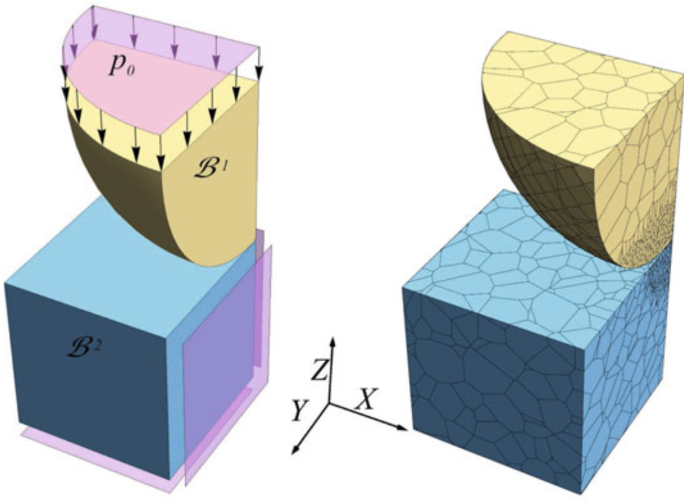
$$E^* = \left(\frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2}\right)^{-1} \quad \text{and} \quad a = \left(\frac{3q_0\pi R^3}{4E^*}\right)^{\frac{1}{3}} \tag{11.91}$$

where q_0 is the load at the top of the half sphere.

Due to its symmetry, the problem can be reduced to a quarter, see Fig. 11.25a. In that case, symmetry boundary conditions have to be introduced which constrain the deformation in normal direction at the symmetry planes. Furthermore, the lower face of the block fixed in z-direction.

A discretization is shown in Fig. 11.25b for the Voronoi type elements. These are denoted by VO while regular shaped hexahedral elements are labeled by H1. Both types of virtual elements are employed to discretize the three-dimensional Hertz problem.

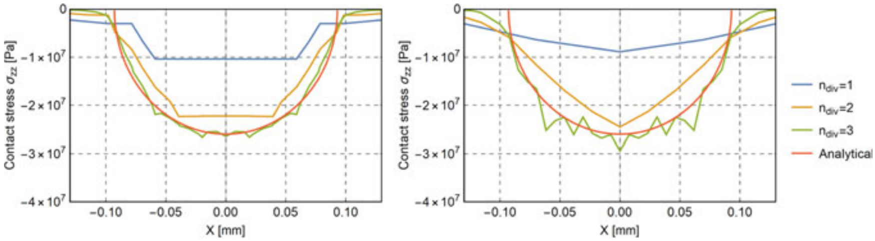
Figure 11.26 depicts the normal stress distribution along the surface. The contact stresses $p_n = \sigma_{zz}$ are obtained by projecting the actual stress $\boldsymbol{\sigma}$ onto the normal direction $\sigma_{zz} = (\boldsymbol{\sigma}\boldsymbol{n}) \cdot \boldsymbol{n}$ where $\boldsymbol{n}$ coincides with the z-direction: $\boldsymbol{n} = \boldsymbol{e}_z$. Figure 11.26 reports the normal stress only for the H1 mesh. It can be seen that the contact pressure converges to the analytical solution for a mesh refinement $n_{div} = 3$. The node-to-node contact (NTN) based on node insertion, see Fig. 11.26a, leads to smoother stress distributions than provided by the node-to-surface (NTS) formulation, see



(a) System and loading

(b) Voronoi mesh

Fig. 11.25 Three-dimensional Hertzian contact problem



(a) NTN VE-H1

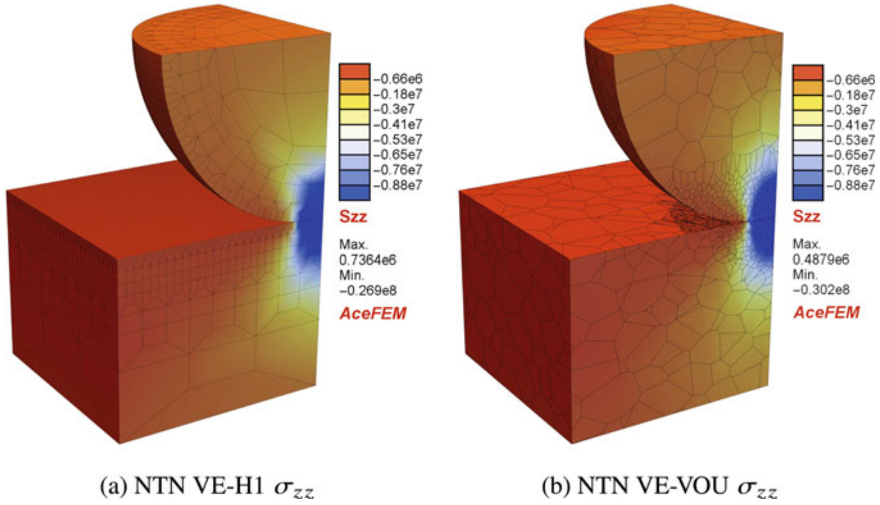
(b) NTS VE-H1

Fig. 11.26 Stress distributions $\sigma_{zz} = (\boldsymbol{\sigma n}) \cdot \boldsymbol{n}$ for node-to-node in (**a**) and node-to-surface in (**b**)

Fig. 11.26b. Oscillations of the stresses are stronger for node-to-surface contact and reduced for the node-to-node contact. More results details can be found in Cihan et al. (2022).

For Voronoi meshes, oscillations are reduced, but constant stresses in the segments are obtained. This is in line with the linear virtual element ansatz, which leads to constant stresses within each virtual element. Hence, the introduction of new nodes during the projection scheme for contact does not result in an improved stress. Clearly, a refinement of the mesh is needed to obtain more accurate results.

Figure 11.27 depict the stress contours of the vertical stress component σ_{zz} using virtual elements for two different meshes. Both meshes are pre-refined near the contact zone. The results demonstrate a good coincidence between the regular VE-H1, in Fig. 11.27a, and the VE-VO Voronoi mesh, see Fig. 11.27b.



(a) NTN VE-H1 σ_{zz}

(b) NTN VE-VOU σ_{zz}

Fig. 11.27 Normal stress distribution σ_{zz} for node-to-node contact

11.7.7 Contacting Beams

Contact between two beams is studied to demonstrate the efficiency of the node-to-node contact using the virtual element method. The two beams are clamped at the right and left side and come into contact in the middle.

The associated boundary value problem is shown in Fig. 11.28 along with the applied load. The specimen sizes are as follows: $H_1 = H_2 = 1$, $L_1 = L_2 = 20$ and $R = 2$. A uniform line load is applied on top of the free end of the upper beam Ω_1 with a magnitude of $\bar{t} = 50$, as shown in Fig. 11.28. The elastic material parameters are selected as $E_1 = E_2 = 70000$ and $\nu_1 = \nu_2 = 0.3$.

Different virtual element mesh types similar to the Hertzian contact problem are investigated to illustrate the performance of the virtual element method under bending deformations and contact. The meshes consist of uniform Voronoi cells

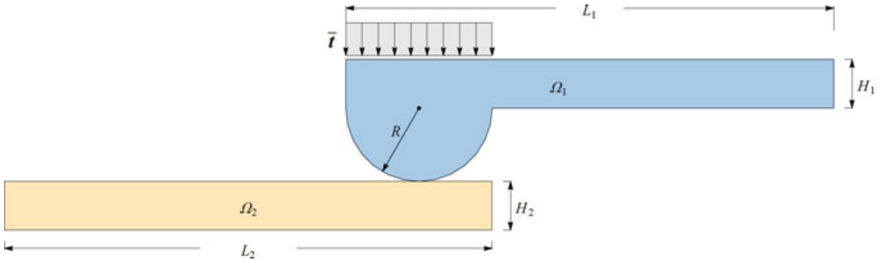


Fig. 11.28 Beams in contact: Geometry and boundary conditions

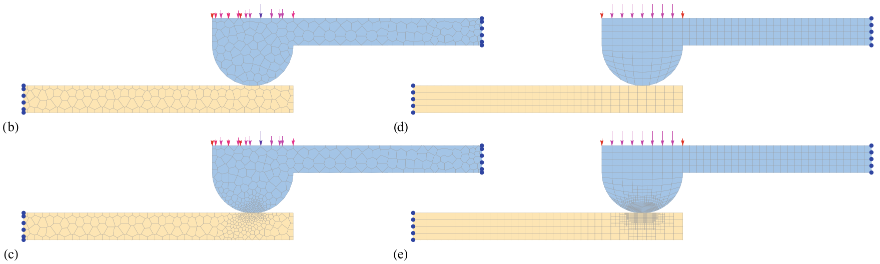


Fig. 11.29 Beams in contact: **b** Voronoi mesh (VOU); **c** locally refined Voronoi mesh (VOU-ref); **d** structured (Q1) mesh; and **e** locally refined structured mesh (Q1-ref)

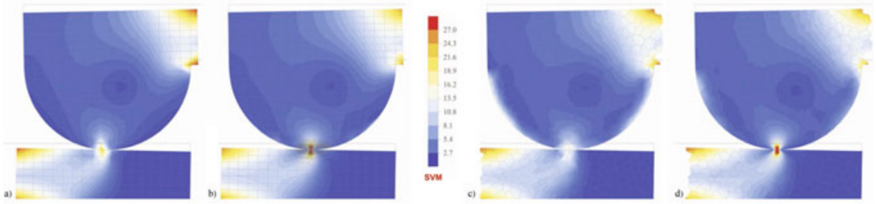


Fig. 11.30 Distribution of von Mises stresses σ_{VM}: **a** uniform FE-Q1; **b** local refined VE-Q1-ref; **c** uniform VE-VOU and **d** local refined VE-VOU-ref

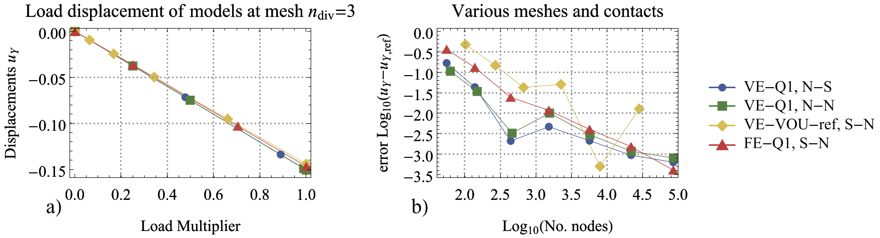


Fig. 11.31 Beams in contact: **a** load-displacement curve and **b** convergences study

and structured Q1 elements. All meshes are depicted in Fig. 11.29b–e. The virtual element solutions are compared with a finite element analysis which is based on the structured quadrilateral Q1 mesh shown in Fig. 11.29d.

Figure 11.30 depicts the distribution of the Von Mises stress σ_{VM} for the four different meshes in Fig. 11.29. We note, that the coarse discretizations in Fig. 11.30a and c lead to a good approximation of the overall deformation state of the two contacting beams. For a good resolution of the local stress field, a mesh refinement in the contact area is necessary, see Fig. 11.30b and d which provide accurate von Mises stresses in the contact zone.

The load displacement curve is shown in Fig. 11.31a for different discretizations. Note, that even coarse discretizations recover the overall deformation behaviour correctly. Figure 11.31b illustrates a convergence study for the displacement at the top left corner point of beam Ω_1 versus mesh refinements. Hereby, all schemes converge to the reference solution $u_{\text{y,ref}}$ obtained with a very fine mesh (overkill solution). Note however, that the node-to-segment discretization of the refined Voronoi mesh does not converge smoothly while the VE-Q1 discretizations depict a high coarse mesh accuracy in the beginning (up to $\log_{10} = 2.5$ elements) which then levels out to the convergence behaviour of the finite element discretization.

No major difference between contact techniques (node-to-node and node-to-segment) was observed in this example,

11.7.8 Wall Mounting of a Bolt

A wall mounting holds a bolt, see Fig. 11.32, and is subjected to loading via the bolt up to fracture. The bolt is in contact with the mount, thus a combined contact and fracture problem has to be solved. The fracture process is modeled using the phase field approach discussed in Sect. 10.3.

A bolt of radius $r_b = 9.5$ cm is attached to a mount that is fixed on its left side to a wall, see Fig. 11.32. On this side all nodes are constraint in vertical direction and the middle node is fixed in normal direction.

The mount has an external radius of $r_a = 20$ cm and an inner radius of $r_i = 10$ cm. The distance of the middle of the bolt from the wall is $x_b = 20$ cm. The material for the bolt is assumed to be elastic with a Young's modulus of $E_b = 70000$ kN/cm^2 and a Poisson ratio of $\nu_b = 0.3$. The mount has a Young's modulus of $E_m = 7000$ kN/cm^2 and a Poisson ratio of $\nu_m = 0.3$. The material of the mount is brittle thus the phase field approach for brittle fracture can be applied. The associated critical energy release rate, see (10.61), is $g_c^m = 0.1$ kN/cm and the viscosity of the crack propagation is $\eta^m = 10^{-5}$ kNs/cm^2 for the mount. The fracture length scale l in (10.56) is chosen as $l = 0.015$ cm.

The bolt just touches the inner radius of the mount in the beginning of the simulation. The focus of this simulations is the failure mechanism of the mount due to

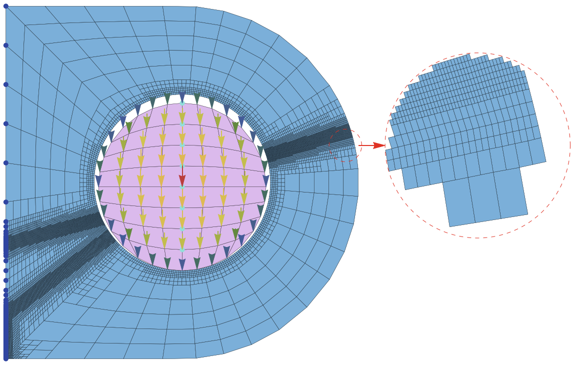


Fig. 11.32 Wall mounting of a bolt—loading, boundary conditions, virtual element mesh and a zoom-in of the mesh

loading. The contact analysis is necessary for a correct load transmission from the bolt to the mount. For the contact analysis a node-to-segment contact discretization is utilized based on the penalty formulation with a penalty parameter of $\epsilon_n = 20000$ kN/cm. Since no tangent movement is expected friction can be neglected at the contact interface. The load is applied as body force in the bolt. A structured mesh is employed, see Fig. 11.32, for the discretization with virtual elements. The energy stabilization is used within the virtual element analysis with the stabilization parameter $\beta = 0.4$.

Mesh refinement in the expected fracture zone is applied in an adaptive manner. First a fine mesh around the bolt is employed to find the location where the crack starts. After crack initiation the mesh is refined in this area. Here another advantage of the virtual element method becomes evident which is related to the possibility of inserting arbitrary number of nodes at each edge of an element in a consistent way. This yields a C^0 continuous mesh, see the magnification in the red circle on the right side of Fig. 11.32 and thus no hanging nodes occur when the mesh is refined, as it would be the case in a finite element discretization.

With increasing load a crack develops which is demonstrated in Fig. 11.34 for four different load stages showing the deformed configuration of the system. The crack starts, as is shown in Fig. 11.34a and then increases to the wall with a new crack developing, see Fig. 11.34b. The second crack grows further, see Fig. 11.34c until the mount finally fails with a third crack developing as depicted in Fig. 11.34d.

The associated load displacement curve is depicted in Fig. 11.33 where the total load is plotted versus the mid displacement of the bolt. Due to the small strain elastic behaviour this curve increases almost linearly up to point ①. Here, at a load of around 900 kN, the first crack starts to develop, see Fig. 11.34a leading to a damaged structure. However the system still has some residual stiffness and the load can be increased further while the second crack develops. Points ② and ③ are associated with the plots in Fig. 11.34b and c. Note that the slope of the load deflection curve has a lower gradient due to the softening of the mount by the first crack. At around 1650 kN, see point ③, the final failure of the mount is initiated and leads with a horizontal tangent of the load displacement curve to the state shown in Fig. 11.34d, which is related to point ④ in Fig. 11.33.

By combining the phase field method with the element cutting technology in Sect. 10.2, see also Hussein et al. (2020), it is possible to predict the cracked deformed configuration, as shown in Fig. 11.35a. In this deformation stage the structure exhibits two main cracks that lead to total failure of the mount. Additionally the von Mises stress is plotted in Fig. 11.35b just before total failure. This plot depicts the maximum stresses around the first crack on the left and the maximum stress at the tip of the crack on the right.

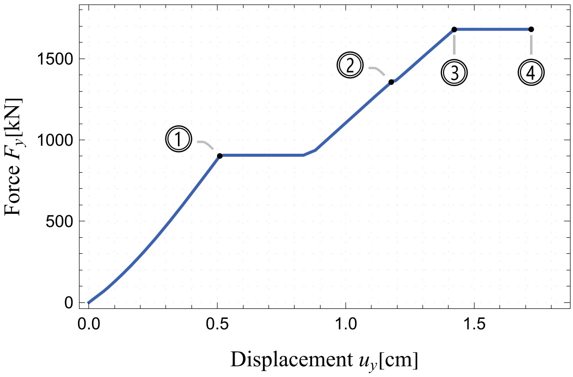
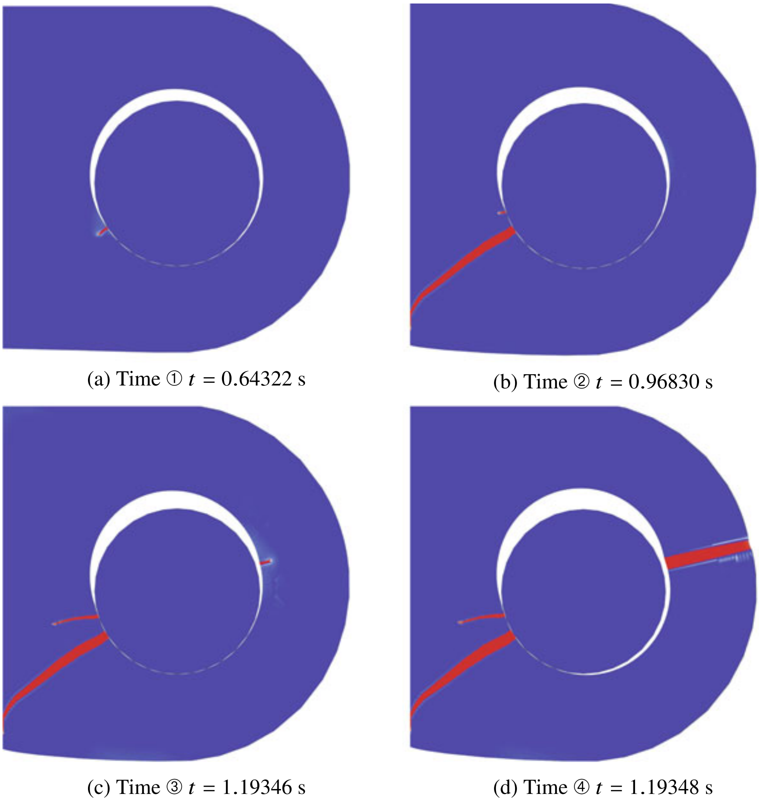


Fig. 11.33 Wall mounting of a bolt—load-deflection curve of the fracture process

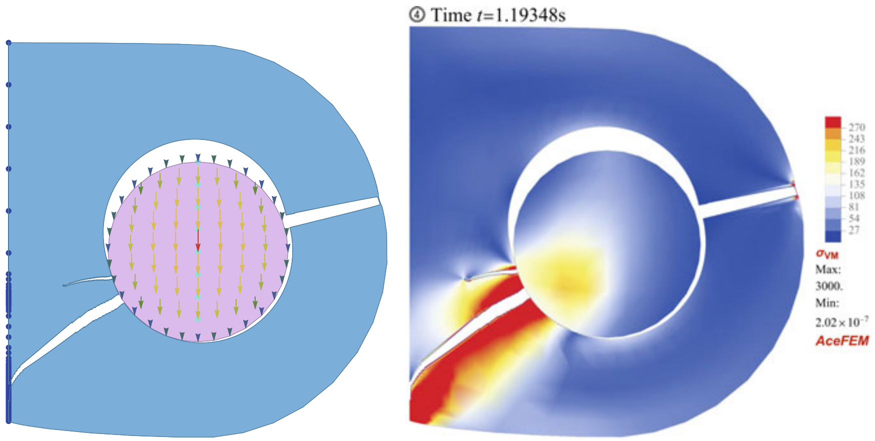

(a) Time ① t = 0.64322 s

(b) Time ② t = 0.96830 s

(c) Time ③ t = 1.19346 s

(d) Time ④ t = 1.19348 s

Fig. 11.34 Wall mounting of a bolt—fracture process



(a) fractured structure at failure load

(b) von Mises stresses

Fig. 11.35 Wall mounting of a bolt

References

Alart, P., and A. Curnier. 1991. A mixed formulation for frictional contact problems prone to Newton like solution methods. *Computer Methods in Applied Mechanics and Engineering* 92: 353–375.

Aldakheel, F., B. Hudobivnik, E. Artioli, L. BeirãodaVeiga, and P. Wriggers. 2020. Curvilinear virtual elements for contact mechanics. *Computer Methods in Applied Mechanics and Engineering* 372: 113394.

Bandeira, A.A., P. Wriggers, and P. de Mattos Pimenta. 2004. Homogenization methods leading to interface laws of contact mechanics. *International Journal for Numerical Methods in Engineering* 59: 173–195.

Barber, J.R. 2018. *Contact mechanics, solid mechanics and its applications*, vol. 250. Berlin: Springer.

Beirão da Veiga, L., F. Brezzi, and L. Marini. 2013. Virtual elements for linear elasticity problems. *SIAM, Journal of Numerical Analysis* 51: 794–812.

Beirão da Veiga, L., A. Russo, and G. Vacca. 2019. The virtual element method with curved edges. *ESAIM: Mathematical Modelling and Numerical Analysis* 53 (2): 375–404.

Belgacem, F.B., P. Hild, and P. Laborde. 1997. Approximation of the unilateral contact problem by the mortar finite element method. *Comptes Rendus de l'Académie des Sciences, Paris, Series* I (324): 123–127.

Chan, S.H., and I.S. Tuba. 1971. A finite element method for contact problems in solid bodies. *International Journal of Mechanical Sciences* 13: 615–639.

Christensen, P.W., A. Klarbring, J.S. Pang, and N. Strömberg. 1998. Formulation and comparison of algorithms for frictional contact problems. *International Journal for Numerical Methods in Engineering* 42: 145–173.

Cihan, M., B. Hudobivnik, J. Korelc, and P. Wriggers. 2022. A virtual element method for 3D contact problems with non-conforming meshes. *Computer Methods in Applied Mechanics and Engineering* 392: 115385.

Cottrell, J.A., T.J.R. Hughes, and Y. Bazilevs. 2009. *Isogeometric analysis: Toward Integration of CAD and FEA*. New York: Wiley.

Coulomb, C. A. 1785. The theory of simple machines (in french). *Memoirs of Mathematics and Physics Academy of Sciences* 10: 161–331.

De Lorenzis, L., İ Temizer, P. Wriggers, and G. Zavarise. 2011. A large deformation frictional contact formulation using NURBS-based isogeometric analysis. *International Journal for Numerical Methods in Engineering* 87: 1278–1300.

De Lorenzis, L., P. Wriggers, and G. Zavarise. 2012. Isogeometric analysis of 3d large deformation contact problems with the augmented Lagrangian formulation. *Computational Mechanics* 49: 1–20.

De Lorenzis, L., J. Evans, T. Hughes, and A. Reali. 2015. Isogeometric collocation: Neumann boundary conditions and contact. *Computer Methods in Applied Mechanics and Engineering* 284: 21–54.

De Lorenzis, L., P. Wriggers, and C. Weißenfels. 2017. Computational contact mechanics with the finite element method. In *Encyclopedia of computational mechanics, part 1, solids and structures*, 2nd ed., ed. E. Stein, R. de Borst, and T.J.R. Hughes, 1–45. New York: Wiley.

Fischer, K.A., and P. Wriggers. 2005. Frictionless 2d contact formulations for finite deformations based on the mortar method. *Computational Mechanics* 36: 226–244.

Fischer, K.A., and P. Wriggers. 2006. Mortar based frictional contact formulation for higher order interpolations using the moving friction cone. *Computer Methods in Applied Mechanics and Engineering* 195: 5020–5036.

Flemisch, B., M.A. Puso, and B. Wohlmuth. 2005. A new dual mortar method for curved interfaces: 2d elasticity. *International Journal for Numerical Methods in Engineering* 63: 813–832.

Franke, D., A. Düster, V. Nübel, and E. Rank. 2010. A comparison of the h-, p-, hp-, and rp-version of the fem for the solution of the 2d hertzian contact problem. *Computational Mechanics* 45: 513–522.

Giannokopoulos, A.E. 1989. The return mapping method for the integration of friction constitutive relations. *Computers and Structures* 32: 157–168.

Hallquist J.O. 1984. Nike 2d: An implicit, finite deformation, finite element code for analyzing the static and dynamic response of two-dimensional solids. Technical Report Report UCRL-52678, Lawrence Livermore National Laboratory, University of California, Livermore, CA.

Hallquist, J.O., G.L. Goudreau, and D.J. Benson. 1985. Sliding interfaces with contact-impact in large-scale lagrange computations. *Computer Methods in Applied Mechanics and Engineering* 51: 107–137.

Hertz, H. 1882. Über die Berührung fester elastischer Körper. *Journal für die Reine und Angewandte Mathematik* 29: 156–171.

Hughes, T.R.J., R.L. Taylor, J.L. Sackman, A. Curnier, and W. Kanoknukulchai. 1976. A finite element method for a class of contact-impact problems. *Computer Methods in Applied Mechanics and Engineering* 8: 249–276.

Hussein, A., B. Hudobivnik, and P. Wriggers. 2020. A combined adaptive phase field and discrete cutting method for the prediction of crack paths. *Computer Methods in Applied Mechanics and Engineering* 372: 113329.

Johnson, K.L. 1985. *Contact mechanics*. Cambridge: Cambridge University Press.

Kikuchi, N., and J.T. Oden. 1988. *Contact problems in elasticity: A study of variational inequalities and finite element methods*. Philadelphia: SIAM.

Krstulovic-Opara, L., P. Wriggers, and J. Korelc. 2002. A C^1-continuous formulation for 3d finite deformation frictional contact. *Computational Mechanics* 29: 27–42.

Kruse, R., N. Nguyen-Thanh, L. De Lorenzis, and T. Hughes. 2015. Isogeometric collocation for large deformation elasticity and frictional contact. *Computer Methods in Applied Mechanics and Engineering* 296: 73–112.

Laursen, T.A. 2002. *Computational contact and impact mechanics*. Berlin, New York, Heidelberg: Springer.

Laursen, T.A., and J.C. Simo. 1993. Algorithmic symmetrization of Coulomb frictional problems using augmented Lagrangians. *Computer Methods in Applied Mechanics and Engineering* 108: 133–146.

Laursen, T.A., M.A. Puso, and J. Sanders. 2012. Mortar contact formulations for deformable-deformable contact: past contributions and new extensions for enriched and embedded interface formulations. *Computer methods in applied mechanics and engineering* 205: 3–15.

Lengiewicz, J., J. Korelc, and S. Stupkiewicz. 2011. Automation of finite element formulations for large deformation contact problems. *International Journal for Numerical Methods in Engineering* 85 (10): 1252–1279.

McDevitt, T.W., and T.A. Laursen. 2000. A mortar-finite element formulation for frictional contact problems. *International Journal for Numerical Methods in Engineering* 48: 1525–1547.

Mindlin, R. 1949. Compliance of elastic bodies in contact. *Journal of Applied Mechanics, ASME* 16: 259–268.

Padmanabhan, V., and T. Laursen. 2001. A framework for development of surface smoothing procedures in large deformation frictional contact analysis. *Finite Elements in Analysis and Design* 37: 173–198.

Pietrzak, G., and A. Curnier. 1999. Large deformation frictional contact mechanics: continuum formulation and augmented lagrangean treatment. *Computer Methods in Applied Mechanics and Engineering* 177: 351–381.

Pietrzak G. and A. Curnier. 1997. Continuum mechanics modeling and augmented lagrange formulation of multibody, large deformation frictional contact problems. In *Proceedings of COMPLAS5*, ed. D.R. Owen, E. Hinton and E. Onate, 878–883. CIMNE, Barcelona.

Popp, A., M.W. Gee, and W.A. Wall. 2009. A finite deformation mortar contact formulation using a primal-dual active set strategy. *International Journal for Numerical Methods in Engineering* 79 (11): 1354–1391.

Popp, A., M. Gitterle, M. Gee, and W.A. Wall. 2010. A dual mortar approach for 3d finite deformation contact with consistent linearization. *International Journal for Numerical Methods in Engineering* 83 (11): 1428–1465.

Puso, M.A. 2004. A 3D mortar method for solid mechanics. *International Journal for Numerical Methods in Engineering* 59 (3): 315–336.

Puso, M.A., and T.A. Laursen. 2004. A mortar segment-to-segment contact method for large deformation solid mechanics. *Computer Methods in Applied Mechanics and Engineering* 193: 601–629.

Simo, J.C., and T.A. Laursen. 1992. An augmented Lagrangian treatment of contact problems involving friction. *Computers and Structures* 42: 97–116.

Simo, J.C., P. Wriggers, and R.L. Taylor. 1985. A perturbed Lagrangian formulation for the finite element solution of contact problems. *Computer Methods in Applied Mechanics and Engineering* 50: 163–180.

Temizer, İ, and P. Wriggers. 2008. A multiscale contact homogenization technique for the modeling of third bodies in the contact interface. *Computer Methods in Applied Mechanics and Engineering* 198: 377–396.

Temizer, İ, P. Wriggers, and T.J.R. Hughes. 2011. Contact treatment in isogeometric analysis with NURBS. *Computer Methods in Applied Mechanics and Engineering* 200: 1100–1112.

Temizer, İ, P. Wriggers, and T.J.R. Hughes. 2012. Three-dimensional mortar-based frictional contact treatment in isogeometric analysis with NURBS. *Computer Methods in Applied Mechanics and Engineering* 209–211: 115–128.

Tur, M., F. Fuenmayor, and P. Wriggers. 2009. A mortar-based frictional contact formulation for large deformations using Lagrange multipliers. *Computer Methods in Applied Mechanics and Engineering* 198: 2860–2873.

Wagner, P., P. Wriggers, C. Klapproth, and C. Prange. 2015. Multiscale FEM approach for hysteresis friction of rubber on rough surfaces. *Computer Methods in Applied Mechanics and Engineering* 296: 150–168.

Wilson, E.A., and B. Parsons. 1970. Finite element analysis of elastic contact problems using differential displacements. *International Journal for Numerical Methods in Engineering* 2: 387–395.

Wohlmuth, B.I. 2000. A mortar finite element method using dual spaces for the lagrange multiplier. *SIAM, Journal of Numerical Analysis* 38: 989–1012.

Wohlmuth, B., and R. Krause. 2004. Monotone methods on non-matching grids for non linear contact problems. *SISC* 25: 324–347.

Wriggers, P. 1987. On consistent tangent matrices for frictional contact problems. In *Proceedings of NUMETA 87* ed. G. Pande and J. Middleton. Dordrecht: M. Nijhoff Publishers.

Wriggers, P. 2006. *Computational Contact Mechanics*, 2nd ed. Berlin, Heidelberg, New York: Springer.

Wriggers, P., and A. Haraldsson. 2003. A simple formulation for two-dimensional contact problems using a moving friction cone. *Communications in Numerical Methods in Engineering* 19: 285–295.

Wriggers, P., and L. Krstulovic-Opara. 2004. The moving friction cone approach for three-dimensional contact simulations. *International Journal of Computational Methods* 1: 105–119.

Wriggers, P., and J. Reinelt. 2009. Multi-scale approach for frictional contact of elastomers on rough rigid surfaces. *Computer Methods in Applied Mechanics and Engineering* 198: 1996–2008.

Wriggers, P., and W. Rust. 2019. A virtual element method for frictional contact including large deformations. *Engineering Computations* 36: 2133–2161.

Wriggers, P., and J. Simo. 1985. A note on tangent stiffnesses for fully nonlinear contact problems. *Communications in Applied Numerical Methods* 1: 199–203.

Wriggers, P., L. Krstulovic-Opara, and J. Korelc. 2001. Smooth C^1- interpolations for two-dimensional frictional contact problems. *International Journal for Numerical Methods in Engineering* 51: 1469–1495.

Wriggers, P., W. Rust, and B. Reddy. 2016. A virtual element method for contact. *Computational Mechanics* 58: 1039–1050.

Wriggers, P., B. Reddy, W. Rust, and B. Hudobivnik. 2017. Efficient virtual element formulations for compressible and incompressible finite deformations. *Computational Mechanics* 60: 253–268.

Wriggers, P., B. Hudobivnik, and F. Aldakheel. 2020. A virtual element formulation for general element shapes. *Computational Mechanics* 66: 963–977.

Zavarise, G., and L. De Lorenzis. 2009. A modified node-to-segment algorithm passing the contact patch test. *International Journal for Numerical Methods in Engineering* 79 (4): 379–416.

Chapter 12
Virtual Elements for Computational Homogenization of Polycrystalline Materials

Heterogeneous materials (such as composites, bones, wood, concrete and metallic-polycrystalline materials) consist of complicated constituents across scales with complex material response. These materials, even with similar properties at macroscopic level, can behave differently at micro-scale. The response of such materials is often related to non-uniform and anisotropic behaviour at microscopic level.

For a predictive numerical simulation of the constitutive properties of heterogeneous materials at least two scales have to be considered, leading to a multi-scale approach. From the modeling point of view, two multi-scale approaches are of importance, namely concurrent and hierarchical multi-scale techniques. These can be classified by the macro characteristic length scale l_{macro} and its micro domain counterpart l_{micro}. For an overview of different techniques, see the textbooks Zohdi and Wriggers (2005) and Fish (2014).

The concurrent multi-scale method assumes $l_{micro} \approx l_{macro}$, as addressed in e.g. Fish and Wagiman (1993), Lloberas-Valls et al. (2012) and Aldakheel et al. (2021b). For the hierarchical multi-scale method, the average size of the heterogeneous micro domain is much smaller than the macro specimen size, such that $l_{micro} \ll l_{macro}$, see e.g. Miehe et al. (1999a), Michel et al. (1999) and Hain and Wriggers (2008a). The corresponding property is generally known as scale separation and the associated computational homogenization approaches are then based on the Hill-Mandel principle, well documented in Hill (1965) and Michel et al. (1999). The aim of all mentioned multi-scale simulation techniques is the reduction of uncertainties and empirical assumptions while simultaneously increasing the accuracy of the material response.

For a better understanding of the scale bridging, consider the multi-scale character of hybrid bearing bush, illustrated in Fig. 12.1 which depicts the presence of the crystalline microstructure of the different metals, steel and aluminum. We observe that each polycrystalline material has a different structure with many crystals at the micro-scale. The behaviour of the crystals at micro-scale determine the constitutive response at macro-scale. High-end industrial applications in engineering, energy

P. Wriggers et al., *Virtual Element Methods in Engineering Sciences*,
https://doi.org/10.1007/978-3-031-39255-9_12

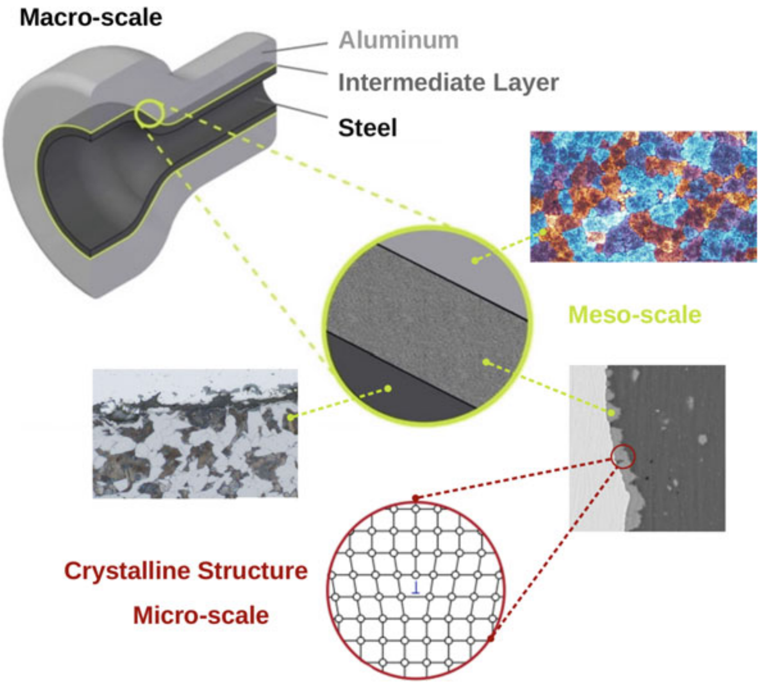


Fig. 12.1 Multi-scale design of a hybrid bearing bush, considering different length-scales, see Behrens et al. (2020)

and medical technologies require consideration of microscopic material behaviour in order to predict the required system-output at macro-scale accurately.

Different approaches exist for the determination of the effective-properties of heterogeneous materials at micro-scale. For composites with recurrent microstructure and linear constitutive behaviour, those properties can be determined by solving a sufficient number of unit cell problems along with the corresponding boundary conditions. Furthermore, an asymptotic homogenization approach can be considered as has been documented in Bensoussan et al. (1978) and Sanchez-Palencia (1980). In the case of an irregular microstructure, the effective-properties cannot be computed exactly. The available methods in such situations are limited to the computation of upper and lower bounds for the effective stiffness, as outlined in Voigt (1887) and Reuss (1929). The analytical methods were further extended in the work of Hashin and Shtrikman (1962), Hashin and Shtrikman (1963) by considering variational principles, leading to improved estimations. Hill (1965) developed a self-consistent method by embedding a single inclusion into an infinite domain of the initially unknown effective matrix material. The variational principles of Ponte Castañeda (1991), Ponte Castañeda (1992), Suquet (1993) and Talbot and Willis (1992) are concerned with general bounds for specific classes of nonlinear composites.

The basic principles of the homogenization can also be employed within a numerical simulation approach, leading to homogenization procedures based on representative volume elements ($\mathcal{RVE}$). We refer to Hill (1972), Suquet (1987) and Nemat-Nasser and Hori (1999) for fundamental homogenization principles of local mechanical response. The applications of these principles to computational homogenization can be found in e.g. Zohdi et al. (1996), Zohdi and Wriggers (1999), Zohdi and Wriggers (2001b), Zohdi and Wriggers (2005), Wriggers and Moftah (2006), Temizer and Zohdi (2007), Temizer and Wriggers (2007), Zohdi (2008), Temizer and Wriggers (2008b), Hain and Wriggers (2008a), Wellmann et al. (2008), Temizer and Wriggers (2011) and citations therein. These contributions cover the theoretical background of the numerical techniques and show applications to different heterogeneous materials.

Another important methodology is the two-scale computational homogenization which treats an engineering problem at macro-level by determining the effective-properties within a nested approach. It is also known under the synonym (FE^2) which should be written for virtual elements as (VE^2). In more detail, two nested boundary-value-problems are solved along with a corresponding scale transition law, see for instance Smit et al. (1998), Miehe et al. (1999b), Miehe et al. (1999a), Feyel and Chaboche (2000), Terada and Kikuchi (2001), Kouznetsova et al. (2002), Miehe (2002), Hain and Wriggers (2008a), Geers et al. (2010), Schröder and Keip (2012), Chatzigeorgiou et al. (2014) and Javili et al. (2013). The material behaviour is analyzed in this approach at microscopic level by employing as homogenization technique the concept of representative volume elements, which then is used to compute the material response at macro-level.

In this chapter, the focus is on the computational homogenization approach with application to polycrystalline materials. For materials like steel or aluminum scale separation can be assumed since the characteristic length of the micro-scale is much smaller than its macro counterpart, see Fig. 12.2. This fact is the general prerequisite for the use of homogenization procedures.

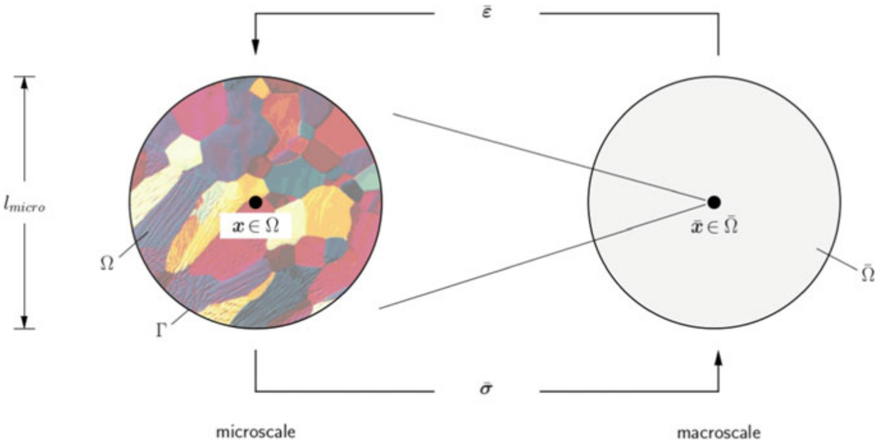


Fig. 12.2 Micro- and macro-scale of polycrystalline materials

The following challenges have to be considered for numerical simulation models related to computational homogenization

- complex shape of grains in the microstructure,
- presence of strong anisotropy and
- random orientation of material symmetries.

Especially the first challenge can be overcome by the virtual element method which allows to model the complex shape of polycrystals with just one element and thus is an obvious choice for this class of materials, see e.g. Marino et al. (2019) and Böhm et al. (2021).

12.1 Micro-to-Macro Transition Concept

The aim of this section is to provide an overview regarding two different micro-to-macro transition concepts which are

- computational homogenization and
- two-scale modeling (FE^2/VE^2).

To this end, a material point $\bar{\boldsymbol{x}}$ within the solid $\bar{\Omega}$ is considered at the macro-level, see Fig. 12.3. Kinematics and balance relations are given by the continuum mechanics framework. Additionally, the constitutive behaviour is needed in order to solve a boundary value problem in $\bar{\Omega}$.

12.1.1 The Concept of Representative Volume Elements

The material behaviour at the macroscale is comprehensively described by the microscopic geometry and constituents. When the material body consists of a heterogeneous microstructure that governs its response, a homogenization approach can be necessary to determine the effective material properties at macro-level. The homogenization approach relies on the assumption that the two considered scales are well separated, i.e. a typical size on the macro-scale is much larger than a typical size of the underlying microstructure, $l_{macro} \gg l_{micro}$, as depicted in Fig. 12.2. This separation of length scale assumption is a crucial requirement for the application of the concept of representative volume elements. At the micro-scale, the representative volume element $\mathcal{RVE}$ models the microstructure that corresponds to the macroscopic material point. By enforcing boundary conditions which are determined by the macroscopic state of deformation, a boundary-value-problem is defined for the representative volume element. At this level both, the governing balance equations and the constitutive behaviour are known. The goal is to return the constitutive information from a finer scale to the macro-level (coarser scale) which then can be either

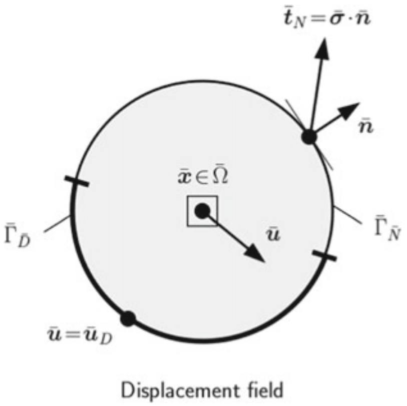


Fig. 12.3 Macroscopic boundary-value-problem. Mechanical displacement field $\bar{\boldsymbol{u}}$ defined on the solid $\bar{\Omega}$

used point-wise within a FE^2 approach or as effective properties in a macro computation of the problem.

The $\mathcal{RVE}$ acts as a statistically representative portion of the heterogeneous microstructure (grains separated by grain boundary, voids, inclusion, crack, and other similar defects), see Nemat-Nasser and Hori (1999). Its size must be chosen such that it is large enough to be representative which implies that it sufficiently accounts for the character and distribution of heterogeneities. Nevertheless, it should be much smaller than the structure or specimen considered at macro-level to ensure on one hand a scale separation, and to achieve on the other hand an increased efficiency. If the material or geometric properties of the underlying microstructures vary spatially within a macro specimen, as e.g. in functionally graded materials, the representative volume element needs to be adjusted for different macro regions.

12.1.2 Macroscopic Boundary Value Problem

Let $\bar{\Omega} \subset \mathcal{R}^\delta$ denote a macroscopic body with dimension $\delta \in \{2, 3\}$, as sketched in Fig. 12.3. We study the mechanical deformation of the body under quasi-static loading. In what follows, $\bar{\nabla}(\bullet) = \frac{\partial(\bullet)}{\partial \bar{x}}$ denotes the gradient of the macroscopic field $(\bullet)$. The primary variable field is the displacement field $\bar{\boldsymbol{u}}(\bar{\boldsymbol{x}})$ of the material point $\bar{\boldsymbol{x}} \in \bar{\Omega}$. The kinematic relation arising from the geometrically linear framework is the strain tensor $\bar{\varepsilon} = \bar{\nabla}_s \bar{\boldsymbol{u}} = \frac{1}{2}(\bar{\nabla}\bar{\boldsymbol{u}} + \bar{\nabla}^T \bar{\boldsymbol{u}})$, see (2.9).

We assume the existence of a macroscopic potential energy $\bar{U}$. Based on this assumption, we can formulate a variational structure resulting in a minimization problem for the determination of macroscopic primary variable

$$\{\bar{\boldsymbol{u}}\} = \mathrm{Arg}\left\{ \inf_{\bar{\boldsymbol{u}}} \bar{U}(\bar{\boldsymbol{u}}) \right\}, \tag{12.1}$$

where $\bar{U}$ is defined as

$$\bar{U}(\bar{\boldsymbol{u}}) = \int_{\bar{\Omega}} \bar{\Psi}(\bar{\varepsilon})\, \mathrm{d}\bar{\Omega} - \bar{\mathcal{P}}_{ext}(\bar{\boldsymbol{u}}) \, . \tag{12.2}$$

The potential energy $\bar{U}$ will be fully determined by another variational formulation using a $\mathcal{RVE}$ at the micro-scale. The mechanical loading contribution follows from

$$\bar{\mathcal{P}}_{ext}(\bar{\boldsymbol{u}}) = \int_{\bar{\Omega}} \bar{\boldsymbol{b}} \cdot \bar{\boldsymbol{u}}\, \mathrm{d}\bar{\Omega} + \int_{\bar{\Gamma}_{\bar{N}}} \bar{\boldsymbol{t}}_N \cdot \bar{\boldsymbol{u}}\, \mathrm{d}\bar{\Gamma} \, . \tag{12.3}$$

The given macroscopic body force per unit volume is introduced as $\bar{\boldsymbol{b}}$, whereas the tractions on the Neumann boundary $\bar{\Gamma}_{\bar{N}}$ are depicted as $\bar{\boldsymbol{t}}_N$. Furthermore, consider a decomposition of the surface $\bar{\Gamma} = \bar{\Gamma}_{\bar{D}} \cup \bar{\Gamma}_{\bar{N}}$ into a part $\bar{\Gamma}_{\bar{D}}$ where the displacements are prescribed and a part $\bar{\Gamma}_{\bar{N}}$ with given tractions, along with $\bar{\Gamma}_{\bar{D}} \cap \bar{\Gamma}_{\bar{N}} = \emptyset$.

The Euler-Lagrange equations of (12.1) result in the equilibrium equations describing the macroscopic problem for the quasi static case. They lead to the balance of linear momentum $\mathrm{div}\bar{\boldsymbol{\sigma}} + \bar{\boldsymbol{b}} = \boldsymbol{0} \quad \text{in } \bar{\Omega}$, see (2.22), along with the Neumann-type boundary conditions $\bar{\boldsymbol{\sigma}} \cdot \bar{\boldsymbol{n}} = \bar{\boldsymbol{t}}_N$, see (2.24), on $\bar{\Gamma}_{\bar{N}}$. The macroscopic stresses

$$\bar{\boldsymbol{\sigma}} = \frac{\partial\, \bar{\Psi}(\bar{\varepsilon})}{\partial\, \bar{\varepsilon}} \, , \tag{12.4}$$

are governed by the macroscopic energy function $\bar{\Psi}$ obtained by a homogenization using the $\mathcal{RVE}$ defined at the micro-scale.

12.1.3 Microscopic Boundary Value Problem

Let $\Omega \subset \mathcal{R}^{\delta}$ denote a periodic microstructure ($\mathcal{RVE}$) as depicted in Fig. 12.4. Hereby, $\nabla(\bullet) = \frac{\partial(\bullet)}{\partial \boldsymbol{x}}$ denotes the gradient of the microscopic field $(\bullet)$. The primary variable field is the displacement field $\boldsymbol{u}(\boldsymbol{x})$ of the material point $\boldsymbol{x} \in \Omega$. The microscopic linear strain tensor ε is the symmetric part of the displacement gradient $\varepsilon = \nabla_s \boldsymbol{u} = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla^T \boldsymbol{u})$. We now postulate a variational principle of homogenization that determines the macroscopic energy $\bar{\Psi}$, introduced in (12.2), as follows

$$\bar{\Psi}(\bar{\varepsilon}) = \inf_{\boldsymbol{u}} \frac{1}{\Omega} \int_{\Omega} \Psi(\varepsilon)\, \mathrm{d}\Omega \, . \tag{12.5}$$

This definition is conceptually in line with the formulations outlined in Ponte Castañeda and Suquet (1997), Miehe (2002). At the micro-level, we identify the stresses as

$$\boldsymbol{\sigma}(\varepsilon) = \frac{\partial\, \Psi(\varepsilon)}{\partial\, \varepsilon} \, , \tag{12.6}$$

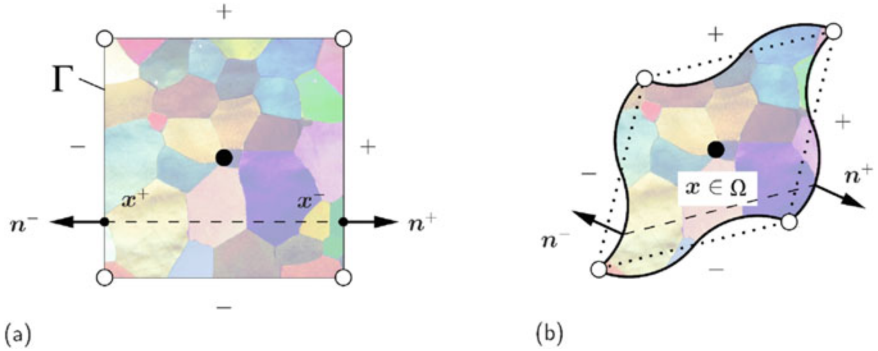


Fig. 12.4 Periodic microstructure, where the surface of $\mathcal{RVE}$ is decomposed into $\Gamma = \Gamma^+ \cup \Gamma^-$ in (**a**) along with periodic boundary conditions for the displacement field in (**b**)

that follow from the strain energy function Ψ valid at the micro-scale and depending at the local strains. With this stress-strain relation, the variation of the principle (12.5) yields

$$\frac{1}{\Omega}\int_\Omega \big(-\operatorname{div}[\boldsymbol{\sigma}]\big)\cdot\delta\boldsymbol{u}\,\mathrm{d}\Omega + \frac{1}{\Omega}\int_\Gamma \big(\boldsymbol{\sigma}\cdot\boldsymbol{n}\big)\cdot\delta\boldsymbol{u}\,\mathrm{d}\Gamma = 0\ . \tag{12.7}$$

The Euler-Lagrange equation for the variational principle (12.5) is $\operatorname{div}[\boldsymbol{\sigma}] = \boldsymbol{0}$ for the microstructure Ω.

12.1.4 Homogenization and Macro-Homogeneity Conditions

The microscopic boundary-value-problem is linked to the macroscopic one by a scale bridging approach accounting for homogenized quantities which are defined on the boundary/volume of the representative volume element. The macroscopic strain $\bar{\varepsilon}$ and the associated work-conjugate stress $\bar{\boldsymbol{\sigma}}$ characterize the effective strains and stresses. They follow by averaging the microscopic strains ε and stresses $\boldsymbol{\sigma}$ over the volume of the micro-structure Ω

$$\bar{\varepsilon} = \frac{1}{\Omega}\int_\Omega \varepsilon\,\mathrm{d}\Omega = \langle\varepsilon\rangle \quad\text{and}\quad \bar{\boldsymbol{\sigma}} = \frac{1}{\Omega}\int_\Omega \boldsymbol{\sigma}\,\mathrm{d}\Omega = \langle\boldsymbol{\sigma}\rangle\ , \tag{12.8}$$

where $\langle(\bullet)\rangle = \frac{1}{\Omega}\int_\Omega(\bullet)\,\mathrm{d}\Omega$ denotes the volume average operator in Ω. The average strain and stress theorem allows, for perfectly bonded materials and by neglecting of body forces at the micro-scale, to shift the volume integrals to surface integrals, see e.g. Zohdi and Wriggers (2005). With $\operatorname{div}(\boldsymbol{\sigma}\otimes\boldsymbol{x}) = \operatorname{div}\boldsymbol{\sigma}\otimes\boldsymbol{x} + \boldsymbol{\sigma}\,\nabla\boldsymbol{x}$, $\boldsymbol{\sigma} = \boldsymbol{\sigma}^T$ and the local equilibrium $\operatorname{div}\boldsymbol{\sigma} = \boldsymbol{0}$ it follows from Gauss' theorem

$$\begin{aligned}\langle \varepsilon \rangle &= \frac{1}{\Omega}\int_{\Omega} \varepsilon \, \mathrm{d}\Omega = \frac{1}{\Omega}\int_{\Omega} \nabla_s \boldsymbol{u} \, \mathrm{d}\Omega = \frac{1}{\Omega}\int_{\Gamma} \operatorname{sym}(\boldsymbol{u} \otimes \boldsymbol{n}) \, \mathrm{d}\Gamma \quad \text{and} \\ \langle \boldsymbol{\sigma} \rangle &= \frac{1}{\Omega}\int_{\Omega} \boldsymbol{\sigma} \, \mathrm{d}\Omega = \frac{1}{\Omega}\int_{\Gamma} (\boldsymbol{\sigma} \otimes \boldsymbol{x})\, \boldsymbol{n} \, \mathrm{d}\Omega = \frac{1}{\Omega}\int_{\Gamma} \operatorname{sym}(\boldsymbol{t}_N \otimes \boldsymbol{x}) \, \mathrm{d}\Gamma \, .\end{aligned} \tag{12.9}$$

A link between the macro- and micro-scale is provided by the Hill-Mandel macro-homogeneity condition, proposed by Hill (1972). It states that the virtual work on the macro-scale is equal to that in the micro-scale for the $\mathcal{RVE}$

$$\bar{\boldsymbol{\sigma}} \cdot \delta\bar{\varepsilon} = \langle \boldsymbol{\sigma} \cdot \delta\varepsilon \rangle = \frac{1}{\Omega}\int_{\Omega} \boldsymbol{\sigma} \cdot \delta\varepsilon \, \mathrm{d}\Omega = \frac{1}{\Omega}\int_{\Gamma} \boldsymbol{t}_N \cdot \delta\boldsymbol{u} \, \mathrm{d}\Gamma \, . \tag{12.10}$$

Let us now assume that we have at the micro-scale sample of aggregate polycrystalline material with a linear elastic but anisotropic material response at each grains which can be described by $\boldsymbol{\sigma} = \mathbb{C}\,\varepsilon$. The effective constitutive relation is given by $\bar{\boldsymbol{\sigma}} = \bar{\mathbb{C}}\,\bar{\varepsilon}$ where the effective constitutive tensor $\bar{\mathbb{C}}$ is defined via the heterogeneous properties at the micro-scale, represented by $\mathbb{C}$.

Appropriate boundary conditions have to be prescribed at the $\mathcal{RVE}$-level for the solution of the microscopic boundary-value-problem. For the micro-elastic model under consideration we focus on three types of boundary constraints:

1. **Dirichlet-type boundary conditions**: linear displacements prescribed at the boundary Γ lead to a homogeneous strain field, being imposed in the full microstructure, i.e. $\varepsilon = \bar{\varepsilon}$ in Ω. Hence, the microscopic strain is identical to the macro-strain at any point $\boldsymbol{x} \in \Omega$ of the microstructure. This is called Voigt-(Taylor) assumption referring to Voigt (1887) and yields an upper bound of the stiffness of the $\mathcal{RVE}$. In more detail we obtain $\bar{\boldsymbol{\sigma}} = \langle \boldsymbol{\sigma} \rangle = \langle \mathbb{C}\,\varepsilon \rangle = \langle \mathbb{C} \rangle\,\bar{\varepsilon}$ leading to $\bar{\mathbb{C}} = \langle \mathbb{C} \rangle$.
2. **Neumann-type boundary conditions**: constant stress prescribed at the boundary Γ, zero micro-tractions. The so-called Reuss-(Sachs) bound, Reuss (1929), provides a lower bound of the stiffness of the microstructure. Here, a homogeneous stress $\boldsymbol{\sigma} = \bar{\boldsymbol{\sigma}}$ is applied in the full domain Ω which gives $\bar{\varepsilon} = \langle \varepsilon \rangle = \langle \mathbb{C}^{-1}\,\boldsymbol{\sigma} \rangle = \langle \mathbb{C}^{-1} \rangle\,\bar{\boldsymbol{\sigma}}$ leading to $\bar{\mathbb{C}} = \langle \mathbb{C}^{-1} \rangle^{-1}$.
3. **Periodic boundary conditions**: periodicity of all primary fields on opposite surfaces Γ^+ and Γ^- of the microstructure.

The Dirichlet (see 1.) and Neumann (see 2.) boundary conditions can be applied for computational homogenization and then yield upper Voigt and lower Reuss bounds. They do not play a crucial role in our application of computational homogenization using virtual elements were we use periodic boundary conditions, see Sect. 12.3. Although the periodic constraints (see 3.) are only applicable for perfectly periodic microstructures, it turns out for many cases that they yield better results than either the Dirichlet-type constraints or the Neumann-type conditions. However, the advantage of one boundary condition over the other diminishes for a large sample of material (large $\mathcal{RVE}$). For more details, we refer to the work of Zohdi and Wriggers (2005) and the references cited therein.

For the periodic boundary conditions, plotted in Fig. 12.4, the surface of the $\mathcal{RVE}$ decomposes into two parts $\Gamma = \Gamma^+ \cup \Gamma^-$ with normals $\boldsymbol{n}^+$ and $\boldsymbol{n}^- = -\boldsymbol{n}^+$ at associated points $\boldsymbol{x}^+ \in \Gamma^+$ and $\boldsymbol{x}^- \in \Gamma^-$. The deformation is extended by a fine scale fluctuation field, marked with a tilde $(\tilde{\bullet})$,

$$\boldsymbol{u} = \bar{\varepsilon} \cdot \boldsymbol{x} + \tilde{\boldsymbol{u}}\,, \tag{12.11}$$

around the macro-modes. The Hill conditions

$$\frac{1}{\Omega} \int_\Gamma \boldsymbol{t}_N \cdot \delta\tilde{\boldsymbol{u}} \,\mathrm{d}\Gamma = 0 \tag{12.12}$$

provides an additional constraint for the fluctuation fields. This constraint can be satisfied for periodic fluctuations and anti-periodic tractions at the boundary Γ resulting in

$$\tilde{\boldsymbol{u}}^+ = \tilde{\boldsymbol{u}}^- \quad \text{and} \quad \boldsymbol{t}_N^+ = -\boldsymbol{t}_N^- \quad \text{on} \quad \Gamma\,, \tag{12.13}$$

at associated points $\boldsymbol{x}^+ \in \Gamma^+$ and $\boldsymbol{x}^- \in \Gamma^-$. Such boundary conditions are the most reasonable choice for the homogenization analysis under consideration, even for microstructures which are non-periodic, see for instance Terada et al. (2000) and Segurado and Llorca (2002). Generally there exist further possibilities to enforce the boundary constraints. One is the so called boxed method, see e.g. Hain and Wriggers (2008b) and Heinze et al. (2015), in which the $\mathcal{RVE}$ is embedded in a homogeneous solid of a certain thickness which then is impinged by a Dirichlet boundary condition.

12.1.5 Computational Homogenization Approach

The effective material properties are calculated using computational homogenization, as well documented in the works of Zohdi and Wriggers (2001a), Zohdi and Wriggers (2005). As discussed in Sect. 12.1.3, the representative volume elements ($\mathcal{RVE}$) define the micro-scale. Those $\mathcal{RVE}$s are either artificially generated or taken directly from filtered CT-data.

There exist two ways to determine the effective properties of the material response at the macroscale using computational homogenization.

- The first method relies on a **testing procedure** on $\mathcal{RVE}$ level. It yields the components of the effective constitutive tensor $\bar{\mathbb{C}}$ which is the mapping between the averaged stresses and strains, and provides the constitutive properties of the micro-heterogeneous material, see (12.9),

$$\langle \boldsymbol{\sigma} \rangle = \bar{\mathbb{C}}\, \langle \boldsymbol{\varepsilon} \rangle\,. \tag{12.14}$$

The effective constitutive tensor $\bar{\mathbb{C}}$ has generally 36 constitutive constants which can be determined by specifying 6 independent loadings (note that stresses and strains are symmetric and have only 6 components)[1]

$$\boldsymbol{u}|_\Gamma = \Xi^L \boldsymbol{x} \quad \text{with} \quad L \in \{1, \ldots, 6\} \tag{12.15}$$

where Ξ^L are different constant strains, e.g.

$$\Xi^1 = \begin{bmatrix} q & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \Xi^3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & q \end{bmatrix}, \quad \Xi^4 = \begin{bmatrix} 0 & q & 0 \\ q & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \Xi^6 = \begin{bmatrix} 0 & 0 & q \\ 0 & 0 & 0 \\ q & 0 & 0 \end{bmatrix} \tag{12.16}$$

that just represent stretch and shear with a constant load parameter q.
By applying these loads at the boundary of the discretized $\mathcal{RVE}$ one can compute the stress response $\boldsymbol{\sigma}$ at microscale and by averaging obtains $\langle\boldsymbol{\sigma}\rangle$. Each independent loading yields 6 equations which are employed to determine the relation between average strain and stress and thus determine all components of $\bar{\mathbb{C}}$. For further details, see Zohdi and Wriggers (2005).

- The second approach is based on **sensitivity analysis** which evaluates the contributions of the representative volume element $\mathcal{RVE}$ with regard to loading conditions at the $\mathcal{RVE}$ boundary Γ. Hereby, only one general loading condition has to be applied which leads to micro-structural stresses and strains, as described in the previous sections. The boundary conditions have to satisfy the Hill's energy criterion (12.12). Thereafter, homogenization is used to describe the averaged material behaviour at macroscale.
  The $\mathcal{RVE}$, defined at the microscale, consists of g different grains. Each grain $g = 1, \ldots, G$ is assumed to respond like a linear elastic material with strain energy function

$$\Psi_g(\boldsymbol{\varepsilon}; \mathcal{P}_g) = \frac{1}{2} \boldsymbol{\varepsilon}^T \mathbb{C}_g \boldsymbol{\varepsilon} \, . \tag{12.17}$$

  Here the strain energy is given in terms of the microscopic strains $\boldsymbol{\varepsilon}$ and the elastic grain constitutive tensor $\mathbb{C}_g$. The constitutive tensor corresponds to a set of parameters $\mathcal{P}_g$ characterizing the grain properties, namely material constants and orientation of material symmetries. Generally, these properties vary from grain to grain. The microscopic stress tensor follows by the derivation of the considered micro-elasticity energetic response function (12.17) with respect to the strain field

$$\boldsymbol{\sigma} = \frac{\partial\, \Psi_g(\boldsymbol{\varepsilon}; \mathcal{P}_g)}{\partial\, \boldsymbol{\varepsilon}} = \mathbb{C}_g \, \boldsymbol{\varepsilon} \, . \tag{12.18}$$

[1] The loading conditions (12.15) define Dirichlet boundary conditions and thus will yield an upper bound for the consitutive constants. Alternatively, Neumann or periodic boundary conditions can be applied. The Neumann conditions, however, need special considerations since the $\mathcal{RVE}$ is then spatially not fixed and thus can undergo rigid body motions, see e.g. Loehnert and Wriggers (2008).

Within the geometrically linear theory the homogenized tangent response is equivalent to the macroscopic stress-strain relation. It relates the average microscopic stress $\langle \boldsymbol{\sigma} \rangle$ associated to a prescribed deformation, see e.g. (12.15), which is given here for a general prescribed deformation by $\bar{\varepsilon} = \frac{1}{2}(\Xi + \Xi^T)$,

$$\bar{\mathbb{C}} = \frac{\mathrm{D}\bar{\boldsymbol{\sigma}}}{\mathrm{D}\bar{\varepsilon}} = \frac{\mathrm{D}\langle \boldsymbol{\sigma} \rangle}{\mathrm{D}\bar{\varepsilon}} = \left\langle \frac{\mathrm{D}\boldsymbol{\sigma}}{\mathrm{D}\bar{\varepsilon}} \right\rangle , \tag{12.19}$$

with $\bar{\mathbb{C}}$ being the tangent homogenized constitutive tensor of the polycrystalline material. Hereby, $\left\langle \frac{\mathrm{D}\boldsymbol{\sigma}}{\mathrm{D}\bar{\varepsilon}} \right\rangle$ is the volume average of the sensitivity of the microscopic stress with respect to the macroscopic strain. By using the chain rule we can determine the constitutive tensor as

$$\boxed{\bar{\mathbb{C}} = \frac{1}{\Omega} \int_{\Omega} \frac{\partial \boldsymbol{\sigma}}{\partial \varepsilon} \frac{\partial \varepsilon}{\partial \boldsymbol{u}} \frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}\bar{\varepsilon}} \, \mathrm{d}\Omega} \tag{12.20}$$

where now the sensitivity of the displacement field with respect to the macroscopic strain is needed. This however can be computed within the discretization scheme using the nodal degrees of freedom of an element.

12.1.6 Multiscale Modeling Approach (FE^2/VE^2)

Another homogenization concept is based on the multiscale (two-scale) modeling approach (FE^2/VE^2). Hereby, the construction of the multiscale method allows to compute the macroscopic response of a solid without knowing the constitutive equation related to the macroscopic scale. This is achieved by considering the structure and constitutive behaviour at microscale and leads to a two-scale micro-to-macro scenario.

We define the macroscopic stresses $\bar{\boldsymbol{\sigma}}$ by homogenization of the micro stresses $\boldsymbol{\sigma}$, which we compute from the micro scale constitutive equation $\boldsymbol{\sigma} = \mathbb{C}\,\varepsilon$ driven by the macro strains $\bar{\varepsilon}$. Figure 12.5 demonstrates the individual steps of such multiscale method[2] as

- **Localization**: Prescribe the deformation at the boundary Γ of the microstructure by the macro strain $\bar{\varepsilon}$: $\boldsymbol{u}|_{\Gamma} = \bar{\varepsilon}\,\boldsymbol{x}$.
- **Micro model**: Solve the weak form in Ω using the constitutive micro model $\boldsymbol{\sigma} = \mathbb{C}\,\varepsilon$ and compute the micro stresses $\boldsymbol{\sigma}$ in the $\mathcal{RVE}$.
- **Homogenization**: Compute the macro stresses $\bar{\boldsymbol{\sigma}}$ by average of $\boldsymbol{\sigma}$ as: $\bar{\boldsymbol{\sigma}} = \langle \boldsymbol{\sigma} \rangle$.

[2] The methodology allows different discretization schemes such as FEM at micro and FEM at macro scale or VEM at micro and FEM at macro scale etc.

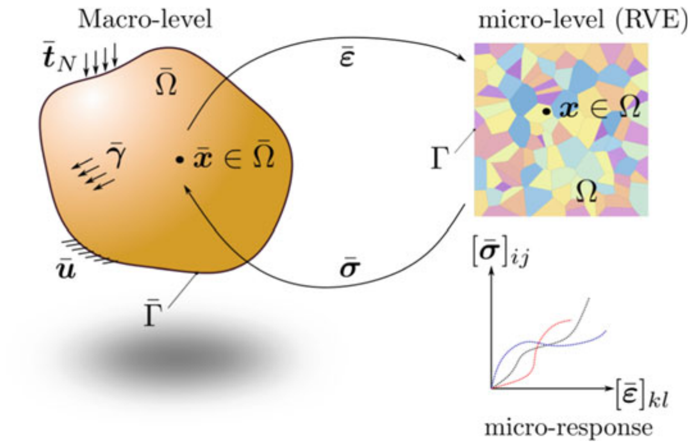


Fig. 12.5 Two-scale bridging scenario: multiscale modeling approach

- **Sensitivity**: To obtain an overall consistent Newton-Rahpson algorithm the sensitivity of the macro stresses with respect to the macro strains has to be computed, see (12.19) and (12.20)

This concept defines a material model on the macroscale by the constitutive response on the microscale defined by the representative volume element. The general two-scale computational framework leads to the following procedure:

1. Evaluate the macroscopic deformation at each material point.
2. Apply in this material point the boundary conditions resulting from the macroscopic deformation at the microscopic $\mathcal{RVE}$.
3. Solve the microscopic boundary value problem under the macroscopic loading from step 2.
4. Transfer the volumetric averaged microscopic quantities to the dedicated material macroscopic points.
5. Solve the macroscopic boundary value problem while retaining by sensitivity analysis the quadratic convergence of the overall solution algorithm Korelc (2009), Zupan and Korelc (2020).

In this way the knowledge of the material behaviour at micro-level is used for the solution of the macroscopic boundary value problem.

12.2 The Virtual Element Method

The virtual element method is very useful for the homogenization of heterogeneous materials due to its ability to treat elements with complex, non-convex shapes and arbitrary number of nodes. This flexibility enables VEM to be employed for materials with crystalline microstructures, as reviewed in e.g. Marino et al. (2019), Artioli et al. (2020b), Cascio et al. (2020), Böhm et al. (2021) and Artioli (2022).

The construction of the virtual element method for linear elasticity is based on the formulations which were developed in Sect. 6.1.1 for two-dimensional problems. Together with the basic formulation of the projection in Chap. 3 and the construction of virtual elements in Chap. 5 the two- and three-dimensional formulation can be summarized.

Based on (6.4) the strains for a virtual element with linear ansatz are given by

$$\boldsymbol{\varepsilon}_\pi = \mathbb{P}^{(n,1)}_{\nabla\varepsilon}\,\mathbf{u}_v \tag{12.21}$$

where $n \in \{2, 3\}$ is the dimension of the problem, $\mathbb{P}^{(n,1)}_{\nabla\varepsilon}$ is the projection operator and $\mathbf{u}_v = \langle \mathbf{u}_1\,\mathbf{u}_2\,\ldots\,\mathbf{u}_{n_V}\rangle^T$ is the vector of the nodal degrees of freedom. This ansatz leads to the consistency part of the internal elastic strain energy for an element Ω_v in the microstructure

$$U^n_c = \frac{\Omega_v}{2}\mathbf{u}_v^T\,[\mathbb{P}^{(n,1)}_{\nabla\varepsilon}]^T\,\mathbb{C}\,\mathbb{P}^{(n,1)}_{\nabla\varepsilon}\,\mathbf{u}_v \tag{12.22}$$

with the constitutive matrix $\mathbb{C}$.

It has been observed in large deformation applications of virtual elements for the analysis of polycrystalline materials that the energy stabilization, see Sect. 6.1.3, leads for virtual elements with linear ansatz to superior performance, see Böhm et al. (2023). This stabilization relies on the same strain energy as the consistency part

$$U^n_s = \sum_{m=1}^{n_{int}} U^n_{s\,m} \quad \text{with} \quad U^n_{s\,m} = \int\limits_{\Omega_m} \beta\,\frac{\Omega_m}{2}\,[\mathbf{u}^m_h]^T\,[\mathbf{B}^m_h]^T\,\mathbb{C}\,\mathbf{B}^m_h\,\mathbf{u}^m_h\,\mathrm{d}\Omega \tag{12.23}$$

where β is the stabilization parameter and Ω_m is an element of the sub-triangularization with n_{int} linear triangles or tetrahedra, depending on the dimension of the problem. As already mentioned in Section 6.1.3, the nodal degrees in (12.22) and (12.23) are the identical since the sub-mesh uses exactly the same nodes as the discretization with virtual elements. The strain within the internal element is given by $\varepsilon^m_h = \mathbf{B}^m_h\,\mathbf{u}^m_h$ with constant strain matrix $\mathbf{B}^m_h$.

This formulation allows the combination of the consistency and stabilization part as, see (6.37),

$$U^n = U^n_c + \beta\,U^n_s = (1-\beta)\,U^n_c(\boldsymbol{u}_\pi) + \beta\,U^n_s(\boldsymbol{u}_h)\,. \tag{12.24}$$

The residual and stiffness matrix of the element Ω_v follow now by diffentiation with respect to the nodal unknown $\mathbf{u}_v$ as

$$\begin{aligned}\mathbf{R}_v^n &= \frac{\partial U^n}{\partial \mathbf{u}_v} = (1-\beta)\frac{\partial U_c^n}{\partial \mathbf{u}_v} + \beta \mathop{\mathsf{A}}_{m=1}^{n_{int}} \frac{\partial U_{s\,m}^n}{\partial \mathbf{u}_m} = (1-\beta)\mathbf{R}_{cv}^n + \beta \mathop{\mathsf{A}}_{m=1}^{n_{int}} \mathbf{R}_m^n \\ \mathbf{K}_v^n &= \frac{\partial \mathbf{R}_v}{\partial \mathbf{u}_v} = (1-\beta)\frac{\partial \mathbf{R}_{cv}^n}{\partial \mathbf{u}_v} + \beta \mathop{\mathsf{A}}_{m=1}^{n_{int}} \frac{\partial \mathbf{R}_m^n}{\partial \mathbf{u}_m}\end{aligned} \tag{12.25}$$

for dimension n. Note that $\mathbf{u}_m \in \mathbf{u}_v$.

12.2.1 Homogenization Procedure: Sensitivity Analysis for Virtual Elements

Sensitivity analysis in computational mechanics requires the computation of derivatives of the response (e.g. stresses, strains, displacements) with respect to arbitrary design parameter. In case of computational homogenization, starting from Eq. (12.20), it is evident that the sensitivity $\frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}\bar{\varepsilon}}$ is related to the unknowns $\boldsymbol{u}$ of the homogenization problem. Thus, the sensitivity parameters are the independent components of $\bar{\varepsilon}$. They can be introduced via the set, see e.g. Šolinc and Korelc (2015),

$$\mathbf{\Phi} = \left\{\bar{\varepsilon}_{11}, \bar{\varepsilon}_{12}, \bar{\varepsilon}_{13}, \bar{\varepsilon}_{21}, \bar{\varepsilon}_{22}, \bar{\varepsilon}_{23}, \bar{\varepsilon}_{31}, \bar{\varepsilon}_{32}, \bar{\varepsilon}_{33}\right\}. \tag{12.26}$$

In what follows, $\mathbf{u}_\pi = \mathbf{u}_v$ and $\mathbf{u}_h^m \in \mathbf{u}_v$ refer to the vector of nodal unknowns for the projection part and for a particular element Ω_m of the sub-mesh for the stabilization part, respectively. The sensitivity problem is solved once the convergence criterion for the primal problem is reached and the current displacements are known. The solution of the primal problem requires

$$\mathbf{R}_v\left(\mathbf{u}_\pi(\mathbf{\Phi}), \mathbf{u}_h^m(\mathbf{\Phi}), \mathbf{u}_D(\mathbf{\Phi})\right) = \mathbf{0}, \tag{12.27}$$

where the stabilization control parameter $\beta\left(\varepsilon\left(\mathbf{u}_\pi(\mathbf{\Phi})\right)\right)$ from (12.24) is a function of the deformation. The function $\mathbf{u}_D(\Phi)$ relates to the micro level nodal unknowns with prescribed essential boundary condition as a function of sensitivity parameters.

The application of the total derivative to the residual in (12.27) leads to

$$\begin{aligned}\frac{\mathrm{D}\mathbf{R}_v}{\mathrm{D}\mathbf{\Phi}} = (1-\beta)\,\frac{\partial \mathbf{R}_{cv}^n}{\partial \mathbf{u}_\pi}\frac{\mathrm{D}\mathbf{u}_\pi}{\mathrm{D}\mathbf{\Phi}} + \mathop{\mathsf{A}}_{m=1}^{n_{int}}\ \beta\frac{\partial \mathbf{R}_m^n}{\partial \mathbf{u}_m}\frac{\mathrm{D}\mathbf{u}_m}{\mathrm{D}\mathbf{\Phi}} \\ + \frac{\partial \mathbf{R}_v}{\partial \mathbf{u}_D}\frac{\mathrm{D}\mathbf{u}_D}{\mathrm{D}\mathbf{\Phi}} + \left(\mathop{\mathsf{A}}_{m=1}^{n_{int}} \mathbf{R}_m^n(\mathbf{u}_m) - \mathbf{R}_{cv}^n(\mathbf{u}_\pi)\right)\frac{\mathrm{D}\beta}{\mathrm{D}\mathbf{\Phi}}\end{aligned} \tag{12.28}$$

where $\frac{\mathrm{D}\mathbf{u}_D}{\mathrm{D}\mathbf{\Phi}}$ denotes the sensitivity of the constrained nodes (i.e. derivatives of boundary conditions with respect to sensitivity parameters) (12.11). In case that the Dirichlet boundary conditions (12.15) are employed the differentiation is simply $\frac{\mathrm{D}\mathbf{u}_D}{\mathrm{D}\mathbf{\Phi}} = \boldsymbol{x}$. The sensitivity of β with respect to $\mathbf{\Phi}$ is obtained by

$$\frac{D\beta}{D\mathbf{\Phi}} = \frac{\partial\beta}{\partial\bar{\varepsilon}}\frac{\partial\bar{\varepsilon}}{\partial\varepsilon}\frac{\partial\varepsilon}{\partial\mathbf{u}_\pi}\frac{D\mathbf{u}_\pi}{D\mathbf{\Phi}} \tag{12.29}$$

Since the internal submesh for stabilization is non-overlapping it yields

$$\Omega_v = \sum_{m=1}^{n_{int}} \Omega_m, \quad \mathbf{u}_v = \mathbf{u}_\pi = \mathop{\mathsf{A}}_{m=1}^{n_{int}} \mathbf{u}_m \quad \Rightarrow \quad \frac{D\mathbf{u}_v}{D\mathbf{\Phi}} = \frac{D\mathbf{u}_\pi}{D\mathbf{\Phi}} = \mathop{\mathsf{A}}_{m=1}^{n_{int}} \frac{D\mathbf{u}_m}{D\mathbf{\Phi}} \tag{12.30}$$

Insertion of (12.29) into (12.28) with application of (12.30) leads after a reordering to a linear system of equations for the solution of the unknown sensitivities

$$\mathbf{K}_v \frac{D\mathbf{u}_v}{D\mathbf{\Phi}} = -\mathbb{R}_v \qquad \text{where} \qquad \mathbb{R}_v = \frac{\partial\mathbf{R}_v}{\partial\mathbf{u}_D}\frac{D\mathbf{u}_D}{D\mathbf{\Phi}} \tag{12.31}$$

the abbreviation $\mathbb{R}_v$ can be interpreted as a load and the element tangent matrix $\mathbf{K}_v$ is defined as

$$\begin{aligned} \mathbf{K}_v &= \left((1-\beta)\frac{\partial\mathbf{R}_v}{\partial\mathbf{u}_v} + \beta\mathop{\mathsf{A}}_{m=1}^{n_{int}}\frac{\partial\mathbf{R}_v}{\partial\mathbf{u}_m}\right) + \Delta\mathbf{R}_v\left(\frac{\partial\beta}{\partial\bar{\varepsilon}}\frac{\partial\bar{\varepsilon}}{\partial\varepsilon}\right)\frac{\partial\varepsilon}{\partial\mathbf{u}_v} \\ \text{with } \Delta\mathbf{R}_v &= \mathop{\mathsf{A}}_{m=1}^{n_{int}}\mathbf{R}_v(\mathbf{u}_m) - \mathbf{R}_v(\mathbf{u}_\pi)\,. \end{aligned} \tag{12.32}$$

This presented approach to computational homogenization via a sensitivity analysis returns the algorithmic consistent linearized element tangent contribution of the micro-problem with regard to a macroscopic material point. For more details on sensitivity analysis based multi-scale methods, the interested reader is referred to Korelc (2009), Korelc and Wriggers (2016), Zupan and Korelc (2020).

In summary, the sensitivity analysis based computational homogenization relies on the steps:

1. Apply boundary conditions representing specific macroscopic deformations $\boldsymbol{u}|_\Gamma = \bar{\varepsilon}\,\boldsymbol{x}$ at the microscopic $\mathcal{RVE}$ where $\bar{\varepsilon}$ is the prescribed macroscopic strain.
2. Solve the microscopic boundary value problem under the macroscopic loading of step 1.
3. Compute the averaged (macroscopic) stresses and strains using (12.9).
4. Determine the homogenized constitutive tensor from the discretized form of (12.20)

$$\boxed{\bar{\mathbb{C}}_h = \frac{1}{\Omega}\mathop{\mathsf{A}}_{v=1}^{n_v}\int_{\Omega_v}\left.\frac{\partial\sigma}{\partial\varepsilon}\right|_v\frac{\partial\varepsilon}{\partial\boldsymbol{u}_v}\frac{D\boldsymbol{u}_v}{D\bar{\varepsilon}}d\Omega} \tag{12.33}$$

which accounts for the contributions of all virtual elements of the microstructure. The sensitivity $\frac{D\boldsymbol{u}_v}{D\bar{\varepsilon}} = \frac{D\boldsymbol{u}_v}{D\mathbf{\Phi}}$ follows from (12.31).

The sensitivity approach will be used in the remaining part of this chapter to compute the effective material constants.

Table 12.1 Material properties of three grains employed in the numerical examples (data taken from Marino et al. (2019))

Grain material	Anisotropy	$\mathbb{C}'_{g\,11}$	$\mathbb{C}'_{g\,12}$	$\mathbb{C}'_{g\,22}$	$\mathbb{C}'_{g\,23}$	$\mathbb{C}'_{g\,44}$	$\mathbb{C}'_{g\,66}$	U^a_g
Iron Selenide (FeSe)	HA	10	3	91	31	62	1	36.19
Lithium Titanate ($LiTiO_2$)	MA	252	34	289	105	92	7	16.65
Tricalcium Silicate (Ca_3SiO_5)	QI	133	53	191	75	58	33	0.46

12.3 Representative Numerical Examples

The section presents the performance of the virtual element method when applied to computational homogenization of polycrystalline materials. Different numerical studies are investigated in two- and three-dimensional settings. These include the stress response of two-dimensional microstructures for uniaxial and shear deformations as well as three-dimensional applications. In the latter the focus is put on the computational error of macroscopic effective properties, furthermore the influence of VEM stabilization on the quality of the results is discussed.

We denote by VE-VO a virtual element discretization with Voronoi cells and energy-stabilization. By FE-O1/O2 we label results, computed with a finite element approach using linear/quadratic tetrahedral 3D elements (in 2D: linear/quadratic triangles FE-T1/T2). Furthermore, the addition -minimal (i.e. O1-minimal) describes a finite element discretization with a minimal number of elements which is provided by the sub-mesh of the energy stabilization used for a VEM analysis. The sub-mesh has the same number of DOFs as the VEM discretization.

The microstructures are generated by Voronoi tessellation with a random distribution of seed points. This yields polycrystalline assemblies which do not describe real microstructures. However for our numerical test this artificial generation of polycrystalline microstructures is sufficient. Note that only one virtual element will be used per polycrystal.

In the two-dimensional case, geometries and meshes are created by *Mathematica* using the Mathematica function `VoronoiMesh`. In three dimensions, the open-source code *Neper* has been used to generate Poisson Voronoi tessellations, see Quey et al. (2011).

The simulations are performed with the material data set from Table 12.1. This table lists material constants of single grains, i.e. components of the constitutive matrix $\mathbb{C}_g$ at the micro-scale. The material constants represent components of the constitutive matrix $\mathbb{C}'_g$ expressed in a local coordinate system $(\boldsymbol{e}'_1, \boldsymbol{e}'_2, \boldsymbol{e}'_3)$. Material properties $\mathbb{C}_g$ introduced in (12.17) with the global coordinate system $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$

are obtained from $\mathbb{C}'_g$ through classical transformation matrices $\mathbf{Q}_g$ as

$$\boldsymbol{e}_i = \mathbf{Q}_g\ \boldsymbol{e}'_i \quad \text{with} \quad i = \{1, 2, 3\}\,. \tag{12.34}$$

The transformation matrices are defined on the basis of the three angles $\{\theta_g, \delta_g, \phi_g\}$ between material preferred direction and coordinate axes which yields

$$\mathbf{Q}_g = \begin{bmatrix} \cos\theta_g & \sin\theta_g & 0 \\ -\sin\theta_g & \cos\theta_g & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\delta_g & \sin\delta_g \\ 0 & -\sin\delta_g & \cos\delta_g \end{bmatrix} \begin{bmatrix} \cos\phi_g & \sin\phi_g & 0 \\ -\sin\phi_g & \cos\phi_g & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{12.35}$$

as outlined in Kollar and Springer (2003).

The universal anisotropy index U_g^a In Table 12.1 is a defined in Ranganathan and Ostoja-Starzewski (2008) as

$$U_g^a = 5\frac{\mu^V}{\mu^R} + \frac{\kappa^V}{\kappa^R} - 6 \tag{12.36}$$

where $\mu^{R/V}$ are the estimates of the effective shear moduli when using the Reuss and Voigt bounds, respectively. in the same way, $\kappa^{R/V}$ represents the Reuss and Voigt estimates of the modulus of compression. For isotropic behaviour we obtain $U_g^a = 0$ while the greater the value of U_g^a the greater the anisotropy.

In most of the numerical simulations, the stabilization parameter for VEM is set to $\beta = 0.1$. This is based on numerical analysis of different values of β, see end of Sect. 12.3.2.

12.3.1 Tensile and Shear Deformations in Two Dimensions

For different two-dimensional deformation states, the stress distributions obtained with different computational approaches are depicted in Figs. 12.6 and 12.7. The reference configurations (first row) and the deformed configuration (second row) are plotted for two different first-order discretization schemes (FEM and VEM) and are compared with benchmark results (using an overkill solution with 111788 DOFs). Hereby, two loading cases are considered:

- Uniaxial deformation mode with given strain $\bar{\varepsilon} = \{\bar{\varepsilon}_{xx}\ \bar{\varepsilon}_{yy}\ \bar{\varepsilon}_{xy}\}^T = \{0.25\ 0\ 0\}^T$, see Fig. 12.6. The results are obtained with VE-VO and FE-T1-minimal discretizations and compared with the benchmark solution.
- Shear deformation mode with given strain $\bar{\varepsilon} = \{\bar{\varepsilon}_{xx}\ \bar{\varepsilon}_{yy}\ \bar{\varepsilon}_{xy}\}^T = \{0\ 0\ 0.25\}^T$, see Fig. 12.7, using the same discretization as in the uniaxial case.

In both figures, the color in each VE-VO or FE-T1 element is associated with the constant stress which is consistent with the low-order element formulation. For comparison purpose, the stress distribution obtained in the benchmark is also reported.

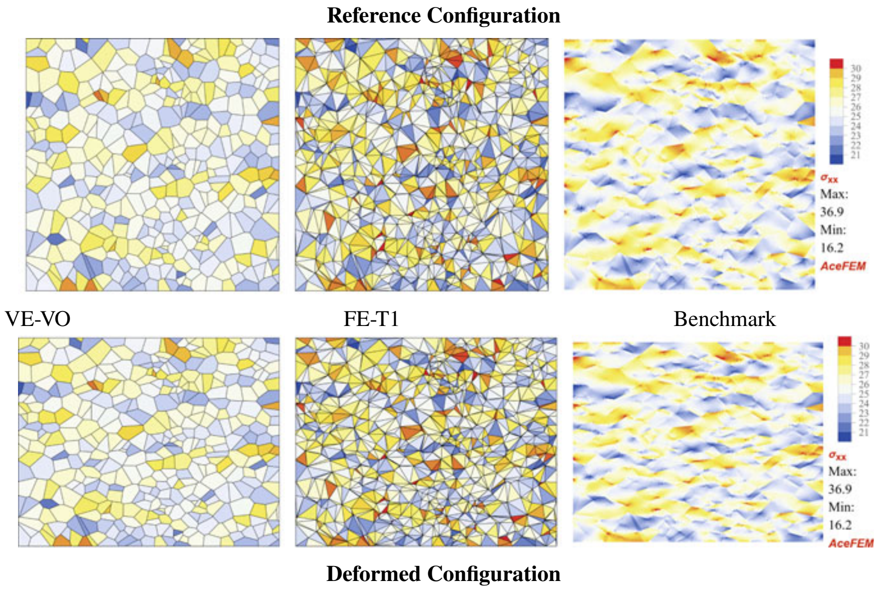


Fig. 12.6 Microscopic stress distribution σ_{xx} for different discretizations and tensile deformation mode

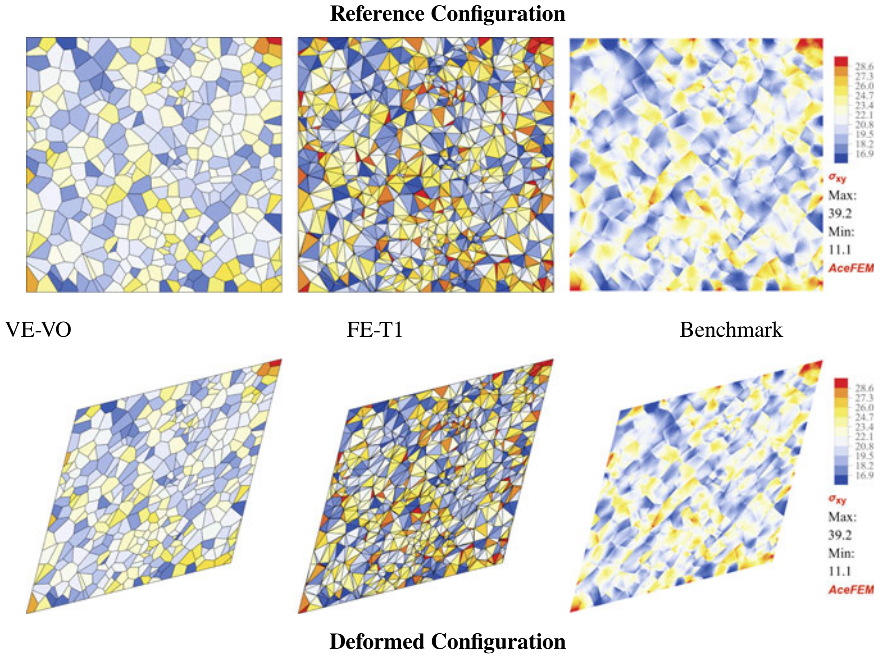
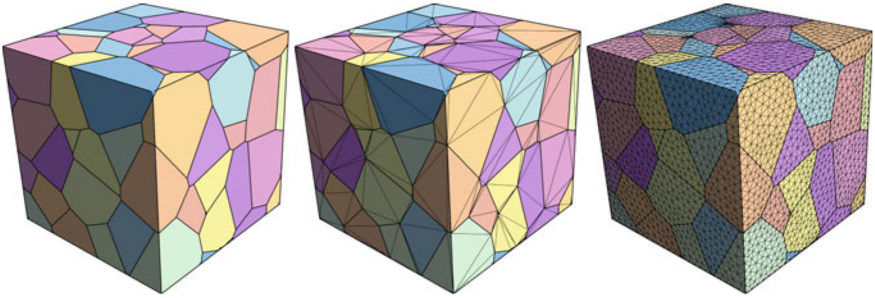


Fig. 12.7 Microscopic stress distribution σ_{xy} for different discretizations and shear deformation mode



(a) Virtual element mesh (b) Finite element sub mesh (c) Finite element mesh

Fig. 12.8 Artificially generated RVE with 200 grains

For FE-T1-minimal discretization, the values obtained in neighboring elements are highly scattered and yield stress concentrations that are not visible in the benchmark solution. Thus, the stresses are unrealistic when compared with the benchmark case. Specifically, there exists some highly distorted elements that are characterized by high stresses. However those are located in regions distinguished by moderate stresses in the benchmark distribution. On the other hand, the stress distribution from the virtual element simulation shows results close to the benchmark solution. These results prove the efficient performance of VEM, since the stress values correspond to the real stresses which are the quantities of interest in the homogenization process.

In summary, the response of virtual element formulations in terms of average stress distributions appears to be less sensitive than finite element schemes when applied to distorted meshes with strong and heterogeneous grain anisotropies. Further comparisons of the virtual element method with higher order finite element methods can be found in Marino et al. (2019).

12.3.2 Three Dimensional Homogenization

A microstructure consisting of 200 polyhedral grains is considered see Fig. 12.8a. It is generated artificially by a Voronoi tessellation. Virtual and finite element approaches will be compared in a parametric study. In the virtual element approach, each polyhedral grain is discretized by one virtual element (VE-VO) as shown in Fig. 12.8a. By employing a sub triangulation, a coarse FE-O1-minimal mesh with the same number of nodes as the VE-VO mesh is constructed, see Fig. 12.8b. This corresponds to the sub-mesh with which the stabilization term is computed for the energy stabilization. The results of these two meshes are compared with a fine mesh, see Fig. 12.8c, which resolves each of the grains by an inscribed mesh. Here two discretizations are introduced based on linear (FE-O1) and quadratic (FE-O2) finite elements.

Different material properties of the grains are introduced and employed in the 3D parametric study. These are iron selenide (FeSe), lithium titanate ($LiTiO_2$) and tricalcium silicate (Ca_3SiO_5). Iron selenide is known to be highly anisotropic (HA) while lithium titanate is mildly anisotropic (ML) and tricalcium silicate is quasi isotropic (QI). This can be seen in Table 12.1 by looking at the components of the constitutive tensor at grain level $\mathbb{C}'_g$ (in GPa) being expressed in the local coordinate system $(\boldsymbol{e}'_1, \boldsymbol{e}'_2, \boldsymbol{e}'_3)$ that is aligned with material preferred directions. Furthermore the universal anisotropy index U^a_g is used to highlight the degree of anisotropy.

Computational error of macroscopic effective properties. When applied to homogenization, the performance of virtual and finite element approaches can be evaluated by an estimator of the computational error $\mathcal{E}_C$. It is designed to describe the deviation of the computed effective constitutive data from a reference solution with a very fine mesh

$$\mathcal{E}_C = \frac{||\bar{\mathbb{C}}^{\mathfrak{D}} - \bar{\mathbb{C}}^{ref}||}{||\bar{\mathbb{C}}^{ref}||} \cdot 100\% \quad \text{with } \mathfrak{D} \in \left\{\text{VE-VO, FE-O1, FE-O2}\right\}. \tag{12.37}$$

Three different cases of anisotropy (highly/mildly anisotropic cases and quasi-isotropic case) are investigated, with data from Table 12.1. These are solved for the different discretizations. $\bar{\mathbb{C}}^{ref}$ is the effective macroscopic constitutive matrix resulting from the solution of the FE-O2 discretization using a fine mesh with $1.4 \cdot 10^6$ DOFs. For VE-VO, we fix the stabilization constant to $\beta = 0.1$.

Table 12.2 presents the computational error estimation $\mathcal{E}_C$ for the effective macroscopic constitutive matrix $\bar{\mathbb{C}}$ related to different discretization schemes and three degrees of anisotropy (highly anisotropic case with trigonal unit cell, mildly anisotropic with hexagonal unit, and quasi isotropic with orthorhombic unit cell). For VE-VO and FE-O1-minimal the same number of DOF are considered ($2.5k$ DOFs) whereas more elements in FEM-O1 fine ($161k$ DOFs) and FEM-O2 ($22k$ DOFs) are used. A good performance of the virtual element method is observed with the smallest error $\mathcal{E}_C$ for all cases when compared with linear and higher order finite element methods. By increasing the degree of anisotropy as well as by varying material symmetry classes, VE-VO demonstrates robust and accurate results with a small error $\mathcal{E}_C$, as illustrated in Table 12.2.

The computational homogenization based on VE-VO/FE-O1/FE-O2 of a RVE yields the following effective macroscopic modulus for highly anisotropic grains:

Table 12.2 Computational error $\mathcal{E}_C$ estimation (in percent) for different discretization schemes and degrees of anisotropy

Discretization	VE-VO	FE-O1	FE-O1(fine)	FE-O2
Number of equations	$2.5k$ DOFs	$2.5k$ DOFs	$161k$ DOFs	$22k$ DOFs
Highly anisotropic	6.32	40.14	9.13	14.27
Mildly anisotropic	1.58	13.09	2.85	4.40
Quasi isotropic	0.07	1.31	0.20	0.33

- VE-VO with $\approx 2.5k$ DOF, $\beta = 0.1$ and 200 Grains:

$$\bar{\mathbb{C}} = \begin{bmatrix} 31.655 & 10.700 & 11.833 & -0.31151 & -1.3472 & 0.23269 \\ 10.700 & 34.126 & 10.952 & 0.67085 & 0.39950 & -0.84181 \\ 11.833 & 10.952 & 44.690 & -0.09844 & -0.95517 & -0.39478 \\ -0.31151 & 0.67085 & -0.09844 & 13.445 & -0.37312 & 0.12591 \\ -1.3472 & 0.3995 & -0.95514 & -0.37312 & 14.442 & 0.17552 \\ 0.23269 & -0.84181 & -0.39478 & 0.12591 & 0.17552 & 10.053 \end{bmatrix}$$

- FE-O1 with $\approx 161k$ DOF and 200 Grains

$$\bar{\mathbb{C}} = \begin{bmatrix} 32.829 & 10.345 & 11.713 & -0.27974 & -1.3182 & 0.30427 \\ 10.345 & 35.290 & 10.891 & 0.78866 & 0.24098 & -0.80396 \\ 11.713 & 10.891 & 45.655 & -0.04606 & -0.99944 & -0.37193 \\ -0.27974 & 0.78866 & -0.04606 & 14.128 & -0.22852 & -0.06552 \\ -1.3182 & 0.24098 & -0.99944 & -0.22851 & 15.126 & 0.11165 \\ 0.30427 & -0.80396 & -0.37193 & -0.06552 & 0.11165 & 10.480 \end{bmatrix}$$

- Reference: FE-O2 with $\approx 1.4M$ DOF and 200 Grains

$$\bar{\mathbb{C}} = \begin{bmatrix} 29.967 & 9.8164 & 11.155 & -0.27974 & -1.3182 & 0.30427 \\ 9.8164 & 32.156 & 10.425 & 0.78866 & 0.24098 & -0.80396 \\ 11.155 & 10.425 & 41.888 & -0.04606 & -0.99944 & -0.37193 \\ -0.27974 & 0.78866 & -0.04606 & 12.540 & -0.22852 & -0.06552 \\ -1.3182 & 0.24098 & -0.99944 & -0.22852 & 13.657 & 0.11165 \\ 0.30427 & -0.80396 & -0.37193 & -0.06552 & 0.11165 & 9.3153 \end{bmatrix}$$

By comparing the obtained values it is clear that the results of VE-VO are closer to the reference solution. This comes with a remarkable reduction of computational costs of two orders of magnitude as demonstrated in Table 12.3.

Table 12.3 Comparison of the run times of different discretization schemes

Method	No. of DOF	Total time (s)
VE-VO (β=0.1)	$\approx 2.5 \cdot 10^3$	1.127
FE-O1 (fine)	$\approx 161 \cdot 10^3$	109.85
FE-O2 (fine, reference)	$\approx 1.4 \cdot 10^6$	7967.7

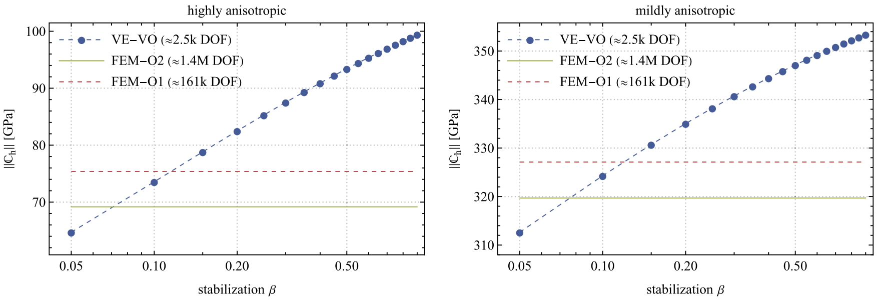


Fig. 12.9 Investigations of Frobenius norm $||\mathbb{C}_h||$ of the homogenized stiffness matrix using VEM for different grain properties

In conclusion, the virtual element method produces with one element per grain for practical purposes very good and accurate results for homogenization of anisotropic materials.

Influence of the stabilization parameter. The computational error $\mathcal{E}_C$ was obtained in the above study by employing a fixed value of the stabilization parameter $\beta = 0.1$ in VE-VO approach. The effect of this stabilization parameter on the final results will be investigated next.

The solutions regarding the norm of the effective macroscopic constitutive tensor $\mathbb{C}_h$, illustrated in Fig. 12.9, depict that VE-VO matches the finest solution for low values of the stabilization parameter $\beta \in [0.075, 0.15]$ in the mild and high anisotropy cases. The low optimal value of β suggests that the computational homogenization application requires only a small stabilization of the virtual element. Thus the stabilization term is only needed to eliminate the rank-deficiency of virtual elements, as outlined in Marino et al. (2019), Böhm et al. (2021), Böhm et al. (2023).

On one hand it is unfortunate that the results depend on the stabilization parameter. But on the other hand, all our experiences using virtual elements for homogenization indicate that one can use $\beta = 0.1$ for a large range of homogenization problems including isotropy and anisotropic elasticity but also crystal plasticity, see Böhm et al. (2023).

References

Aldakheel, F., N. Noii, T. Wick, and P. Wriggers. 2021. A global-local approach for hydraulic phase-field fracture in poroelastic media. *Computers and Mathematics with Applications* 91: 99–121.

Artioli, E., L. Beirão da Veiga, and M. Verani. 2020. An adaptive curved virtual element method for the statistical homogenization of random fibre-reinforced composites. *Finite Elements in Analysis and Design* 177: 103418.

Artioli, E. 2022. VEM approach for homogenization of fibre-reinforced composites with curvilinear inclusions. In *Current trends and open problems in computational mechanics*, 31–39. Berlin: Springer.

Behrens, B.-A., H.J. Maier, G. Poll, P. Wriggers, F. Aldakheel, C. Klose, F. Nuernberger, F. Pape, C. Böhm, A. Chugreeva, D. Duran, S.E. Thuerer, S. Herbst, J.I. Hwang, T. Matthias, N. Heimes, and J. Uhe. 2020. Numerical investigations regarding a novel process chain for the production of a hybrid bearing bushing. *Production Engineering* 14: 569–581.

Bensoussan, A., J.L. Lions, and G. Papanicolaou. (1978. Asymptotic analysis for periodic structures. In *Studies in mathematics and its applications*, vol. 5, ed. J.L. Lions, G. Papanicolaou, and R.T. Rockafellar. North-Holland Publishing Company.

Böhm, C., B. Hudobivnik, M. Marino, and P. Wriggers. 2021. Electro-magneto-mechanically response of polycrystalline materials: Computational homogenization via the virtual element method. *Computer Methods in Applied Mechanics and Engineering* 375: 113775.

Böhm, C., L. Munk, B. Hudobivnik, F. Aldakheel, J. Korelc, and P. Wriggers. 2023. Virtual elements for computational anisotropic crystal plasticity. *Computer Methods in Applied Mechanics and Engineering* 405: 115835.

Cascio, M.L., A. Milazzo, and I. Benedetti. 2020. Virtual element method for computational homogenization of composite and heterogeneous materials. *Composite Structures* 232: 111523.

Chatzigeorgiou, G., A. Javili, and P. Steinmann. 2014. Unified magnetomechanical homogenization framework with application to magnetorheological elastomers. *Mathematics and Mechanics of Solids* 19: 193–211.

Feyel, F., and J.L. Chaboche. 2000. FE^2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials. *Computer Methods in Applied Mechanics and Engineering* 183: 309–330.

Fish, J. 2014. *Practical multiscaling*. United Kingdom: Wiley.

Fish, J., and A. Wagiman. 1993. Multiscale finite element method for a locally nonperiodic heterogeneous medium. *Computational Mechanics* 12 (3): 164–180.

Geers, M.G.D., V.G. Kouznetsova, and W.A.M. Brekelmans. 2010. Multi-scale computational homogenization: Trends and challenges. *Journal of Computational and Applied Mathematics* 234: 2175–2182.

Hain, M., and P. Wriggers. 2008. On the numerical homogenization of hardened cement paste. *Computational Mechanics* 42: 197–212.

Hain, M., and P. Wriggers. 2008. Computational homogenization of micro-structural damage due to frost in hardened cement paste. *Finite Elements in Analysis and Design* 44: 223–244.

Hashin, Z., and S. Shtrikman. 1962. On some variational principles in anisotropic and nonhomogeneous elasticity. *Journal of the Mechanics and Physics of Solids* 10: 335–342.

Hashin, Z., and S. Shtrikman. 1963. A variational approach to the theory of the elastic behaviour of multiphase materials. *Journal of the Mechanics and Physics of Solids* 11: 127–140.

Heinze, S., M. Joulaian, and A. Düster. 2015. Numerical homogenization of hybrid metal foams using the finite cell method. *Computers & Mathematics with Applications* 70 (7): 1501–1517.

Hill, R. 1972. On constitutive macro-variables for heterogeneous solids at finite strain. *Proceedings of the Royal Society (Series A), London* 326: 131–147.

Hill, R. 1965. A self-consistent mechanics of composite materials. *Journal of the Mechanics and Physics of Solids* 13: 213–222.

Javili, A., G. Chatzigeorgiou, and P. Steinmann. 2013. Computational homogenization in magneto-mechanics. *International Journal of Solids and Structures* 50 (25–26): 4197–4216.

Kollar, L.P. and G.S. Springer. 2003. *Mechanics of composite structures*. Cambridge University Press.

Korelc, J. 2009. Automation of primal and sensitivity analysis of transient coupled problems. *Computational mechanics* 44: 631–649.

Korelc, J., and P. Wriggers. 2016. *Automation of finite element methods*. Berlin: Springer.

Kouznetsova, V., M.G.D. Geers, and W.A.M. Brekelmans. 2002. Multi-scale constitutive modelling of heterogeneous materials with a gradient-enhanced computational homogenization scheme. *International Journal for Numerical Methods in Engineering* 54: 1235–1260.

Lloberas-Valls, O., D.J. Rixen, A. Simone, and L.J. Sluys. 2012. Multiscale domain decomposition analysis of quasi-brittle heterogeneous materials. *International Journal for Numerical Methods in Engineering* 83: 1337–1366.

Loehnert, S., and P. Wriggers. 2008. Effective behaviour of elastic heterogeneous thin structures at finite deformations. *Computational Mechanics* 41: 595–606.

Marino, M., B. Hudobivnik, and P. Wriggers. 2019. Computational homogenization of polycrystalline materials with the virtual element method. *Computer Methods in Applied Mechanics and Engineering* 355: 349–372.

Michel, J., H. Moulinec, and P. Suquet. 1999. Effective properties of composite materials with periodic microstructure: A computational approach. *Computer Methods in Applied Mechanics and Engineering* 172 (1): 109–143.

Miehe, C. 2002. Strain-driven homogenization of inelastic microstructures and composites based on an incremental variational formulation. *International Journal for Numerical Methods in Engineering* 55: 1285–1322.

Miehe, C., J. Schröder, and J. Schotte. 1999. Computational homogenization analysis in finite plasticity simulation of texture development in polycrystalline materials. *Computer Methods in Applied Mechanics and Engineering* 171: 387–418.

Miehe, C., J. Schotte, and J. Schröder. 1999. Computational micro-macro transitions and overall moduli in the analysis of polycrystals at large strains. *Computational Materials Science* 16: 372–382.

Nemat-Nasser, S. and M. Hori. 1999. *Micromechanics: overall properties of heterogeneous materials*, vol. 36, 2nd edn. *North-Holland series in applied mathematics and mechanics*. Elsevier Science Publisher B. V.

Ponte, Castañeda P. 1991. The effective mechanical properties of nonlinear isotropic composites. *Journal of the Mechanics and Physics of Solids* 39: 45–71.

Ponte, Castañeda P. 1992. New variational principles in plasticity and their application to composite materials. *Journal of the Mechanics and Physics of Solids* 40: 1757–1788.

Ponte, Castañeda P., and P. Suquet. 1997. Nonlinear composites. *Advances in Applied Mechanics* 34: 171–302.

Quey, R., P. Dawson, and F. Barbe. 2011. Large-scale 3D random polycrystals for the finite element method: Generation, meshing and remeshing. *Computer Methods in Applied Mechanics and Engineering* 200 (17–20): 1729–1745.

Ranganathan, S.I., and M. Ostoja-Starzewski. 2008. Universal elastic anisotropy index. *Physical Review Letters* 101 (5): 055504.

Reuss, A. 1929. Berechnung der Fließgrenze von Mischkristallen aufgrund der Plastizitätsbedingung für Einkristalle. *Zeitschrift für angewandte Mathematik und Mechanik* 9: 49–58.

Sanchez-Palencia, E. 1980. *Non-homogeneous media and vibration theory*, vol. 127, *Lecture notes in physics*. Berlin: Springer.

Schröder, J., and M.A. Keip. 2012. Two-scale homogenization of electromechanically coupled boundary value problems. *Computational Mechanics* 50 (2): 229–244.

Segurado, J., and J. Llorca. 2002. A numerical approximation to the elastic properties of sphere-reinforced composites. *Journal of the Mechanics and Physics of Solids* 50 (10): 2107–2121.

Smit, R.J.M., W.A.M. Brekelmans, and H.E.H. Meijer. 1998. Prediction of the mechanical behavior of nonlinear heterogeneous systems by multi-level element modeling. *Computer Methods in Applied Mechanics and Engineering* 155: 181–192.

Šolinc, U., and J. Korelc. 2015. A simple way to improved formulation of FE^2 analysis. *Computational mechanics* 56 (5): 905–915.

Suquet, P.M. 1987. Elements of homogenization for inelastic solid mechanics. In *Lecture notes in physics: homogenization techniques for composite materials*, ed. E. Sanchez-Palenzia, and A. Zaoui, 193–278. Berlin: Springer.

Suquet, P.M. 1993. Overall potential and extremal surfaces of power law or ideal plastic materials. *Journal of the Mechanics and Physics of Solids* 41: 981–1002.

Talbot, D.R.S., and J.R. Willis. 1992. Some explicit bounds for the overall behaviour of nonlinear composites. *International Journal of Solids and Structures* 29: 1981–1987.

Temizer, İ, and P. Wriggers. 2007. An adaptive method for homogenization in orthotropic nonlinear elasticity. *Computer Methods in Applied Mechanics and Engineering* 196: 3409–3423.

Temizer, İ, and P. Wriggers. 2008. On a mass conservation criterion in micro-to-macro transitions. *Journal of Applied Mechanics - Transaction of the ASME* 75: 054503.

Temizer, İ, and P. Wriggers. 2011. Homogenization in finite thermoelasticity. *Journal of the Mechanics and Physics of Solids* 59: 344–372.

Temizer, İ, and T.I. Zohdi. 2007. A numerical method for homogenization in non-linear elasticity. *Computational Mechanics* 40: 281–298.

Terada, K., and T. Kikuchi. 2001. A class of general algorithms for multi-scale analyses of heterogeneous media. *Computer Methods in Applied Mechanics and Engineering* 190: 5427–5464.

Terada, K., M. Hori, T. Kyoya, and N. Kikuchi. 2000. Simulation of the multi-scale convergence in computational homogenization approaches. *International Journal of Solids and Structures* 37 (16): 2285–2311.

Voigt, W. 1887. Über die Beziehung zwischen den beiden Elastizitätskonstanten isotroper Körper. *Annalen der Physik* 38: 573–587.

Wellmann, C., C. Lillie, and P. Wriggers. 2008. Homogenization of granular material modelled by a three-dimensional discrete element method. *Computers and Geotechnics* 35: 394–405.

Wriggers, P., and S.O. Moftah. 2006. Mesoscale models for concrete: Homogenization and damage behaviour. *Finite Element Analysis and Design* 42: 623–636.

Zohdi, T.I. and P. Wriggers. 2005. *Introduction to Computational Micromechanics*, vol. 20, *LNACM*, ed. F. Pfeiffer, and P. Wriggers. Berlin: Springer.

Zohdi, T.I. 2008. On the computation of the coupled thermo-electromagnetic response of continua with particulate microstructure. *International Journal for Numerical Methods in Engineering* 76 (8): 1250–1279.

Zohdi, T., and P. Wriggers. 1999. Microstructural decomposition error estimates. *Zeitschrift für angewandte Mathematik und Mechanik* 79: 155–158.

Zohdi, T.I., and P. Wriggers. 2001. Computational micro-macro material testing. *Archives of Computational Methods in Engineering* 8: 131–228.

Zohdi, T., and P. Wriggers. 2001. Aspects of the computational testing of the mechanical properties of microheterogeneous material samples. *International Journal for Numerical Methods in Engineering* 50: 2573–2599.

Zohdi, T.I., J.T. Oden, and G.J. Rodin. 1996. Hierarchical modeling of heterogeneous bodies. *Computer Methods in Applied Mechanics and Engineering* 138 (1–4): 273–298.

Zupan, N., and J. Korelc. 2020. Sensitivity analysis based multi-scale methods of coupled path-dependent problems. *Computational Mechanics* 65 (1): 229–248.

Chapter 13
Virtual Elements for Beams and Plates

Beams and plates serve as individual structures or structural members of many technical constructions such as buildings, airplanes, cars, or ships. Thus, many analytical and numerical simulation schemes were developed over the last century that can predict kinematical quantities, like deflections and rotations, and stress resultants, like normal forces, shear forces and bending moments. Classical beam models are described in any textbook on engineering mechanics, plate models and related analytical solutions can be found as well in many textbooks, for a broad range of analytical solutions see e.g. Timoshenko and Woinowsky-Krieger (1959). The underlying mathematical models are based on kinematical assumptions which relate to the names of Euler and Bernoulli for beams and Kirchhoff and Love for plates. They lead to fourth order differential and partial differential equations. Numerical solutions schemes were developed first within the finite difference method. About 60 years ago finite beam and plate elements were developed starting with the work of Melosh (1961), Clough and Tocher (1965), Bazeley et al. (1965) for plates.

The Euler Bernoulli theory requires for beams C^1-continuous ansatz functions which, for a finite element formulation, is easily available using Hermite polynomials, see e.g. Oñate (2013). Thus a virtual element formulation for beams does not necessarily provide any advantages. However, based on the virtual element method beam elements of any ansatz order can be formulated easily, as will be shown in Sect. 13.1.

The usual forms of isoparametric finite element ansatz functions are not able to fulfill C^1-continuity needed to discretize problems when using the Kirchhoff-Love plate theory. Approximations lead to non-conforming elements. First elements that met C^1-continuity were developed in Withum (1966) and then within the TUBA series of triangular elements in Argyris et al. (1968) and by Bell (1969) using a different mathematical formulation. Quadrilateral elements were designed by De Veubeke (1968) based on a composite approach that included four triangular elements. Strictly rectangular elements can be found e.g. in Bufler and Stein (1970) which however have a reduced application range, being not able to discretize general shapes of plates. Even if C^1-continuity is fulfilled, like within the TUBA element family, there is still

P. Wriggers et al., *Virtual Element Methods in Engineering Sciences*,
https://doi.org/10.1007/978-3-031-39255-9_13

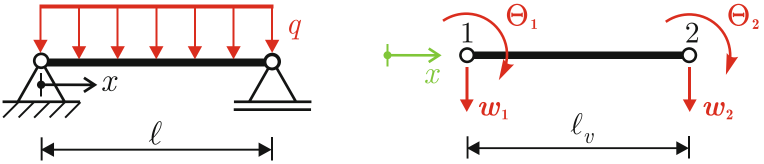
a disadvantage since higher order kinematical quantities, e.g. components of the curvature, $w_{,xx}$, $w_{,yy}$, appear as nodal unknowns, besides the deflection w and rotations $w_{,x}$, $w_{,y}$. It is not obvious and intuitive how to formulate Dirichlet boundary conditions for these variables which renders the application of such elements difficult in an engineering environment.

The latter complication was circumvented by the introduction of Reissner-Mindlin elements that only have to fulfil C^0-continuity and thus allow the use of ansatz functions well known in solid mechanics. The development of related finite elements started with the work of e.g. Zienkiewicz et al. (1971), Hughes et al. (1977). Due to locking of these elements in the thin plate limit, numerous different variants were considered in the following years to overcome this problem. Locking is related to the vanishing shear strains $\gamma_\alpha \to 0$ for plate thickness $h \to 0$, see e.g. the textbooks Hughes (2012), Reddy (1999), Oñate (2013). One of the numerous variants to overcome locking is based on the discrete, point-wise fulfillment of C^1-continuity that led to the discrete Kirchhoff triangles (DKT), see Batoz et al. (1980), and quadrilaterals (DKQ), see Batoz and Tahar (1982).

In the virtual element method the ansatz is defined at the edges of the elements which allows to fulfill C^1-continuity in a relatively easy way. Thus, Kirchhoff-Love plate elements can be developed without any restrictions, see Brezzi and Marini (2013). As shown in Sect. 13.2, VEM has the advantage that no higher order kinematical quantities have to be introduced as nodal unknowns. Construction of virtual plate elements with e.g. three- and four nodes allows to revisit the construction of Kirchhoff-Love elements and their integration into classical software packages. Numerical results for two virtual plate elements with low order ansatz spaces can be found in Chinosi and Marini (2016) which includes error estimates. Its application for a range of engineering problems is described in Wriggers et al. (2021) which has a section where the virtual element technology is employed to develop triangular and quadrilateral finite element like plate elements. Other researcher have so far developed virtual element schemes for dynamic plate problems, see Mora et al. (2018), and buckling analysis of plates, see Mora and Velásquez (2020), Meng and Mei (2020). Furthermore, virtual plate elements using Reissner-Mindlin kinematics were developed in Beirão da Veiga et al. (2019) while non-conforming discretisations were considered in Antonietti et al. (2018) to model plate problems with VEM.

13.1 Virtual Element Formulations for Euler-Bernoulli Beams

As an introduction of C^1-continuous virtual elements we consider a beam of length l with a stiffness of EI where E is the Young's modulus and I the moment of inertia of the cross section. The beam is loaded by a line load of magnitude q, see Fig. 13.1a.



(a) Beam under constant line load (b) Element degrees of freedom

Fig. 13.1 Beam problem and virtual element

The fourth order ordinary differential of the Euler-Bernoulli beam with a constant cross section is given for the deflection w by

$$E\,I\,w''''(x) = q(x)\,. \tag{13.1}$$

By introducing a potential form the differential equation (13.1) can be written as a minimization problem

$$\frac{1}{2}\int_0^l EI[w''(x)]^2\mathrm{d}x - \int_0^l q(x)\,w(x)\mathrm{d}x \longrightarrow MIN\,. \tag{13.2}$$

When using the virtual element formulation, the approximation of the deflection w_h is not defined along the beam axis. The unknown function w_h is only known at the nodal points of the element, as has been already discussed for trusses in Sect. 1.2.1. For a fourth order differential equation not only the nodal deflection w_i but also its derivative $w_{,x\,i}$ (rotations) and, depending on the ansatz order, some moments m_k have to be introduced as unknowns.

13.1.1 Third Order Ansatz for a One-Dimensional Virtual Beam Element

The first and simplest virtual element formulation for a beam relies on a cubic ansatz. Here the nodal unknowns w_i are introduced together with the rotations $w_{,x\,i} = \theta_i$ to describe a beam element of length l_v, as depicted in Fig. 13.1b. The four unknowns w_1, θ_1, w_2 and θ_2 lead to a choice of a cubic ansatz

$$w_\pi = a_1 + a_2 x + a_3 x^2 + a_4 x^3 = \langle 1\; x\; x^2\; x^3\rangle \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} = \mathbf{H}_w^{(1,3)}(x)\,\mathbf{a}\,. \tag{13.3}$$

As in the bar example in Sect. 1.2.1 the parameters can be determined from the L^2-projection which has now to be written for the curvature w''

$$\int_0^{l_v} p''\,(w_h'' - w_\pi'')\,\mathrm{d}x = 0 \rightarrow \int_0^{l_v} p''\,w_\pi''\,\mathrm{d}x = \int_0^{l_v} p''\,w_h''\,\mathrm{d}x \tag{13.4}$$

Again p is a polynomial of the same order as w_π. Since second derivatives appear on the left hand side of the projection only the coefficients a_3 and a_4 can be determined from (13.4). Thus one can write

$$w_\pi'' = \langle 2\;6x\rangle \begin{Bmatrix} a_3 \\ a_4 \end{Bmatrix} = \mathbf{B}_w^{(1,3)}\hat{\mathbf{a}} \quad \text{and} \quad p'' = \langle 2\;6x\rangle = \mathbf{B}_w^{(1,3)} \tag{13.5}$$

which yields after integration a 2 × 2 matrix for the left side of the projection (13.4)

$$\int_0^{l_v} p''\,w_\pi''\,\mathrm{d}x = \int_0^{l_v} [\mathbf{B}_w^{(1,3)}]^T\mathbf{B}_w^{(1,3)}\mathrm{d}x\,\hat{\mathbf{a}} = l_v\begin{bmatrix} 4 & 6l_v \\ 6l_v & 12l_v^2 \end{bmatrix}\hat{\mathbf{a}} = \mathbf{G}_\pi\,\hat{\mathbf{a}}\,. \tag{13.6}$$

The right hand side of the projection (13.4) cannot be computed directly since w_h is not known along the beam axis. However by using partial integration twice we are able to shift w_h to single values at the boundary of the element which are known

$$\int_0^{l_v} p''\,w_h''\,\mathrm{d}x = (p''w_h')\big|_0^{l_v} - (p'''w_h)\big|_0^{l_v} + \int_0^{l_v} p''''\,w_h\,\mathrm{d}x\,. \tag{13.7}$$

In (13.3) we use a cubic polynomial hence the fourth order derivative p'''' is zero. Noting that the derivative p''' can be written as $\nabla\mathbf{B}_w^{(1,3)} = \langle 0\,6\rangle$ the matrix form of (13.7) follows with the right side of (13.5) as

$$\int_0^{l_v} p''\,w_h''\,\mathrm{d}x = \left([\mathbf{B}_w^{(1,3)}]^T w_h'\right)\big|_0^{l_v} - \left([\nabla\mathbf{B}_w^{(1,3)}]^T\,w_h\right)\big|_0^{l_v}\,. \tag{13.8}$$

The rotation w_h' is known at the nodes of the beam element: $w_h'(x=0) = \theta_1$ and at $w_h'(x = l_v) = \theta_2$, see Fig. 13.1b. The same is true for the deflection w_h: $w_h(x=0) = w_1$ and $w_h(x = l_v) = w_2$. By introducing these relations into the above equation and by combining the result with (13.6) the explicit matrix form of (13.4)

$$\mathbf{G}_\pi\,\hat{\mathbf{a}} = \begin{Bmatrix} 2 \\ 6l_v \end{Bmatrix}\theta_2 - \begin{Bmatrix} 2 \\ 0 \end{Bmatrix}\theta_1 - \begin{Bmatrix} 0 \\ 6 \end{Bmatrix}(w_2 - w_1) \tag{13.9}$$

is obtained. The solution of this equation yields the coefficients a_3 and a_4 as a function of the nodal unknowns $\mathbf{w}_v$

$$\hat{\mathbf{a}} = \frac{1}{l_v^2}\begin{bmatrix} -1 & -2l_v & 3 & -l_v \\ \frac{2}{l_v} & 1 & -\frac{2}{l_v} & 1 \end{bmatrix} \mathbf{w}_v = \hat{\mathbb{P}}\,\mathbf{w}_v \tag{13.10}$$

with $\mathbf{w}_v^T = \langle w_1\,\theta_1\,w_2\,\theta_2\rangle$. Based on this result the curvature w''_π can be expressed with (13.5) as $w''_\pi = \mathbf{B}_w^{(1,3)}(x)\,\hat{\mathbf{a}} = \mathbf{B}_w^{(1,3)}(x)\,\hat{\mathbb{P}}\,\mathbf{w}_v$. Now the strain energy of the beam element (13.2) will be approximated using w''_π

$$U_B = \frac{1}{2}\int\limits_0^l EI[w''_\pi(x)]^2 \mathrm{d}x = \frac{1}{2}\mathbf{w}_v^T\,\hat{\mathbb{P}}^T\,EI\int\limits_0^{l_v}[\mathbf{B}_w^{(1,3)}(x)]^T\,\mathbf{B}_w^{(1,3)}(x)\,\mathrm{d}x\,\hat{\mathbb{P}}\,\mathbf{w}_v\,. \tag{13.11}$$

We note that the integral in this equation was already evaluated in (13.6). Thus the stiffness matrix $\mathbf{K}^{B,V}$ of the virtual Euler-Bernoulli beam element can be computed explicitly by a second derivative of the potential with respect to the nodal unknowns

$$\mathbf{K}^{B,V} = \frac{\partial^2 U_B}{\partial \mathbf{w}_v \partial \mathbf{w}_v} = EI\,\hat{\mathbb{P}}^T\,\mathbf{G}_\pi\,\hat{\mathbb{P}} = \frac{EI}{l_v^3}\begin{bmatrix} 12 & 6l_v & -12 & 6l_v \\ 6l_v & 4l_v^2 & -6l_v & 2l_v^2 \\ -12 & -6l_v & 12 & -6l_v \\ 6l_v & 2l_v^2 & -6l_v & 4l_v^2 \end{bmatrix} \tag{13.12}$$

which is exactly the same stiffness matrix as for the finite element method with a cubic Hermitian ansatz function, see e.g. Oñate (2013).

What remains is to compute the loading term in the potential (13.2). The ansatz w_h is not known inside the element but can be approximated by w_π in (13.3). So far only a_3 and a_4 are known. To compute coefficients a_1 and a_2 the relation (1.13) in Sect. 1.2.1 has to be extended to include also the derivative of the deflection (the rotations). The idea is to equalise not only the average of the nodal degrees of freedom of w_h and the value of the projection w_π at the element nodes but also the rotations w'_h and w'_π. Hence the following conditions can be employed to compute the remaining coefficients

$$\sum_{i=1}^{2} w_h(x_i) = \sum_{i=1}^{2} w_\pi(x_i) \quad \text{and} \quad \sum_{i=1}^{2} w'_h(x_i) = \sum_{i=1}^{2} w'_\pi(x_i)\,. \tag{13.13}$$

By evaluating these equations at the nodal points $x = 0$ and $x = l_v$ we obtain for the deflections

$$w_1 + w_2 = a_1 + (a_1 + a_2 l_v + a_3 l_v^2 + a_4 l_v^3) \tag{13.14}$$

and the rotations

$$\theta_1 + \theta_2 = a_2 + (a_2 + 2a_3 l_v + 3a_4 l_v^2)\,. \tag{13.15}$$

By inserting a_3 and a_4 from (13.10) the result $a_2 = \theta_1$ and $a_1 = w_1$ follows. This result is in a way obvious since the ansatz (13.3) immediately reveals that $w_\pi(0) = a_1 \doteq w_1$ and $w'_\pi(0) = a_2 \doteq \theta_1$.

Now all coefficients are known as functions of the nodal degrees of freedom $\mathbf{w}_v$ which can be expressed by the projection

$$\mathbf{a} = \mathbb{P}\,\mathbf{w}_v \Longrightarrow \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -2l_v & 3 & -l_v \\ \frac{2}{l_v} & 1 & -\frac{2}{l_v} & 1 \end{bmatrix} \begin{Bmatrix} w_1 \\ \theta_1 \\ w_2 \\ \theta_2 \end{Bmatrix} \tag{13.16}$$

With (13.3) the projection w_π is complete

$$w_\pi(x) = \mathbf{H}_w^{(1,3)}(x)\,\mathbb{P}\,\mathbf{w}_v\,. \tag{13.17}$$

By inserting this ansatz into the loading term in (13.2), here written for the element v, we derive for $q(x) = q_0 = const.$

$$\int_0^{l_v} q(x)\,w_\pi(x)\mathrm{d}x = q_0 \int_0^{l_v} \mathbf{H}_w^{(1,3)}(x)\,\mathrm{d}x\,\mathbb{P}\,\mathbf{w}_v = \frac{q_0\,l_v}{2}\left\langle 1\ \tfrac{l_v}{6}\ 1\ -\tfrac{l_v}{6}\right\rangle \mathbf{w}_v \tag{13.18}$$

which is exactly the same result as provided by a cubic Hermitian finite element ansatz, see e.g. Oñate (2013).

Like for the truss in Sect. 1.2.1,the virtual element method leads for the Euler-Bernoulli beam to an identical result as the finite element method where the stiffness matrix has full rank. Thus the remainder $(w_h - w_\pi)$ in $w_h = w_\pi + (w_h - w_\pi)$ is zero.

13.1.2 Fourth Order Ansatz for a One-Dimensional Virtual Beam Element

This simple example will underline the methodology used to derive virtual elements for higher order ansatz function. For this purpose we discuss a fourth order (quadratic) ansatz for the problem $EI\,w''''(x) = q(x)$, see Fig. 1.8.

Analogously to the case of the cubic projection we select a quartic function

$$w_\pi = a_1 + a_2\,x + a_3\,x^2 + a_4\,x^3 + a_5\,x^4 = \mathbf{H}_w^{(1,4)}(x)\,\mathbf{a} \tag{13.19}$$

with $\mathbf{H}_w^{(1,4)}(x) = \langle 1\ x\ x^2\ x^3\ x^4\rangle$ and $\mathbf{a}^T = \langle a_1\,a_2\,a_3\,a_4\,a_5\rangle$. This projection function has five unknown parameters and the question is: how will it be possible to derive a virtual element since we have only two points at the edges of the element with nodal displacements w_1 and w_2 and the nodal rotations θ_1 and θ_2?

The solution is analogous to Sect. 1.2.2. We look at the least square projection (13.4) together with (13.7)

$$\int_0^{l_v} p''\,(w_h'' - w_\pi'')\,\mathrm{d}x = 0 \rightarrow \int_0^{l_v} p''\,w_\pi''\,\mathrm{d}x = (p''w_h')\big|_0^{l_v} - (p'''w_h)\big|_0^{l_v} + \int_0^{l_v} p''''\,w_h\,\mathrm{d}x \tag{13.20}$$

where now p has the same polynomial degree as (13.19) and thus is a quartic polynomial with $p'''' = const$. Hence the last integral $\int w_h\,\mathrm{d}x$ does not vanish. But it is also not computable since w_h is not known inside the element. As in Sects. 1.2.2 and 3.1.7 we introduce the internal variable (so called moment)

$$m_0 = \frac{1}{l_v}\int_0^{l_v} w_h\,\mathrm{d}x \tag{13.21}$$

which is scaled by the element length such that m_0 has the same dimensions as w_1 and w_2. With this new variable the projection in (13.20) can be determined, as we will see next.

It is convenient to introduce a matrix formulation in order to shorten notation. This leads to the second derivative of w_π

$$w_\pi'' = 2a_3 + 6a_4\,x + 12a_5\,x^2 = \mathbf{B}_w^{(1,4)}(x)\,\hat{\mathbf{a}} \tag{13.22}$$

where $\mathbf{B}_w^{(1,4)}(x) = \langle 2\;6x\;12x^2\rangle$ and $\hat{\mathbf{a}}^T = \langle a_3\;a_4\;a_5\rangle$. By writing the second derivative of the polynomial p in matrix form $p'' = [\mathbf{B}_w^{(1,4)}]^T = \langle 2\;6x\;12x^2\rangle^T$ the left hand side of (13.20) yields

$$\int_0^{l_v} p''\,w_\pi''\,\mathrm{d}x = \int_0^{l_v} [\mathbf{B}_w^{(1,4)}(x)]^T\mathbf{B}_w^{(1,4)}(x)\,\mathrm{d}x\,\hat{\mathbf{a}} = \int_0^{l_v} \begin{bmatrix} 4 & 12x & 24x^2 \\ 12x & 36x^2 & 72x^3 \\ 24x^2 & 72x^3 & 144x^4 \end{bmatrix} \mathrm{d}x \begin{Bmatrix} a_3 \\ a_4 \\ a_5 \end{Bmatrix}. \tag{13.23}$$

The polynomials can be integrated exactly, leading to

$$\int_0^{l_v} p''\,w_\pi''\,\mathrm{d}x = \begin{bmatrix} 4l_v & 6l_v^2 & 8l_v^3 \\ 6l_v^2 & 12l_v^3 & 18l_v^4 \\ 8l_v^3 & 18l_v^4 & \frac{144}{5}l_v^5 \end{bmatrix} \begin{Bmatrix} a_3 \\ a_4 \\ a_5 \end{Bmatrix} = \mathbf{G}\,\hat{\mathbf{a}} \tag{13.24}$$

The right hand side in (13.20) can be evaluated with $p''' = \langle 0\;6\;24x\rangle^T$ and $p'''' = \langle 0\;0\;24\rangle^T$ resulting in

$$
\begin{aligned}
\mathbf{r}(w_{hi}) &= (p''\, w_h')\big|_0^{l_v} - (p'''\, w_h)\big|_0^{l_v} + \int\limits_0^{l_v} p''''\, w_h \,\mathrm{d}x \\
&= \left(\begin{Bmatrix} 2 \\ 6x \\ 12x^2 \end{Bmatrix} w_h' - \begin{Bmatrix} 0 \\ 6 \\ 24x \end{Bmatrix} w_h \right)\Bigg|_0^{l_v} + \begin{Bmatrix} 0 \\ 0 \\ 24 \end{Bmatrix} \int\limits_0^{l_v} w_h \,\mathrm{d}x \,.
\end{aligned}
\tag{13.25}
$$

With the unknowns defined in the right part of Fig. 13.1, $w_h(0) = w_1$, $w_h(l_v) = w_2$, $w_h'(0) = \theta_1$ and $w_h'(l_v) = \theta_2$, and the internal unknown m_0 the right hand side of (13.25) follows as

$$
\mathbf{r}(w_i, \theta_i, m_0) = \begin{Bmatrix} 2(\theta_2 - \theta_1) \\ 6\, l_v \theta_2 \\ 12 l_v^2 \theta_2 \end{Bmatrix} - \begin{Bmatrix} 0 \\ 6\,(w_2 - w_1) \\ 24 l_v w_2 \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ 24 l_v m_0 \end{Bmatrix}
\tag{13.26}
$$

The projection equation (13.20) can now be solved for the unknowns **a** by combining (13.24) and (13.26)

$$
\mathbf{G}\,\hat{\mathbf{a}} = \mathbf{r}(w_i, \theta_i, m_0) \rightarrow \begin{Bmatrix} a_3 \\ a_4 \\ a_5 \end{Bmatrix} = \begin{Bmatrix} -\frac{3}{2 l_v^2}(12 w_1 + 3 l_v \theta_1 + 8 w_2 - 4 l_v \theta_2 - 20 m_0) \\ \frac{2}{l_v^3}(16 w_1 + 3 l_v \theta_1 + 14 w_2 - 2 l_v \theta_2 - 30 m_0) \\ -\frac{5}{2 l_v^4}(6 w_1 + l_v \theta_1 + 6 w_2 - l_v \theta_2 - 12 m_0) \end{Bmatrix} .
\tag{13.27}
$$

Furthermore the constants a_1 and a_2 can be obtained by the conditions that the average of the projection w_π and its derivative w_π' evaluated at the nodal points is equal to the average of the ansatz w_h and its derivative w_h'at the nodal points for one element

$$
\sum_{k=1}^{2} w_\pi(x_k) = \sum_{k=1}^{2} w_k \,, \qquad \sum_{k=1}^{2} w_\pi'(x_k) = \sum_{k=1}^{2} \theta_k \,.
\tag{13.28}
$$

The first equation leads with $x_1 = 0$, $x_2 = l_v$ and the ansatz (13.19) to

$$
a_1 + (a_1 + a_2\, l_v + a_3\, l_v^2 + a_4\, l_v^3 + a_5\, l_v^4) = w_1 + w_2
\tag{13.29}
$$

and the second condition results in

$$
a_2 + (a_2 + 2 a_3\, l_v + 3 a_4\, l_v^2 + 4 a_5\, l_v^3) = \theta_1 + \theta_2 \,.
\tag{13.30}
$$

The two equations above yield a matrix system from which a_1 and a_2 can be determined

$$
\begin{bmatrix} 2 & l_v \\ 0 & 2 \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} w_1 + w_2 \\ \theta_1 + \theta_2 \end{Bmatrix} - \begin{bmatrix} l_v^2 & l_v^3 & l_v^4 \\ 2 l_v & 3 l_v^2 & 4 l_v^3 \end{bmatrix} \begin{Bmatrix} a_3 \\ a_4 \\ a_5 \end{Bmatrix} .
\tag{13.31}
$$

This leads after some algebra with (13.27) to $a_1 = w_1$ and $a_2 = \theta_1$. Again, the result is obvious since for $x = 0$ the ansatz function should give the nodal values w_1 and θ_1 at this point.

As for the virtual beam element with cubic ansatz in Sect. 13.1 a projector, see (13.16), can be defined which expresses the constants $\mathbf{a}^T = \langle a_1\, a_2\, a_3\, a_4\, a_5 \rangle$ in terms of the nodal values and the moment $\mathbf{w}_v^T = \langle w_1\, \theta_1\, w_2\, \theta_2\, m_0 \rangle$ as

$$\mathbf{a} = \mathbb{P}\,\mathbf{w}_v \Longrightarrow \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{Bmatrix} = \frac{1}{l_v^4} \begin{bmatrix} l_v^4 & 0 & 0 & 0 & 0 \\ 0 & l_v^4 & 0 & 0 & 0 \\ -18 l_v^2 & -\frac{9}{2} l_v^3 & -12 l_v^2 & +6 l_v^3 & 30 l_v^2 \\ 32 l_v & 6 l_v^2 & 28 l_v & -4 l_v^2 & -60 l_v \\ -15 & -\frac{5}{2} l_v & -15 & \frac{5}{2} l_v & 30 \end{bmatrix} \begin{Bmatrix} w_1 \\ \theta_1 \\ w_2 \\ \theta_2 \\ m_0 \end{Bmatrix} \tag{13.32}$$

The definitions in (13.19) and (13.22) together with

$$\mathbb{B} = \frac{1}{l_v^4} \begin{bmatrix} -18 l_v^2 & -\frac{9}{2} l_v^3 & -12 l_v^2 & +6 l_v^3 & 30 l_v^2 \\ 32 l_v & 6 l_v^2 & 28 l_v & -4 l_v^2 & -60 l_v \\ -15 & -\frac{5}{2} l_v & -15 & \frac{5}{2} l_v & 30 \end{bmatrix} \tag{13.33}$$

yield the matrix form of the projection of the ansatz function and its second derivative

$$w_\pi = \mathbf{H}_w^{(1,4)}(x)\,\mathbb{P}\,\mathbf{w}_v \quad \text{and} \quad w_\pi'' = \mathbf{B}_w^{(1,4)}(x)\,\mathbb{B}\,\mathbf{w}_v\,. \tag{13.34}$$

The derivative w_π'' from (13.34) can now be inserted into the potential energy and integrated using the result from (13.24)

$$U_v = \frac{1}{2}\int_0^{l_v} EI\,(w_\pi'')^2\,\mathrm{d}x = \frac{1}{2}\mathbf{w}_v^T \mathbb{B}^T \int_0^{l_v} [\mathbf{B}_w^{(1,4)}(x)]^T \mathbf{B}_w^{(1,4)}(x)\,\mathrm{d}x\,\mathbb{B}\mathbf{w}_v = \frac{1}{2}\mathbf{w}_v^T \mathbb{B}^T \mathbf{G}\,\mathbb{B}\mathbf{w}_v. \tag{13.35}$$

Now the stiffness matrix of the virtual element follows by differentiation with respect to $\mathbf{w}_v$ as

$$\mathbf{K}^{B,V} = \frac{\partial^2 U_v}{\partial \mathbf{w}_v \partial \mathbf{w}_v} = \mathbb{B}^T \mathbf{G}\,\mathbb{B} = \frac{EI}{l_v^3} \begin{bmatrix} 192 & 36 l_v & 168 & -24 l_v & -360 \\ 36 l_v & 9 l_v^2 & 24 l_v & -3 l_v^2 & -60 l_v \\ 168 & 24 l_v & 192 & -36 l_v & -360 \\ -24 l_v & -3 l_v^2 & -36 l_v & 9 l_v^2 & 60 l_v \\ -360 & -60 l_v & -360 & 60 l_v & 720 \end{bmatrix} \tag{13.36}$$

It is interesting to note that there is no adequate FEM formulation that yields a finite beam element with 5 degrees of freedom.

The loading term follows from the potential, see (13.2),

$$U_q = \int\limits_0^{l_v} q(x)\, w_h \, \mathrm{d}x \tag{13.37}$$

Interestingly, for the case of $q(x) = q_0 = const.$ the integral $\int w_h \, \mathrm{d}x$ can be evaluated directly using the definition (13.21) of the variable m_0. This yields

$$U_q = q_0 \int\limits_0^{l_v} w_h \, \mathrm{d}x = q_0 \, m_0 \, l_v \tag{13.38}$$

Thus the matrix form of the loading term is simply given by[1]

$$\mathbf{f}_0^{B,V} = q_0 \, l_v \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{Bmatrix} \tag{13.39}$$

which is similar to the constant loading term of the truss element with quadratic ansatz function, see (1.36). For a load $q(x) = (1 - \frac{x}{l_v})\, q_1 + \frac{x}{l_v}\, q_2$ that varies linearly in the virtual element the potential of the loading term follows by inserting w_π from (13.34)

$$\begin{aligned} U_q = &\int\limits_0^{l_v} \left[\left(1 - \frac{x}{l_v} \right) q_1 + \frac{x}{l_v} q_2 \right] \mathbf{H}_w^{(1,4)}(x)\, \mathbb{P}\, \mathbf{w}_v \, \mathrm{d}x = \\ &\frac{l_v}{120} \left[(q_1 - q_2)[12(w_1 - w_2) + l_v(\theta_1 + \theta_2)] + 60\,(q_1 + q_2) m_0 \right] \end{aligned} \tag{13.40}$$

where the two constants q_1 and q_2 describe the load magnitude at the nodes. Differentiation of U_q with respect to the nodal unknowns yields the load vector for one element

$$\mathbf{f}_1^{B,V} = \frac{\partial U_q}{\partial \mathbf{w}_v} = \frac{l_v}{120} \begin{Bmatrix} 12(q_1 - q_2) \\ (q_1 - q_2) l_v \\ -12(q_1 - q_2) \\ (q_1 - q_2) l_v \\ 60(q_1 + q_2) l_v \end{Bmatrix} \tag{13.41}$$

The accuracy of the developed element is investigated using a clamped beam, see Fig. 13.2. Only one element is considered for a constant load, $q_1 = q_2 = q_0$ leading to $\mathbf{f}_0$ in (13.39). With the boundary conditions for the clamped beam ($w_1 = \theta_1 = 0$),

[1] One can evaluate U_q in a different way by inserting the ansatz for w_π, see (13.34) into the integral in (13.38). This will actually lead after some algebraic manipulations to exactly the same result.

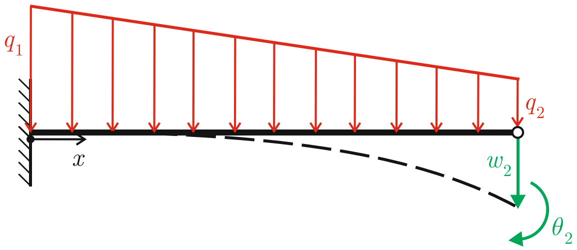


Fig. 13.2 One beam element under linearly varying line load

see Fig. 13.2, the first and second row and column of the element stiffness matrix (13.36) can be eliminated. In line with that, the first two rows of the load vector are deleted as well which yields the equation system and the solution

$$\begin{bmatrix} 192 & -36l_v & -360 \\ -36l_v & 9l_v^2 & 60l_v \\ -360 & 60l_v & 720 \end{bmatrix} \begin{Bmatrix} w_2 \\ \theta_2 \\ m_0 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ q_0\, l_v \end{Bmatrix} \Longrightarrow \begin{Bmatrix} w_2 \\ \theta_2 \\ m_0 \end{Bmatrix} = \frac{q_0\, l_v^3}{EI} \begin{Bmatrix} \frac{l_v}{8} \\ \frac{1}{6} \\ \frac{l_v}{20} \end{Bmatrix} \tag{13.42}$$

The nodal values w_2 and θ_2 are identical to the analytical results. Furthermore the moment distribution can be computed within the beam: $M(x) = -EI\, w''_\pi$. By introducing the results of (13.42) in (13.34) the moment distribution

$$M(x) = EI\,\mathbf{B}_w^{(1,4)}(x)\,\mathbb{B}\,\mathbf{w}_v = -\frac{q_0}{2}(l_v - x)^2 \tag{13.43}$$

is obtained. Again, the result matches the analytical solution hence the analytical solution is a fourth order polynomial and thus the quartic ansatz (13.19) approximates the solutions exactly.

A computation using the linearly distributed load yields the equation system and solution

$$\begin{aligned} &\begin{bmatrix} 192 & -36l_v & -360 \\ -36l_v & 9l_v^2 & 60l_v \\ -360 & 60l_v & 720 \end{bmatrix} \begin{Bmatrix} w_2 \\ \theta_2 \\ m_0 \end{Bmatrix} = \begin{Bmatrix} -(q_1 - q_2)\frac{l_v}{10} \\ (q_1 - q_2)\frac{l_v^2}{120} \\ (q_1 + q_2)\,\frac{l_v}{2} \end{Bmatrix} \\ &\Longrightarrow \begin{Bmatrix} w_2 \\ \theta_2 \\ m_0 \end{Bmatrix} = \frac{l_v^3}{24EI} \begin{Bmatrix} \frac{l_v}{5}(4q_1 + 11q_2) \\ q1 + 3q_2 \\ \frac{l_v}{15}(5q_1 + 13q_2 \end{Bmatrix} \end{aligned} \tag{13.44}$$

The result demonstrates that w_2 and θ_2 coincide with the analytical solution at the right element node. This can actually be shown for all complete polynomial ansatz functions that fulfil the homogeneous differential equation of the beam. The result (13.44) produces the approximate bending moment distribution

$$M_1^{B,V}(x) = -EI\mathbf{B}_w^{(1,4)}(x)\mathbb{B}\,\mathbf{w}_v = -(19q_1 + 41q_2)\frac{l_v^2}{120} + (2q_1 + 3q_2)\frac{l_v x}{5} - (q_1 + q_2)\frac{x^2}{4}$$

which is not the same as the analytical solution

$$M_1^{B,A}(x) = -\frac{(l_v - x)^2}{6l_v}\left[(q_1 + 2q_2)l_v + (q_2 - q_1)x\right] , \tag{13.45}$$

being a cubic function in x.

13.1.3 Static Condensation of the Moments

When using the classical ansatz functions, linear for the truss and cubic Hermitian for the beam, the virtual elements are identical to finite elements. For higher order ansatz functions there is a difference between virtual and finite element formulations, as shown in Sects. 1.2 and 13.1. These higher order ansatz spaces yield a more accurate resolution of the displacements and resultant stresses. Hence higher order virtual element ansatz functions can be employed when an increased accuracy for non-homogeneous problems is of interest.

As outlined in Wriggers (2022), the basic linear finite element formulation is recovered when a static condensation of the moments m_i is performed. The stiffness matrix and right hand side can be split at element level in the nodal displacement variables $\mathbf{u}_I$ and the moments $\mathbf{m}_M$

$$\begin{bmatrix} \mathbf{K}_{II} & \mathbf{K}_{IM} \\ \mathbf{K}_{MI} & \mathbf{K}_{MM} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_I \\ \mathbf{m}_M \end{Bmatrix} = \begin{Bmatrix} \mathbf{R}_I \\ \mathbf{R}_M \end{Bmatrix} . \tag{13.46}$$

With $\bar{\mathbf{K}}_{II} = \left(\mathbf{K}_{II} - \mathbf{K}_{IM}\, \mathbf{K}_{MM}^{-1}\mathbf{K}_{MI}\right)$ we obtain two equations for $\mathbf{u}_I$ and $\mathbf{m}_M$

$$\begin{aligned} \bar{\mathbf{K}}_{II}\, \mathbf{u}_I &= \mathbf{R}_I - \mathbf{K}_{IM}\mathbf{K}_{MM}^{-1}\mathbf{R}_M \\ \mathbf{K}_{MM}\, \mathbf{m}_M &= (\mathbf{R}_M - \mathbf{K}_{MI}\, \mathbf{u}_I) \end{aligned} \tag{13.47}$$

It can be shown that the matrix $\bar{\mathbf{K}}_{II}$ is equivalent to the matrix representing the cubic ansatz given in (13.12) for any order of the ansatz used to derive the virtual element.[2] It is also well known that the stiffness matrix related to the cubic ansatz leads to exact nodal deflections w_i and nodal rotations θ_i for any given right hand side, see Tong (1969), Hughes (1987).

However the approximation of the deflection within the cubic beam element are only exact for $q(x) = 0$ in (13.1). Hence it is for $q(x) \neq 0$ appealing to use a virtual element formulation with higher order approximations which yields a better representation of the beam deflection and with this also of the bending moment and shear force within the element. Classically, analytical results can be recovered for the load

[2] This is consistent with the result that the third order ansatz exactly solves the homogenous part of the beam Eq. (13.1).

$q(x) = \sum_{k=0}^{m} q_k x^m$, for $m \geq 0$ by using an ansatz $\mathbf{H}^{(1,4+m)}$, see e.g. (13.19) for $m = 0$.

Based on these observations, a postprocessing step can be introduced which is based on the following procedure:

1. Compute the nodal deflections and rotations using the cubic beam formulation which is equivalent to the finite element beam. This yields the exact nodal degrees of freedom $\mathbf{u}_I$.
2. Determine the moments $\mathbf{m}_M$ that follow from $(13.47)_2$ which completes the unknowns for the virtual element with a higher order ansatz.
3. The higher order approximation w_π of the deflection w within the element can then be computed by constructing a projector, see e.g. (13.32) for the fourth order virtual element ansatz.

Thus the accuracy of a classical beam element can be enhanced and a higher order solution at local element level is obtained by using the virtual element formulation.

The above procedure of static condensation can also be applied for the truss element discussed in the introduction, see Sect. 1.2. In case of trusses, a linear ansatz solves already the homogeneous equations (1.1). Hence, analytical results can be recovered for the load $q(x) = \sum_{k=0}^{m} q_k x^m$, for $m \geq 0$ by employing an ansatz $\mathbf{H}^{(1,2+m)}$. Again, an elimination of the internal degrees of freedom is possible by using $(13.47)_1$ which yields the stiffness matrix (1.33) related to the linear ansatz function. Then from $(13.47)_2$ the nodal degrees of freedom u_i and moments m_k for a higher order approximation can be obtained. These lead to an enhanced approximation u_π for the virtual truss element.

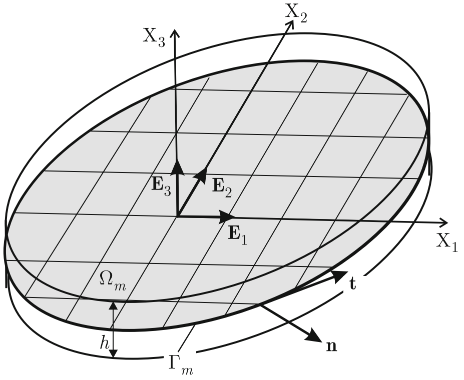
13.2 Kirchhoff-Love Plates

In this section the basic equations are summarized that govern the deformation of plates under general loading for small deflections and rotations. The theory is written as well for isotropic as for anisotropic elastic constitutive behaviour.

13.2.1 Mathematical Model of the Plate and Constitutive Relations

The geometry of the plate is described by its mid surface Ω_m and its thickness h. The thickness is assumed to be constant. A point $\boldsymbol{X}$ within the plate can be described by the sum of its projection $\boldsymbol{X}_m$ onto the mid surface and a vector in normal direction: $\boldsymbol{X} = \boldsymbol{X}_m + \boldsymbol{Z}$. The normal vector is given as $\boldsymbol{Z} = X_3\, \boldsymbol{E}_3$. The base vector $\boldsymbol{E}_3$ is

Fig. 13.3 Plate geometry



normal to the mid-surface of the plate Ω_m and X_3 is the associated coordinate, see Fig. 13.3,

$$\boldsymbol{X} = \boldsymbol{X}_m + X_3 \boldsymbol{E}_3 = X_\alpha \boldsymbol{E}_\alpha + X_3 \boldsymbol{E}_3 \tag{13.48}$$

where $(\boldsymbol{E}_1, \boldsymbol{E}_2, \boldsymbol{E}_3)$ is an orthonormal Cartesian basis and index $\alpha \in \{1, 2\}$.

The displacement $\boldsymbol{u}(\boldsymbol{X}_m, X_3)$ of point $\boldsymbol{X}$ can be described in the Kirchhoff-Love theory by the deflection w of the plate and the rotation $w_{,\alpha}$ around the axes $\boldsymbol{E}_\alpha$[3]

$$\boldsymbol{u}(\boldsymbol{X}_m, X_3) = w(\boldsymbol{X}_m)\, \boldsymbol{E}_3 - X_3\; w_{,\alpha}(\boldsymbol{X}_m)\, \boldsymbol{E}_\alpha \tag{13.50}$$

where $w_{,\alpha}$ is the derivative of w with respect to the coordinates X_α which can be written as the gradient $\nabla w = w_{,\alpha}\, \boldsymbol{E}_\alpha$.

The in-plane strain $\boldsymbol{\varepsilon}$ associated with the displacement $\boldsymbol{u}$ is given with (2.9) by

$$\boldsymbol{\varepsilon}(\boldsymbol{u}) = \nabla_s \boldsymbol{u} = X_3\, \boldsymbol{\kappa} \tag{13.51}$$

where the curvature operator $\boldsymbol{\kappa}$ is given by

$$\boldsymbol{\kappa} = -w_{,\alpha\beta}\, \boldsymbol{E}_\alpha \otimes \boldsymbol{E}_\beta = -\nabla\nabla w\,. \tag{13.52}$$

The equilibrium equations of the plate derive from the balance of momentum (2.22) where it is assumed that the plate is symmetrical with respect to its mid-surface leading to the range of X_3: $(-\frac{h}{2} \leq X_3 \leq \frac{h}{2})$. After some algebraic work the partial differential equations describing the plate equilibrium follows as

[3] A rotation $\boldsymbol{\theta}$ that rotates in positive direction around the axes $\boldsymbol{E}_\alpha$ follows from

$$\boldsymbol{\theta} = \theta_X \boldsymbol{E}_X + \theta_Y \boldsymbol{E}_Y = w_{,\alpha} \boldsymbol{E}_\alpha \times \boldsymbol{E}_3 = w_{,2}\, \boldsymbol{E}_1 - w_{,1}\, \boldsymbol{E}_2 \tag{13.49}$$

.

$$\begin{array}{rcl} M_{\alpha\beta,\beta} - Q_\alpha = 0 & \Leftrightarrow & \mathrm{Div}\,\boldsymbol{M} - \boldsymbol{Q} = 0 \\ Q_{\alpha,\alpha} + \bar{q} = 0 & \Leftrightarrow & \mathrm{Div}\,\boldsymbol{Q} + \bar{q} = 0 \end{array} \tag{13.53}$$

with the distributed load $\bar{q}$, the shear force $\boldsymbol{Q} = Q_\alpha\, \boldsymbol{E}_\alpha$ and the moments

$$\boldsymbol{M} = M_{\alpha\beta}\, \boldsymbol{E}_\alpha \otimes \boldsymbol{E}_\beta \quad \text{with} \quad M_{\alpha\beta} = \int_{-h/2}^{h/2} X_3\, \sigma_{\alpha\beta}\, \mathrm{d}X_3 \tag{13.54}$$

where $\sigma_{\alpha\beta}$ are the stresses within the plate. Equations (13.53) yield

$$M_{\alpha\beta,\alpha\beta} + \bar{q} = 0 \;\Leftrightarrow\; \mathrm{Div}[\mathrm{Div}\boldsymbol{M}] - \bar{q} = 0 \tag{13.55}$$

The Dirichlet and von Neumann boundary conditions can be summarized as

$$\begin{array}{rccl} w = \bar{w} & \text{and} & \nabla w \cdot \boldsymbol{n} = \bar{w}_{,n} & \text{on } \Gamma_D \\ (\mathrm{Div}\,\boldsymbol{M}) \cdot \boldsymbol{n} + M_{nt,s} = \bar{Q}_n & \text{and} & \boldsymbol{n} \cdot \boldsymbol{M}\boldsymbol{n} = \bar{M}_{nn} & \text{on } \Gamma_N \end{array} \tag{13.56}$$

where $(\bar{\bullet})$ are the prescribed values. The normal and tangent vectors $\boldsymbol{n}$ and $\boldsymbol{t}$ at the boundary Γ_m can be found in Fig. 13.3.

The shear force cannot be determined from a constitutive relation and follows from the equilibrium of the moments (13.53) as $\boldsymbol{Q} = \mathrm{Div}\,\boldsymbol{M}$. In the first von Neumann boundary condition, the shear force is amended by the twisting moment and thus a so called substitute or total shear force $Q_n + M_{nt,s} = (\mathrm{Div}\,\boldsymbol{M}) \cdot \boldsymbol{n} + M_{nt,s}$ is generally introduced, see e.g. Timoshenko and Woinowsky-Krieger (1959). Here the last term is given by $M_{nt,s} = M_{12,2}\, n_1 + M_{12,1}\, n_2$ where n_1 and n_2 are the components of the normal vector $\boldsymbol{n}$.

Isotropic elastic behaviour. The elastic deformation of the plate is described by Hooke's law under plane stress condition, see (2.47). With the associated constitutive tensor $\mathbb{C}_{pl\sigma}$ a constitutive relation can be derived with $\boldsymbol{\sigma} = \mathbb{C}_{pl\sigma}\boldsymbol{\varepsilon}$ that links the moment of the in-plane stress to the curvature using (13.51) and (13.54)

$$\boldsymbol{M} = \int_{-h/2}^{h/2} X_3^2\, \mathbb{C}_{pl\sigma}\boldsymbol{\kappa}\, \mathrm{d}X_3\,. \tag{13.57}$$

For a plate with constant thickness the integral (13.57) yields

$$\boldsymbol{M} = \frac{h^3}{12}\,\mathbb{C}_{pl\sigma}\,\boldsymbol{\kappa} = D_b\left(\boldsymbol{\kappa} + \frac{\nu}{1-\nu}\,\mathrm{tr}(\boldsymbol{\kappa})\,\boldsymbol{1}\right) = \mathbb{D}_b\,\boldsymbol{\kappa} \tag{13.58}$$

with the unit tensor in two dimensions $\boldsymbol{1} = \delta_{\alpha\beta}\, \boldsymbol{E}_\alpha \otimes \boldsymbol{E}_\beta$, Young's modulus E, Poisson ratio ν and the bending stiffness D_b

$$D_b = \frac{E\,h^3}{12(1-\nu^2)}\,. \tag{13.59}$$

By inserting the constitutive relations (13.58) into the equilibrium equation (13.55) one arrives at the classical fourth order partial differential equation for the plate deflection w

$$D_b \, \Delta\Delta w = \bar{q} \, . \tag{13.60}$$

Furthermore, the elastic strain energy density can be introduced

$$\psi(w) = \frac{1}{2}\,\mathrm{tr}[\boldsymbol{M}\,\boldsymbol{\kappa}] = \frac{1}{2}\boldsymbol{\kappa} \cdot \mathbb{D}\,\boldsymbol{\kappa} = \frac{D_b}{2}\left[\boldsymbol{\kappa} \cdot \boldsymbol{\kappa} + \frac{\nu}{1-\nu}(\mathrm{tr}\,\boldsymbol{\kappa})^2\right] \tag{13.61}$$

and a potential describing the plate deflection follows with (13.56) as[4]

$$U_{pl} = \int\limits_{\Omega_m} (\psi(w) - \bar{q}\, w)\mathrm{d}A - \int\limits_{\Gamma_m} (\bar{Q}_n\, w + \bar{M}_{nn}\, w_{,n})\, \mathrm{d}\Gamma \tag{13.62}$$

where $\bar{q}$ is the distributed transverse load and $\bar{Q}_n$, $\bar{M}_n$ and M_t are the transverse load and moments at the Neumann boundary Γ_N, respectively.

Often it is advantageous to introduce Voigt notation which can be used to rewrite the moments and curvatures

$$\mathbf{M} = \left\{ \begin{matrix} M_{xx} \\ M_{yy} \\ M_{xy} \end{matrix} \right\}, \quad \boldsymbol{\Upsilon} = \left\{ \begin{matrix} w_{,xx} \\ w_{,yy} \\ 2\, w_{,xy} \end{matrix} \right\}. \tag{13.63}$$

where the strain $\boldsymbol{\Upsilon}$ is now defined as the negative curvature. The constitutive relation may be expressed through the matrix $\mathbb{D}$ that is defined with (13.59) by

$$\mathbf{M} = -\mathbb{D}\,\boldsymbol{\Upsilon} = -D_b \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \boldsymbol{\Upsilon} \tag{13.64}$$

and the strain energy in (13.61) can be rewritten in Voigt notation as

$$\psi(w) = \frac{1}{2}\Upsilon^T\, \mathbb{D}\,\boldsymbol{\Upsilon} \, . \tag{13.65}$$

Anisotropic elastic behaviour. Laminated plates consist of layers of fibers (composite plies) that are embedded in a matrix and have different directions. Thus it is necessary to model the anisotropy of these layer, see e.g. Reddy (2004). Generally the constitutive response of several layers (stack of composite plies) has to be modelled.

[4] The appearance of second order derivatives of w in the strain energy density (13.61) requires a C^1-continuous ansatz for the deflection w within a numerical method, like the finite or virtual element formulation.

Each ply consists of fibers that point into a direction with the angle ϕ with respect to the X-axis.

For each ply k the orthotropic constitutive matrix is defined with respect to the orthotropic basis (the notation $(\bar{\bullet})$ refers to the orthotropic basis, while not using a bar refers to the global Cartesian basis):

$$\bar{\mathbb{D}}_k = \frac{1}{1 - \nu_{12}\nu_{21}} \begin{bmatrix} E_1 & \nu_{12}E_2 & 0 \\ \nu_{12}E_2 & E_2 & 0 \\ 0 & 0 & G_{12} \end{bmatrix} \tag{13.66}$$

where E_1 relates to the stiffness in fiber direction and E_2 is the stiffness perpendicular to the fiber direction. The Poisson ration of the ply is given by ν_{12} and the shear modulus is G_{12}. The Poisson ratio $\nu_{21} = \frac{E_2}{E_1}\nu_{12}$ is a dependent variable. The orthotropic constitutive matrix $\bar{\mathbb{D}}_k$ has to be transformed to the Cartesian X-Y coordinate system, describing the mid-plane of the plate by the transformation,

$$\mathbb{D}_k = \boldsymbol{T}^{-1}\bar{\mathbb{D}}_k\boldsymbol{T}^{-T} \tag{13.67}$$

with

$$\boldsymbol{T}^{-1} = \begin{bmatrix} \cos\phi^2 & \sin\phi^2 & -2\sin\phi\cos\phi \\ \sin\phi^2 & \cos\phi^2 & 2\sin\phi\cos\phi \\ \sin\phi\cos\phi & -\sin\phi\cos\phi & \cos\phi^2 - \sin\phi^2 \end{bmatrix}. \tag{13.68}$$

Now an integration over the thickness has to be performed considering all n_{pl} plies. This leads, analogous to (13.57), to a sum over all plies and defines the global material stiffness matrix $\mathbb{D}_G$

$$\mathbb{D}_G = \frac{1}{3}\sum_{k=1}^{n_{pl}} \mathbb{D}_k(Z_k^3 - Z_{k-1}^3) \tag{13.69}$$

where the integration is executed for each ply over the ply thickness $h_k = Z_k - Z_{k-1}$. Now the strain energy for the laminated plate is given in Voigt notation by

$$\psi(w) = \frac{1}{2}\boldsymbol{\Upsilon}^T\,\mathbb{D}_G\,\boldsymbol{\Upsilon} \tag{13.70}$$

and the moments follow from $\mathbf{M} = -\mathbb{D}_G\,\boldsymbol{\Upsilon}$.

13.3 Formulation of the Virtual Element

The virtual plate element for Kirchhoff-Love kinematics can be derived in the same way as the virtual elements for solids. However, the ansatz space has to be changed due to the C^1-continuity requirements. Furthermore, the projection onto the polynomial space is different, as has been shown in Sect. 13.1 for the beam.

13.3.1 General Notations

A virtual plate element Ω_v is described in Fig. 13.4.

The element has n_V vertices. The boundary Γ_v consists of n_E edges Γ_e with normal and tangential vectors $(\boldsymbol{N}_e\,, \boldsymbol{T}_e)$ at each edge. The normal $\boldsymbol{N}_e$ and tangent $\boldsymbol{T}_e$ are given for a segment e by

$$\boldsymbol{N}_e = \begin{Bmatrix} n_X \\ n_Y \end{Bmatrix}_e = \frac{1}{l_e} \begin{Bmatrix} -(Y_{2e+1} - Y_{2e-1}) \\ X_{2e+1} - X_{2e-1} \end{Bmatrix}_e \quad \text{and } \boldsymbol{T}_e = \begin{Bmatrix} t_X \\ t_Y \end{Bmatrix}_e = \frac{1}{l_e} \begin{Bmatrix} X_{2e+1} - X_{2e-1} \\ Y_{2e+1} - Y_{2e-1} \end{Bmatrix}_e \qquad (13.71)$$

where l_e is the length of the segment e and $(X_k\,, Y_k)$ with $k = \{2e-1, 2e+1\}$ are the vertex nodes defining the segment. Furthermore a non-dimensional local coordinate $\xi = \frac{X_e}{l_e}$ is introduced along each element edge Γ_e.

The virtual element has in general different degrees of freedom at the vertices and the midnodes as pointed out in Fig. 13.4 where unknowns for the deflection w and the rotations $\boldsymbol{\theta}$ are provided at the vertices and the rotation θ_t at a midnodes.

The rotations $\boldsymbol{\theta}$ are used as degrees of freedom. Their vectors point in positive direction of the coordinate axes ($\boldsymbol{\theta} = \theta_\alpha\, \boldsymbol{E}_\alpha$) and are connected to the derivatives of the deflection by $\theta_x = w_{,y}$ and $\theta_y = -w_{,x}$. At an edge Γ_e, see Fig. 13.4, the following relations hold

$$\begin{aligned} \nabla w \cdot \boldsymbol{T}_e &= w,_s = \quad \boldsymbol{\theta} \cdot \boldsymbol{N}_e = \theta_n \\ \nabla w \cdot \boldsymbol{N}_e &= w,_n = -\boldsymbol{\theta} \cdot \boldsymbol{T}_e = -\theta_t \end{aligned} \qquad (13.72)$$

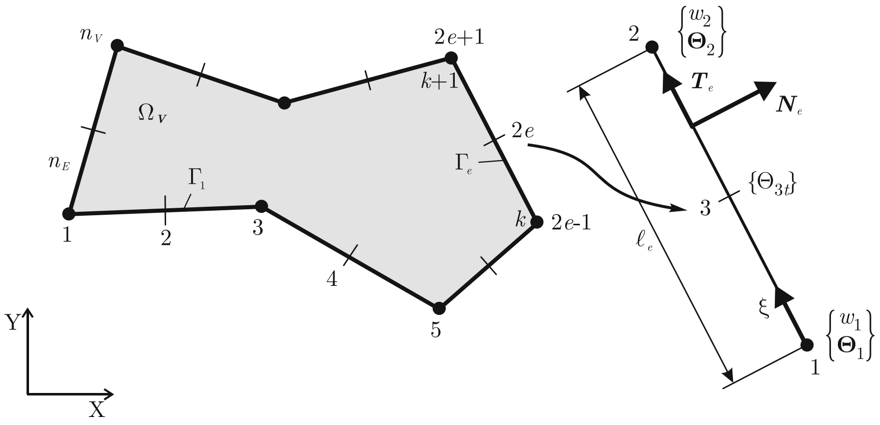


Fig. 13.4 Virtual element for the Kirchhoff-Love plate

13.3.2 Ansatz and Projection

In Sect. 3.1 an ansatz space was defined for the virtual solid element. Such a space has now to be defined for the plate problem as well. Since the Kirchhoff-Love plates need a C^1-continuous ansatz the following space for an ansatz of order n is introduced

$$V_{h|\Omega_v} = \{w_h \in H^2(\Omega_v),\, \Delta^2 w_h \in P_{n-4}(\Omega_v),\, w_{h,t} \in P_r(\Gamma_e),\, w_{h,n} \in P_s(\Gamma_e)\} \tag{13.73}$$

where we set $P_{n-4}(\Omega_v) = \{0\}$ for $n < 4$. The parameters r and s describe the ansatz order for the rotations around normal and tangent direction $w_{h,t}$ and $w_{h,n}$, respectively, and can be chosen differently.[5] The condition $\Delta^2 w \in P_{n-4}(\Omega_v)$ is fundamental for the element to be uni-solvent which ensures the uniqueness of the deflection w inside the element.

It is not difficult to create virtual plate elements that fulfill C^1-continuity since the ansatz has only to be formulated at the boundary. Within such ansatz only the deflection and the two rotations w_h, $w_{h,x}$ and $w_{h,y}$ appear as unknowns which is different to classical Kirchhoff-Love plate elements when using the finite element method. There C^1-continuous elements need higher order variables, e.g. $w_{,xx}$ as unknowns which need special interpretation, see e.g. Argyris et al. (1968).

The ansatz function for the deflection w_h has to be formulated in two dimensions.[6] It has the properties

(a0): w_h , $w_{h,t}$ and $w_{h,n}$ are known at the vertices k of the polygon Ω_v,
(a1): w_h is a polynomial P_n of degree n at each edge $\Gamma_e \in \Gamma_v$,
(a2): $w_{h,t}$ is a polynomial P_r of degree $r = n - 1$ at each edge $\Gamma_e \in \Gamma_v$,
(a3): $w_{h,n}$ is a polynomial P_s of degree s at each edge $\Gamma_e \in \Gamma_v$ with additional unknowns of $w_{h,n}$ at $s - 1$ equally spaced points in Γ_e,
(a4): w_h , $w_{h,t}$ and $w_{h,n}$ are continuous at all edges $\Gamma_e \in \Gamma_v$ of the polygon Ω_v,
(a5): $\Delta^2 w_h$ is a polynomial of degree P_{n-4} on the polygon Ω_v.

With these definitions, the ansatz for the deflection w_h is a harmonic function inside Ω_v which is only known at the edges Γ_e of Ω_v. The ansatz w_h is not known inside of the polygon Ω_v.

The L^2-projection, as defined in (3.10) for solids, is now formulated in terms of the second order gradients that are equivalent to the curvature (13.52). Here the weighting function $v \in N_\pi^n$ is a polynomial that has the same order as the ansatz for w_π, see (3.3). Two more averaging conditions, equivalent to (3.11), take care of the constant and linear terms that disappear when evaluating $\nabla\nabla w_\pi$. Thus three conditions, see e.g. Brezzi and Marini (2013), have to be introduced

[5] By selecting adequate ansatz functions for w_h. $w_{h,t}$ and $w_{h,n}$ it is even possible to generate virtual plate elements that have a higher continuity than C^1, see e.g. Brezzi and Marini (2013).

[6] As discussed for beams, it is necessary to introduce moments $\boldsymbol{m}$ when the ansatz order exceeds a certain polynomial degree. In this section we avoid the introduction of moments by choosing low ansatz orders of $n = 3$ and $s \in \{1, 2\}$.

$$\begin{aligned}\int\limits_{\Omega_v} \nabla\nabla w_\pi \cdot \nabla\nabla v \,\mathrm{d}\Omega &= \int\limits_{\Omega_v} \nabla\nabla w_h \cdot \nabla\nabla v \,\mathrm{d}\Omega\,,\\ \int\limits_{\Omega_v} \nabla w_\pi \,\mathrm{d}\Omega &= \int\limits_{\Omega_v} \nabla w_h \,\mathrm{d}\Omega\,,\\ \int\limits_{\Gamma_v} w_\pi \,\mathrm{d}\Omega &= \int\limits_{\Gamma_v} w_h \,\mathrm{d}\Omega\,.\end{aligned} \tag{13.74}$$

Since w_h is not known within Ω_v the right hand side of the first two equations has to be transferred to boundary integrals as in the solid case, see e.g. (3.14). The first equation yields

$$\begin{aligned}\int\limits_{\Omega_v} \nabla\nabla w_h \cdot \nabla\nabla v \,\mathrm{d}\Omega &= \int\limits_{\Gamma_v} (\nabla w_h \otimes \boldsymbol{N}) \cdot \nabla\nabla v \,\mathrm{d}\Gamma - \int\limits_{\Omega_v} \nabla w_h \cdot \mathrm{Div}(\nabla\nabla v) \,\mathrm{d}\Omega\\ &= \int\limits_{\Gamma_v} (\nabla w_h \otimes \boldsymbol{N}) \cdot \nabla\nabla v \,\mathrm{d}\Gamma - \int\limits_{\Gamma_v} w_h [\boldsymbol{N} \cdot \mathrm{Div}(\nabla\nabla v)] \,\mathrm{d}\Gamma\\ &\quad + \int\limits_{\Omega_v} w_h \,\Delta^2 v \,\mathrm{d}\Omega\end{aligned} \tag{13.75}$$

and the second one leads to

$$\int\limits_{\Omega_v} \nabla w_h \,\mathrm{d}\Omega = \int\limits_{\Gamma_v} w_h \,\boldsymbol{N} \,\mathrm{d}\Omega\,. \tag{13.76}$$

All terms on the right hand side of the above equations are computable with ansatz functions that have the properties (a0)–(a5).

The left hand side in (13.74) needs the introduction of an ansatz for the projection w_π of order k

$$w_\pi = \langle\, 1 \; X \; Y \; \frac{1}{2}X^2 \; XY \; \frac{1}{2}Y^2 \; \dots \; \frac{1}{k(k-1)}Y^k \rangle \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ \dots \\ a_{\frac{1}{2}n(n+1)} \end{Bmatrix} = \mathbf{H}_w^{(2,k)}(X,Y)\,\mathbf{a} \tag{13.77}$$

which then can be inserted into (13.74). Due to the computability of the second derivative $\nabla\nabla w_\pi$ the ansatz must be at least of second order ($k=2$) which then results in an element with constant curvature. The same ansatz can be made for the weighting function v leading to $v = \mathbf{H}_w^{(2,k)}(X,Y)\,\mathbf{c}$ where $\mathbf{c}$ is a vector of constants that are eliminated like constants related to test functions.

13.3.3 Ansatz Function

In this section we will provide ansatz functions for two simple virtual plate elements, that have been discussed in Chinosi and Marini (2016), Wriggers et al. (2022). With the notation, introduced in Sect. 13.3.1, a cubic Hermite interpolation ($n = 3$) is selected for the deflection at the element edge Γ_e for both elements

$$(w_h)_e \; = H_1(\xi_e) w_1 + l_e H_1'(\xi_e)\theta_{1n} + H_2(\xi_e) w_2 + l_e H_2'(\xi_e)\theta_{2n} \tag{13.78}$$

where the basis functions are defined in terms of Hermite splines

$$\begin{aligned} H_1(\xi_e) &= 2\xi_e^3 - 3\xi_e^2 + 1\,, \; H_1'(\xi_e) = \xi_e^3 - 2\xi_e^2 + \xi_e\,, \\ H_2(\xi_e) &= -2\xi_e^3 + 3\xi_e^2\,, \quad H_2'(\xi_e) = \xi_e^3 - \xi_e^2\,. \end{aligned} \tag{13.79}$$

The derivative θ_n in (13.72) is then obtained by the scaled derivative of (13.79) with respect to ξ_e

$$(\theta_{hn})_e \; = \frac{1}{l_e}\frac{\mathrm{d}(w_h)_e}{\mathrm{d}\xi_e} \tag{13.80}$$

which has the explicit form

$$(\theta_{hn})_e \; = \frac{6}{l_e}(\xi_e^2 - \xi_e) w_1 + (3\xi_e^2 - 4\xi_e + 1)\theta_{1n} + \frac{6}{l_e}(-\xi_e^2 + \xi_e) w_2 + (3\xi_e^2 - 2\xi_e)\theta_{2n} \tag{13.81}$$

for both elements.

The tangential rotation $(\theta_{ht})_e$ at the edge Γ_e is defined by a linear ansatz ($s = 1$) for element 1 and a quadratic ansatz ($s = 2$) for element 2

$$\begin{aligned} &\text{element 1: } (\theta_{ht})_e = && (1 - \xi_e)\,\theta_{1t} + \xi_e\,\theta_{2t}\,, \\ &\text{element 2: } (\theta_{ht})_e = && (2\xi_e^2 - 3\xi_e + 1)\,\theta_{1t} + (2\xi_e^2 - 1)\,\theta_{2t} + 4(\xi_e - \xi_e^2)\,\theta_{3t}\,. \end{aligned} \tag{13.82}$$

Note that this selection corresponds to the property (a3) for the ansatz function $w_{,n}$ which is related to θ_t, see (13.72). This yields for the quadratic ansatz of element 2 with $s = 2$ one additional unknown θ_{3t} at the edge as demonstrated in Fig. 13.4.

13.3.4 Plate Element with Constant Curvature

The simplest possible C^1-continuous plate element is element 1. This element has only vertex nodes, see Fig. 13.5.

In element 1 the linear ansatz $(13.82)_1$ is selected for the rotation θ_t. This choice results in a second order scheme in which the projection w_π has to be a quadratic function which is defined at element level by

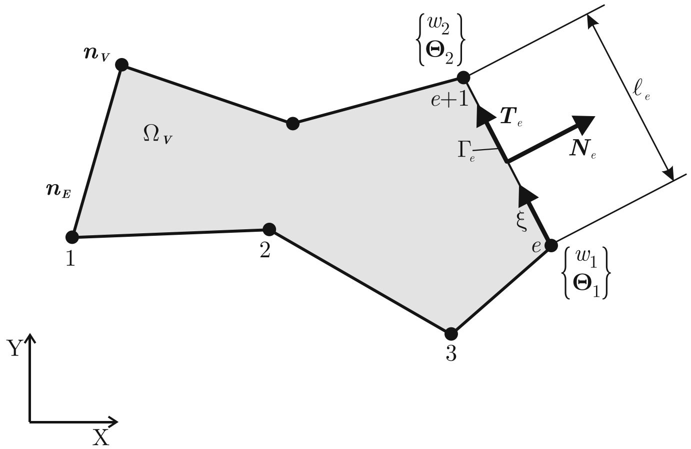

n_V
Ω_V
n_E
1
2
3
$e+1$
e
$\begin{Bmatrix} w_2 \\ \boldsymbol{\Theta}_2 \end{Bmatrix}$
$\begin{Bmatrix} w_1 \\ \boldsymbol{\Theta}_1 \end{Bmatrix}$
$\boldsymbol{T}_e$
$\boldsymbol{N}_e$
Γ_e
ξ
ℓ_e
Y
X

Fig. 13.5 Element with constant curvature for the Kirchhoff-Love plate

$$w_\pi = \mathbf{H}_w^{(2,2)}\,\mathbf{a} = \left\langle 1 \;\; X \;\; Y \;\; \tfrac{1}{2}X^2 \;\; XY \;\; \tfrac{1}{2}Y^2 \right\rangle \begin{Bmatrix} a_1 \\ a_2 \\ \dots \\ a_6 \end{Bmatrix} . \tag{13.83}$$

The unknown parameters a_i have to be linked to the element unknowns by the projection procedure (13.74). The polynomial N_π^n has to be selected in the same way as the ansatz for w_π using the quadratic function $\mathbf{H}^{(2,2)}$.

Now we can evaluate the first equation in (13.74). Noting that Div$(\nabla\nabla N_\pi^n)$ and $\Delta^2 N_\pi^n$ are zero for the ansatz (13.83) we arrive with (13.75) at

$$\int\limits_{\Omega_v} \nabla\nabla w_\pi \, \mathrm{d}\Omega = \frac{1}{2}\int_{\Gamma_v} (\nabla w_h \otimes \boldsymbol{N} + \boldsymbol{N} \otimes \nabla w_h) \; \mathrm{d}\Gamma \, . \tag{13.84}$$

since $\nabla\nabla N_\pi^n = const$ and can be eliminated from both sides of $(13.74)_1$. Note that the symmetry of the curvature tensor $\nabla\nabla w_\pi$ has been taken into account. By inserting the ansatz (13.83) into (13.84) this equation simplifies to

$$\Omega_v \; \nabla\nabla w_\pi|_v = \frac{1}{2}\int_{\Gamma_v} (\nabla w_h \otimes \boldsymbol{N} + \boldsymbol{N} \otimes \nabla w_h) \; \mathrm{d}\Gamma \tag{13.85}$$

with the constant curvature

$$\nabla\nabla w_\pi|_v = \begin{bmatrix} a_4 & a_5 \\ a_5 & a_6 \end{bmatrix} \tag{13.86}$$

at element level. The right hand side of (13.84) consists of the second order polynomial $w_h(\xi)_{,\xi}$ and the linear ansatz for $\theta_t(\xi)$ per segment Γ_e. It can be exactly integrated which results in a sum of all n_E edge contributions

$$\sum_{e=1}^{n_E} \int_0^1 \left[\left(\frac{1}{2l_e} w_h(\xi)_{,\xi} \; (\boldsymbol{T}_e \otimes \boldsymbol{N}_e + \boldsymbol{N}_e \otimes \boldsymbol{T}_e) - \theta_t(\xi)\, \boldsymbol{N}_e \otimes \boldsymbol{N}_e \right) l_e \; \mathrm{d}\xi \right]_e$$
$$= \sum_{e=1}^{n_E} \left[\frac{1}{2}(w_2 - w_1)(\boldsymbol{T}_e \otimes \boldsymbol{N}_e + \boldsymbol{N}_e \otimes \boldsymbol{T}_e) - l_e \frac{\theta_{t1} + \theta_{t2}}{2} \boldsymbol{N}_e \otimes \boldsymbol{N}_e \right]_e . \tag{13.87}$$

All terms in the sum are related to the edge Γ_e and thus values change from segment to segment.

By comparing (13.86) and (13.87) the unknowns a_4 to a_6 are obtained by inspection once all contributions related to the segments have been added. We observe that a_4 to a_6 are defined as a linear combination of the nodal values of w and θ_t. Thus (13.86) can be generally written in Voigt notation, see (13.64), as

$$\boldsymbol{\Upsilon}_\pi = \mathbb{B}_1 \, \mathbf{u}_1 \tag{13.88}$$

where $\mathbf{u}_1$ denotes the vector of all nodal unknowns[7]

$$\mathbf{u}_1 = \langle w_1 \, \theta_{X1} \, \theta_{Y1} \, w_2 \, \theta_{X2} \, \theta_{Y2} \; \ldots \; w_{n_V} \, \theta_{Xn_V} \, \theta_{Yn_V} \rangle^T . \tag{13.90}$$

Note that the matrix $\mathbb{B}_1$ is constant due to the quadratic ansatz for w_π.

The constants a_1 to a_3 follow from the two conditions (13.76) and (13.74)$_3$.

$$\int_{\Omega_v} \nabla w_\pi \, \mathrm{d}\Omega_v = \int_{\Gamma_v} w_h \boldsymbol{N} \, \mathrm{d}\Gamma \tag{13.91}$$

and

$$\int_{\Gamma_v} w_\pi \, \mathrm{d}\Gamma = \int_{\Gamma_v} w_h \; \mathrm{d}\Gamma . \tag{13.92}$$

The term on the left hand side in (13.91) yields with (13.83)

$$\int_{\Omega_v} \nabla w_\pi \, \mathrm{d}\Omega_v = \int_{\Omega_v} \begin{Bmatrix} a_2 + a_4 X + a_5 Y \\ a_3 + a_5 X + a_6 Y \end{Bmatrix} \mathrm{d}\Omega_v \tag{13.93}$$

[7] The global rotations (θ_X, θ_Y) are related to the local rotations (θ_n, θ_t) by the transformation

$$\begin{Bmatrix} \theta_X \\ \theta_Y \end{Bmatrix} = \begin{pmatrix} n_X & t_X \\ n_Y & t_Y \end{pmatrix}^T \begin{Bmatrix} \theta_n \\ \theta_t \end{Bmatrix} \tag{13.89}$$

using (13.71).

and involves only polynomials. The right hand side of (13.91) can be explicitly determined as

$$\int\limits_{\Gamma_v} w_h \boldsymbol{N}\, \mathrm{d}\Gamma = \sum_{e=1}^{n_E} \int\limits_0^1 [(w_1 H_1(\xi) + l_e \theta_{1n} H_1'(\xi) + w_2 H_2(\xi) + l_e \theta_{2n} H_2'(\xi))\, \mathrm{d}\xi]_e l_e\, \boldsymbol{N}_e$$
$$= \sum_{e=1}^{n_E} \left[\frac{1}{2}(w_1 + w_2) + \frac{l_e}{12}(\theta_{1n} - \theta_{2n}) \right]_e \, l_e\, \boldsymbol{N}_e \tag{13.94}$$

Now Eqs. (13.93) and (13.91) can be combined with (13.94) which leads to

$$\begin{Bmatrix} a_2 \\ a_3 \end{Bmatrix} = \frac{1}{\Omega_v} \left[\sum_{e=1}^{n_E} \left[\frac{1}{2}(w_1 + w_2) + \frac{l_e}{12}(\theta_{1n} - \theta_{2n}) \right]_e l_e\, \boldsymbol{N}_e - \int\limits_{\Omega_v} \begin{Bmatrix} a_4 X + a_5 Y \\ a_5 X + a_6 Y \end{Bmatrix} \mathrm{d}\Omega_v \right] \tag{13.95}$$

Here the last term can be exactly integrated over the boundary using relations (A.22) and (A.24) or the explicit sums (A.2)–(A.4). In the same way as Eq. (13.87) formula (13.95) leads to a linear mapping between a_2, a_3 and the element unknowns for w and θ_n.

The last term that has to be connected to the nodal values of the ansatz is the coefficient a_1. Using Eq. (13.92) the coefficient a_1 follows directly. With (13.83) and (13.92) one writes

$$\int\limits_{\Gamma_v} (a_1 + a_2 X + a_3 Y + \frac{a_4}{2} X^2 + a_5 XY + \frac{a_6}{2} Y^2)\, \mathrm{d}\Gamma = \sum_{e=1}^{n_E} \left[\frac{w_1 + w_2}{2} + \frac{l_e}{12}(\theta_{1n} - \theta_{2n}) \right]_e l_e \tag{13.96}$$

which yields a_1 immediately since a_2 to a_6 are already known from the relations (13.87) and (13.95).

With (13.87), (13.95) and (13.96) the L^2 projection w_π is completely defined for the virtual element Ω_v in terms of the nodal unknowns. Thus the linear mapping

$$w_\pi = \mathbb{M}_1(X, Y)\, \mathbf{u}_1 \tag{13.97}$$

can be introduced which provides the projection w_π in terms of the $3\, n_V$ nodal degrees of freedom (13.90) of the plate element.

Matrix forms for these relations were derived in e.g. Sect. 3.1.6 for solid elements. Here no explicit form is provided. This is mainly due to the fact that the software *AceGen*, see e.g. Korelc and Wriggers (2016), was employed to automatically derive and code[8] the residual vector and stiffness matrix associated with element 1.

[8] For a different implementation using *MATLAB* we refer to Yu (2022) who included the source code for a plate element in the paper.

13.3.5 Plate Element with Linear Curvature

The L^2 projector for element 2 relies on the ansatz for w and θ_t in (13.79) and $(13.82)_2$ element 2 which yields a virtual plate element of order 3. Thus the projection w_π has to be cubic function in x and y. It is defined at element level by

$$w_\pi = \mathbf{H}_w^{(2,3)}\,\mathbf{a} = \left\langle 1\;\, X\;\, Y\;\, \tfrac{1}{2}X^2\;\, XY\;\, \tfrac{1}{2}Y^2\;\, \tfrac{1}{6}X^3\;\, \tfrac{1}{2}X^2Y\;\, \tfrac{1}{2}XY^2\;\, \tfrac{1}{6}Y^3 \right\rangle \begin{Bmatrix} a_1 \\ a_2 \\ \dots \\ a_{10} \end{Bmatrix} \qquad (13.98)$$

Using Voigt notation the gradient of w_π is given by

$$\nabla w_\pi = \begin{Bmatrix} w_{\pi,x} \\ w_{\pi,y} \end{Bmatrix} = \nabla \mathbf{H}_w^{(2,3)}\,\mathbf{a} = \begin{bmatrix} 0 & 1 & 0 & X & Y & 0 & \frac{1}{2}X^2 & XY & \frac{1}{2}Y^2 & 0 \\ 0 & 0 & 1 & 0 & X & Y & 0 & \frac{1}{2}X^2 & XY & \frac{1}{2}Y^2 \end{bmatrix} \mathbf{a}\,. \qquad (13.99)$$

In the same way the curvature follows with $w_{\pi,xy} = w_{\pi,yx}$ as

$$\nabla\nabla w_\pi = \begin{Bmatrix} w_{\pi,xx} \\ w_{\pi,yy} \\ 2\,w_{\pi,xy} \end{Bmatrix} = \nabla\nabla \mathbf{H}_w^{(2,3)}\,\mathbf{a}_\kappa = \begin{bmatrix} 1 & 0 & 0 & X & Y & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & X & Y \\ 0 & 2 & 0 & 0 & 2X & 2Y & 0 \end{bmatrix} \mathbf{a}_\kappa\,. \qquad (13.100)$$

which is a linear function in X and Y. The matrix was shortened to non zero entries. Only the unknown coefficients $\mathbf{a}_\kappa = \{a_4\,\dots\,a_{10}\}$ affect the curvature since the first three columns in $\nabla \mathbf{H}_w^{(2,3)}$, see (13.99), would lead to zero values in (13.100).

The ansatz for the weighting function v results directly from (13.98) and its gradients follow from (13.99) and (13.100) as

$$v = \mathbf{H}_w^{(2,3)}\,\mathbf{c}\,, \quad \nabla v = \nabla \mathbf{H}_w^{(2,3)}\,\mathbf{c} \text{ and } \nabla\nabla v = \nabla\nabla \mathbf{H}_w^{(2,3)}\,\mathbf{c}_\kappa \qquad (13.101)$$

Let us consider again the L^2 projection (13.74) together with (13.75). For the cubic ansatz the relation

$$\int\limits_{\Omega_v} \nabla\nabla w_\pi \cdot \nabla\nabla v \,d\Omega = \int\limits_{\Gamma_v} \nabla w_h \cdot [\nabla\nabla v\, \boldsymbol{N}]\,d\Gamma - \int\limits_{\Gamma_v} w_h [\boldsymbol{N} \cdot \text{Div}(\nabla\nabla v)]\,d\Gamma \qquad (13.102)$$

can be used to obtain a_4 to a_{10} in terms of the nodal unknowns. Note that the term $\Delta^2 N_\pi^n$ in (13.75) is zero again for the ansatz (13.98) . The integral on the left side yields with (13.100)

$$\int\limits_{\Omega_v} \nabla\nabla w_\pi \cdot \nabla\nabla v \,d\Omega = \mathbf{c}_\kappa^T \int\limits_{\Omega_v} [\nabla\nabla \mathbf{H}_w^{(2,3)}]^T \nabla\nabla \mathbf{H}_w^{(2,3)}\,d\Omega\,\mathbf{a}_\kappa \qquad (13.103)$$

which depends only on polynomials up to order 2. Hence the integral can be evaluated directly using (A.2)–(A.7). However it can also be transformed in the same way as the right hand side to a boundary integral and then integrated. The choice depends on the available software and the most efficient approach. Note that the matrix inside the integral has the size (7 × 7) due to the reduction to the unknowns a_4 to a_{10}.

After some algebra the right hand side can be computed with

$$\nabla\nabla v\, \boldsymbol{N} = \nabla\nabla(\mathbf{H}_w^{(2,3)})_N\, \mathbf{c}_\kappa = \begin{bmatrix} n_x & n_y & 0 & n_x X & (n_y X + n_x Y) & n_y Y & 0 \\ 0 & n_x & n_y & 0 & n_x X & (n_y X + n_x Y) & n_y Y \end{bmatrix} \mathbf{c}_\kappa \tag{13.104}$$

and the constant term

$$\boldsymbol{N} \cdot \mathrm{Div}(\nabla\nabla v) = \nabla\nabla(\mathbf{H}_w^{(2,3)})_D\, \mathbf{c}_\kappa = \begin{bmatrix} 0 & 0 & 0 & n_x & n_y & n_x & n_y \end{bmatrix} \mathbf{c}_\kappa \tag{13.105}$$

as

$$\int\limits_{\Gamma_v} \nabla w_h \cdot [\nabla\nabla v\, \boldsymbol{N}]\, \mathrm{d}\Gamma = \mathbf{c}_\kappa^T \int\limits_{\Gamma_v} [\nabla\nabla(\mathbf{H}_w^{(2,3)})_N]^T \{\nabla w_h\}\, \mathrm{d}\Gamma \tag{13.106}$$

$$\int\limits_{\Gamma_v} w_h [\boldsymbol{N} \cdot \mathrm{Div}(\nabla\nabla v)]\, \mathrm{d}\Gamma = \mathbf{c}_\kappa^T [\nabla\nabla(\mathbf{H}_w^{(2,3)})_D]^T \int\limits_{\Gamma_v} w_h\, \mathrm{d}\Gamma \tag{13.107}$$

which leads to the equation system for the unknown coefficients $\mathbf{a}_\kappa$ of w_π in terms of the nodal degrees of freedom

$$\begin{aligned} &\int\limits_{\Omega_v} [\nabla\nabla\mathbf{H}_w^{(2,3)}]^T \nabla\nabla\mathbf{H}_w^{(2,3)}\, \mathrm{d}\Omega\; \mathbf{a}_\kappa = \\ &\quad \int\limits_{\Gamma_v} [\nabla\nabla(\mathbf{H}_w^{(2,3)})_N]^T \{\nabla w_h\}\, \mathrm{d}\Gamma - [\nabla\nabla(\mathbf{H}_w^{(2,3)})_D]^T \int\limits_{\Gamma_v} w_h\, \mathrm{d}\Gamma \end{aligned} \tag{13.108}$$

The first integral of the right hand side can be evaluated analogously to (13.95). The only difference is that now some integral terms occur where the coordinates X and Y appear. The second integral is exactly the same as the right hand side in (13.96). Equation 13.108 can now be solved which yields the unknown parameters $\mathbf{a}_\kappa$ in terms of the nodal degrees of freedom $\mathbf{u}_2$, in short $\mathbf{a}_\kappa = \mathbb{P}_\kappa\, \mathbf{u}_2$. Here, the vector $\mathbf{u}_2$ contains all nodal unknowns of element 2

$$\mathbf{u}_2 = \langle w_1\, \theta_{X1}\, \theta_{Y1}\, \theta_{t1}\, w_2\, \theta_{X2}\, \theta_{Y2}\, \theta_{t2}\, \ldots\, w_{n_V}\, \theta_{Xn_V}\, \theta_{Yn_V}\, \rangle^T . \tag{13.109}$$

Again, the local rotations associated with specific edges can be transformed to global rotations using (13.89).

In Voigt notation, see (13.64), the curvature $\boldsymbol{\Upsilon}_\pi$ of the plate element follows by inserting $\mathbf{a}_\kappa$ into (13.100) which leads to the form

$$\boldsymbol{\Upsilon}_\pi = \mathbb{B}_2(X,Y)\,\mathbf{u}_2 \qquad \text{with } \mathbb{B}_2(X,Y) = \nabla\nabla\mathbf{H}_w^{(2,3)}(X,Y)\,\mathbb{P}_\kappa \tag{13.110}$$

which is a linear function in X and Y.

Equation system (13.108) has to be complemented by the conditions (13.91) and (13.92). The first one yields after inserting the gradient (13.99)

$$\int\limits_{\Omega_v} \nabla\mathbf{H}_w^{(2,3)}\,\mathrm{d}\Omega_v\,\mathbf{a} = \sum_{e=1}^{n_E}\left[\frac{1}{2}(w_1+w_2) + \frac{l_e}{12}(\theta_{1n}-\theta_{2n})\right]_k l_e\,\boldsymbol{N}_e\,. \tag{13.111}$$

Here the left hand side can be simply integrated using relations (A.2)–(A.7). The right hand side of (13.91) has the same form as (13.94).

Finally the coefficient a_1 can be determined from (13.92)

$$\int\limits_{\Gamma_v} \mathbf{H}_w^{(2,3)}\,\mathrm{d}\Gamma\,\mathbf{a} = \sum_{e=1}^{n_E}\left[\frac{w_1+w_2}{2} + \frac{l_e}{12}(\theta_{1n}-\theta_{2n})\right]_e l_e \tag{13.112}$$

where the ansatz function (13.78) is inserted on the right side which yields exactly the result reported in (13.96).

The three Eqs. (13.108), (13.111) and (13.112) can be solved successively, starting (13.108). It is also possible to put the equations into one equation system which has the matrix form

$$\left[\begin{array}{c} \int\limits_{\Gamma_v} \mathbf{H}_w^{(2,3)}\,\mathrm{d}\Gamma \\ \hdashline \int\limits_{\Omega_v} \nabla\mathbf{H}_w^{(2,3)}\,\mathrm{d}\Omega_v \\ \hdashline \int\limits_{\Omega_v} [\nabla\nabla\mathbf{H}_w^{(2,3)}]^T\,\nabla\nabla\mathbf{H}_w^{(2,3)}\,\mathrm{d}\Omega \end{array}\right]\mathbf{a} = \left\{\begin{array}{c} \int\limits_{\Gamma_v} w_h\,\mathrm{d}\Gamma \\ \hdashline \int\limits_{\Gamma_v} w_h\boldsymbol{N}\,\mathrm{d}\Gamma \\ \hdashline \int\limits_{\Gamma_v} [\nabla\nabla(\mathbf{H}_w^{(2,3)})_N]^T\{\nabla w_h\}\,\mathrm{d}\Gamma \\ -[\nabla\nabla(\mathbf{H}_w^{(2,3)})_D]^T \int\limits_{\Gamma_v} w_h\,\mathrm{d}\Gamma \end{array}\right\} \tag{13.113}$$

and can be solved for the unknown coefficients $\mathbf{a}$. Inserting these coefficients now into the ansatz w_π, see (13.98), leads to the mapping

$$w_\pi = \mathbb{M}_2(X,Y)\,\mathbf{u}_2 \tag{13.114}$$

which describes w_π in terms of the vector $\mathbf{u}_2$, see (13.109), containing all nodal unknowns of element 2.

13.3.6 Residual and Stiffness Matrix of the Virtual Plate Element

As in the previous chapters, the virtual element is derived from the total energy. Two contributions have to be considered. These are the consistency part that stems from inserting the ansatz function (13.97) and (13.114) into the element and a stabilization part that is needed to obtain the proper rank for the resulting stiffness matrix.

We note, that element 1 has 9 degrees of freedom (DOFs) and constant curvature with three independent components $c_{ans} \in \{w_{xx}\,, w_{xy}\,, w_{yy}\}$, see (13.86). This leads to a stiffness matrix, related to the consistency part, that has a rank deficiency of

$$r_d^1 = DOFs^1 - c_{ans}^1 - RGB = 9 - 3 - 3 = 3$$

including the three-rigid body modes (RGB). The rank deficiency of element 2 with 12 degrees of freedom and cubic ansatz for w_π is given by

$$r_d^2 = DOFs^2 - c_{ans}^2 - RGB = 12 - 7 - 3 = 2$$

due to the 7 independent components related to the curvature in (13.100).[9] Thus even the simplest possible element (triangle with three vertices, $n_V = 3$) needs to be stabilized, hence all other elements with $n_V > 3$ are rank deficient as well and a stabilization is inevitable.

By assembling all element contributions for n_T virtual elements the total energy is given by

$$U(w_h) = \mathbf{A}_{v=1}^{n_T} \left[U_c^v(w_\pi) + U_{stab}^v(w_h - w_\pi) \right] . \tag{13.116}$$

Part of the virtual element due to projection (consistency term). With the strain $\boldsymbol{\Upsilon}$ of the plate that depends on the projection w_π and the loading $\bar{q}$ we obtain the potential energy of an element

$$U_c^v(w_\pi) = \int\limits_{\Omega_v} \left[\frac{1}{2} [\boldsymbol{\Upsilon}(w_\pi)]^T \mathbb{D}\, \boldsymbol{\Upsilon}(w_\pi) - \bar{q}\, w_\pi \right] \mathrm{d}\Omega \tag{13.117}$$

with the constitutive tensor $\mathbb{D}$ for the isotropic and $\mathbb{D}_G$for the anisotropic case.

The curvature $\boldsymbol{\Upsilon}$ is constant for element 1 which leads with (13.88) to the trivial evaluation of the first term

[9] In general, the rank deficiency of a plate element can be computed by the formula

$$r_d = n_E\,(r + s - 1) - \frac{(n+1)(n+2)}{2} \tag{13.115}$$

where n_E are the number of edges, r is the polynomial degree of the ansatz w_h, s is the polynomial degree of the ansatz for $w_{h\,,n}$ and n is the ansatzorder of w_π, see also (13.73).

$$\int\limits_{\Omega_v} \frac{1}{2}[\boldsymbol{\Upsilon}(w_\pi)]^T \mathbb{D}\, \boldsymbol{\Upsilon}(w_\pi)\, \mathrm{d}\Omega = \frac{\Omega}{2}\mathbf{u}_1^T\, \mathbb{B}_1^T \mathbb{D}\, \mathbb{B}_1\, \mathbf{u}_1 \tag{13.118}$$

For element 2 the strain $\boldsymbol{\Upsilon}$ is linear in X and Y, hence the integration involves terms up to second order in the coordinates which again can be performed over the boundary using (A.22) and (A.24)

$$\int\limits_{\Omega_v} \frac{1}{2}[\boldsymbol{\Upsilon}(w_\pi)]^T \mathbb{D}\, \boldsymbol{\Upsilon}(w_\pi)\, \mathrm{d}\Omega = \frac{1}{2}\mathbf{u}_2^T \int\limits_{\Omega_v} \mathbb{B}_2^T(X,Y)\mathbb{D}\, \mathbb{B}_2(X,Y)\, \mathrm{d}\Omega\, \mathbf{u}_2\,. \tag{13.119}$$

The projection w_π is known in terms of the unknowns $(w\,,\theta_X\,,\theta_Y)_k$ at the vertices of element 1 and 2 and the unknown rotation θ_t at the mid node of element 2, see (13.97) and (13.114). The mappings $w_\pi = \mathbb{M}_{el}(X,Y)\,\mathbf{u}_{el}$ can be employed in (13.117) to compute the loading part for elements $el = 1, 2$

$$\int\limits_{\Omega_v} \bar{q}\, w_\pi\, \mathrm{d}\Omega = \int\limits_{\Omega_v} \bar{q}\, \mathbb{M}_{el}(X,Y)\, \mathrm{d}\Omega\, \mathbf{u}_{el}\,. \tag{13.120}$$

Once the integration over the element area is carried out the potential is just a function of the unknowns of the element. This fact can be used to compute the element residual and tangent by differentiation with respect to the nodal unknowns.

Stabilization part. Within the virtual element method, the rank deficient consistency part of the element has to be stabilized. See for weak formulations of virtual plate elements Brezzi and Marini (2013), Mora et al. (2018). This approach leads to a stabilization operator for the potential energy which addresses the error at all nodal points n_N (vertices and midpoints):

$$U^v_{stab}(w_h - w_\pi) = \frac{D_b}{2\,h_v^2}\sum_{k=1}^{n_N}\left[\widehat{w}(\mathbf{X}_k)^2 + \left\|\frac{l_{k-1}+l_k}{2}\nabla\widehat{w}(\mathbf{X}_k)\right\|^2 + \left\|l_k\,\widehat{\theta}_t(\mathbf{X}_k)\right\|^2\right] \tag{13.121}$$

$$\begin{aligned}
\text{with } \widehat{w}(\mathbf{X}_k) &= w_h(\mathbf{X}_k) - w_\pi(\mathbf{X}_k)\,,\\
\nabla\widehat{w}(\mathbf{X}_k) &= \nabla w_h(\mathbf{X}_k) - \nabla w_\pi(\mathbf{X}_k) \quad \text{and}\\
\widehat{\theta}_t(\mathbf{X}_k) &= \begin{cases} \theta_{t\,h}(\mathbf{X}_k) - \theta_{t\,\pi}(\mathbf{X}_k) & \text{for } el = 2\\ 0 & \text{for } el = 1\,.\end{cases}
\end{aligned}$$

Here, D_b is the bending stiffness, see (13.59), and h_v is the maximum diameter of the virtual element v thus h_v^2 can be interpreted as the element area Ω_v. The rotation at each vertex node is scaled by the mean value of the lengths l_k and l_{k-1} of the adjacent segments. The function w_h and the projection w_π have to be evaluated at the vertices $\mathbf{X}_k = \{X_k, Y_k\}$.

A different stabilization energy is provided by an extension of (13.121) which describes the total error on the edge in a mean way by an integral instead of using discrete values, see Wriggers et al. (2022). Thus an integration has to be performed over the edge which is described by the coordinate $\boldsymbol{X}_e$. This new stabilization takes the distribution of each degree of freedom along the edge into account

$$U^v_{stab}(w_h - w_\pi) = \frac{D_b}{2\,\Omega_v} \sum_{e=1}^{n_E} \frac{1}{l_e} \int\limits_{\Gamma_e} \left[\widehat{w}(\boldsymbol{X}_e)^2 + \| l_e \nabla \widehat{w}(\boldsymbol{X}_e) \|^2 \right] \mathrm{d}\Gamma\,. \tag{13.122}$$

Again h_v^2 can be used in this equation instead of Ω_v. The edge integral in (13.122) is evaluated numerically using Gauss quadrature

$$\int\limits_{\Gamma_e} f(\boldsymbol{x})\mathrm{d}\Gamma = \int\limits_0^1 f(\boldsymbol{x}) J_\xi \mathrm{d}\xi = l_e \sum_{g=1}^{n_g} w_g\, f(\mathbf{x}_g) \tag{13.123}$$

where the Jacobian J_ξ in the transformation $\mathrm{d}\Gamma = J_\xi \mathrm{d}\xi$ is the length of the eth edge $J_\xi = \| \frac{\partial \boldsymbol{X}_e}{\partial \xi} \| = l_e$, $\xi \in [0, 1]$ is the local coordinate and $\mathbf{x}_g$ is the Gauss point.

When employing stabilizations (13.121) and (13.122) the resulting tangent matrices have full rank. Note, that implementing stabilization (13.121) leads to a more efficient code at element level.

In Sect. 6.1.3 the energy stabilization was introduced which is based on the formulation of a stabilization energy that has to be evaluated using a finite element submesh inside a virtual element. Such approach is also possible for the plate element. However it is not straight forward to select simple finite elements with the same nodal degrees of freedom that can be inserted. In Wriggers et al. (2022) the Morley, see Morley (1968), and the DKT plate element, see Batoz et al. (1980), were employed to stabilize the consistency part. The results show that such stabilization impairs the order of convergence due to a reduced convergence order of the finite elements when compared to the virtual plate elements. Hence the stabilization, discussed above, is optimal for virtual plate elements.

Residual and tangent matrix. Once the formulation for the consistency (13.120) and stabilization (13.121) part of the potential is completed, these are function of the element unknowns: $U^v_c(w_\pi) = U^v_c[\mathbf{u}_v]$ and $U^v_{stab}(w_h - w_\pi) = U^v_{stab}[\mathbf{u}_v]$. Then residual vector $\mathbf{R}_v$ and tangent (stiffness) matrix $\mathbf{K}_v$ follow from

$$\mathbf{R}_v = \frac{\partial (U^v_c[\mathbf{u}_v] + U^v_{stab}[\mathbf{u}_v])}{\partial \mathbf{u}_v} \quad \text{and } \mathbf{K}_v = \frac{\partial \mathbf{R}_v}{\partial \mathbf{u}_v}\,. \tag{13.124}$$

for the virtual plate element v.

13.4 Numerical Examples

In this section we first study the behaviour of the virtual plate elements 1 and 2 and the effect of the two stabilizations. The performance of the elements will be illustrated by means of numerical examples that are classical benchmark problems and also related to engineering applications. All numerical solutions are compared with analytical solutions when available. The range of problems includes isotropic material as well as anisotropic materials. Some of the examples can be found in Wriggers et al. (2022).

13.4.1 Notation Used in the Examples

The following notation is introduced to distinguish the different versions of the virtual plate elements 1 and 2:

- VE-1: first order VEM formulation. Degrees of freedom (DOF) per vertex are: $(w_k, \boldsymbol{\theta}_k)$, see Sect. 13.3.4,
- VE-2: second order VEM formulation. DOFs per vertex are: $(w_k, \boldsymbol{\theta}_k)$ and per element edge $\theta_{t\,e}$, see Sect. 13.3.5.

For comparison purposes the DKT plate element will be used which is known in the engineering community as a well performing element, see Batoz et al. (1980). Its DOFs per vertex are: $(w_k, \boldsymbol{\theta}_k)$. The element will be denoted by FE-DKT.

The following types of stabilization will be employed for the virtual plate elements:

- st-1a/b: nodal error stabilization (13.121), where “a” refers to using the element size h_v^2 and “b” to employing the element area Ω_v in the scaling factor.
- st-2a/b: edge-integrated error stabilization (13.122) where “a” and “b” have the same meaning as above.

Virtual plate elements can have arbitrary number of nodes, starting from triangular shape via quadrilateral shape to general non-convex forms. Here we will employ the following types of elements and meshes

- T1: triangle mesh with 3 edges per element,
- Q1: quadrilateral mesh with 4 edges per element,
- VO-U: Voronoi-type mesh, with uniformly distributed cell seeds,
- VO-R: Voronoi-type mesh, with randomly distributed cell seeds.

A T1 mesh consists of virtual elements with triangular shape having 9 unknowns (element 1) or 12 unknowns (element 2) per element. Q1 meshes are composed of quadrilateral virtual elements with four edges and 12 unknowns (element 1) or 16 unknowns (elements 2) per element. The number of nodes per element depends in a Voronoi type mesh on the shape of the Voronoi cells. In our examples the number of edges in a Voronoi-type mesh varies from three to ten.

13.4.2 Clamped Plate Under Uniform Load

The general convergence behaviour of element 1 and 2 is reported for a clamped plate. Furthermore, the impact of the different stabilizations is discussed. The clamped plate under uniform load has an analytical solution, see e.g. Taylor and Govindjee (2004), which allows an exhaustive convergence analysis. The clamped square plate has size $B \times H = 8\,\mathrm{m} \times 8\,\mathrm{m}$ and is subjected to a uniform load $\bar{q}$. Due to symmetry only one quarter of the problem has to be modeled, see Fig. 13.6.

The material parameters are provided in Table 13.1.

The analytical results of the clamped plate problem are summarized in Table 13.2 for the data provided in Table 13.1. Table 13.2 depicts the total strain energy W_{ref}, the mid deflection w_{ref} as well as minimum, $M_{\mathrm{ref}}^{\mathrm{min}}$, and maximum, $M_{\mathrm{ref}}^{\mathrm{max}}$, bending moments of the plate up to ten digits.

Figure 13.7a, b show the discretisation of the plate with square and Voronoi elements. The convergence study for virtual element 1 (VE-1) using the two types of classical stabilization, is shown for square elements in Fig. 13.8a. The same study is performed for virtual element 2 (VE-2), see Fig. 13.8b. For regular and unstructured Voronoi meshes the convergence results are illustrated in Fig. 13.9. The convergence study is performed with respect to the deflection at the center of the plate and the energy using the analytical values w_{ref} and W_{ref} from Table 13.2 as reference. Furthermore, the results of the DKT element (FE-DKT) which is know to be one of the best finite plate elements are included for comparison.

Fig. 13.6 Boundary value problem of the clamped plate

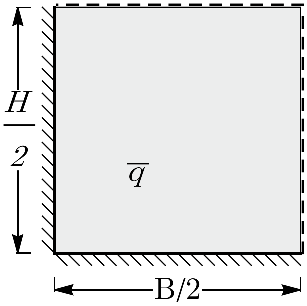
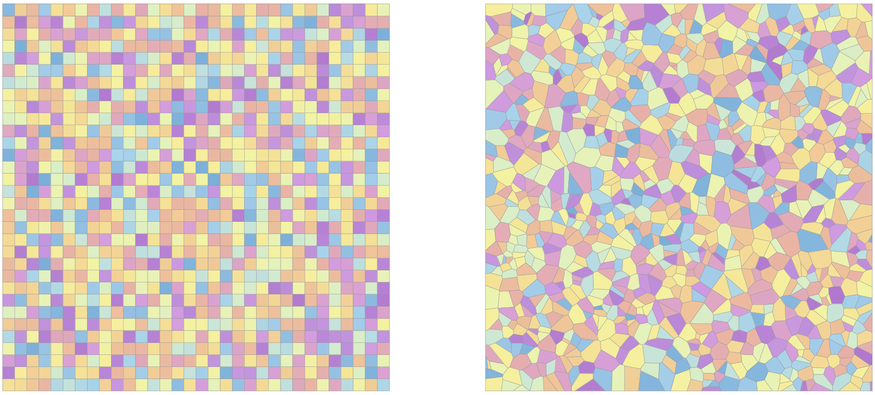


Table 13.1 Material parameters

Description	Symbol	Value	Unit
Young's modulus	E	2×10^8	$\frac{kN}{\mathrm{m}^2}$
Poisson ratio	ν	0.3	
Thickness	t	0.01	m
Uniform surface load	$\bar{q}$	-1.0	$\frac{kN}{\mathrm{m}^2}$
Bending stiffness	D_b (13.59)	18.315	Nm

Table 13.2 Analytical solutions, see Taylor and Govindjee (2004)

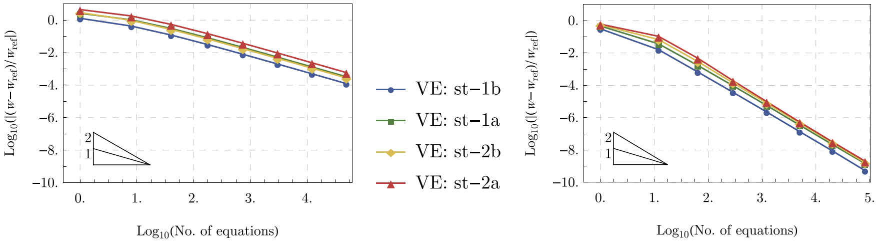
Description	Symbol	Formula	Value	Unit
Energy	W_{ref}	$3.891200775 \times 10^{-4} q^2 B^3 H^3 / D_b$	5.569	kJ
Deflection	w_{ref}	$1.265319087 \times 10^{-3} q\, B^2 H^2 / D_b$	−0.283	m
Min. Moment	$M_{\text{ref}}^{\text{min}}$	$2.290509078 \times 10^{-2} q\; B\; H$	−1.466	kN m
Max. Moment	$M_{\text{ref}}^{\text{max}}$	$-5.13337648 \times 10^{-2} q\; B\; H$	3.285	kN m



(a) Q1 square

(b) VO-R Voronoi

Fig. 13.7 Different meshes: quadrilateral Q1 and random-seed Voronoi mesh



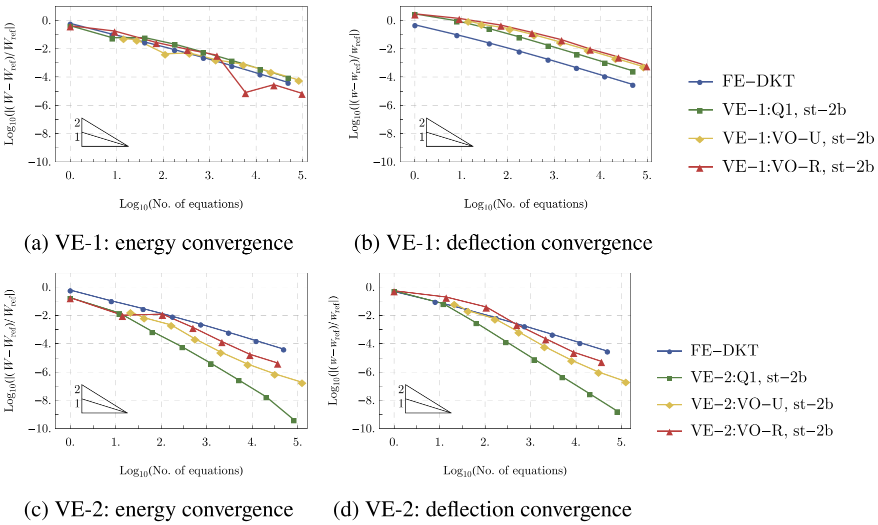
(a) VE-1: deflection convergence

(b) VE-2: deflection convergence

Fig. 13.8 Convergence study of deflection w for square Q1 meshes

All results demonstrate that the choice of stabilization does not influence the asymptotic convergence behaviour since the slopes of the curves are the same. However there is an influence regarding the coarse mesh accuracy and the type of element. The order of convergence is 1 for VE-1 while VE-2 demonstrates an order of convergence of 2, see Fig. 13.8.

In this section only the behaviour of VE-1 and VE-2 when using different stabilizations is investigated for quadrilateral, regular and unstructured Voronoi meshes



(a) VE-1: energy convergence (b) VE-1: deflection convergence

(c) VE-2: energy convergence (d) VE-2: deflection convergence

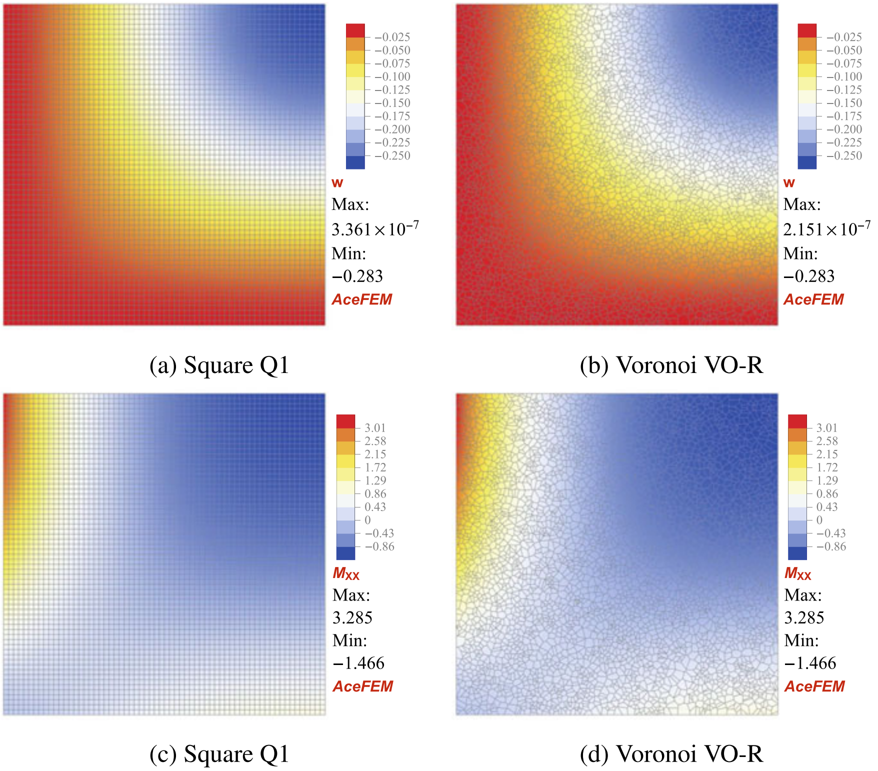
Fig. 13.9 Convergence study of energy W and deflection w for different forms of the elements (structured and unstructured meshes)

depicted in Fig. 13.7. The influence of the stabilization when using virtual elements with triangular shape is further discussed in Sect. 13.5.

The effect between stabilization type "a" and "b" is related to the normalisation with the element size h_v^2 (st-□a) or the area Ω_v (st-□b). The difference between both scalings is negligible. But generally, using the area yields slightly better results in all cases.

There is a slight difference in stabilizations "1" and "2". For Q1 meshes stabilization "1" is superior, but for Voronoi meshes, stabilization "2" is better by a small amount. It seems that the best stabilization depends on the mesh type. As expected, structured meshes perform better than unstructured Voronoi type meshes. In this example the Q1 meshes yield the best results but also the Voronoi type meshes yield solutions that have a high accuracy, Hence the virtual plate elements can be employed safely for the solution of engineering structures. The use of Q1-meshes will be explored further in Sect. 13.5.

Figure 13.10 depicts contour plots of the deflection w and the bending moment M_{XX}, for the virtual element VE-2:st-1b. The contour plots of the structured meshes as well as the unstructured Voronoi meshes report minimum and maximum bending moments (max $M_{XX} = 3.285$ and min $M_{XX} = -1.466$) that match exactly the analytical results in Table 13.2. But element VE-1 and all other investigated mesh types produce equally good results and demonstrate that the developed virtual plate elements are capable of computing meaningful engineering solutions.



(a) Square Q1

(b) Voronoi VO-R

(c) Square Q1

(d) Voronoi VO-R

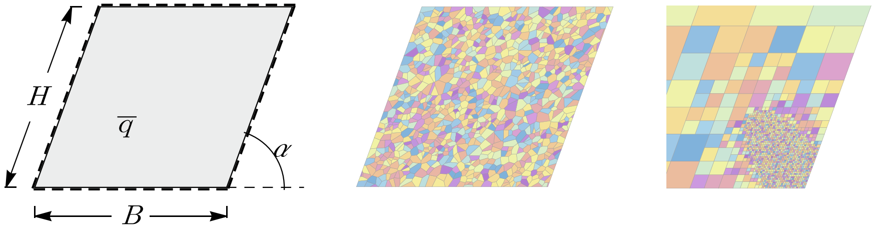
Fig. 13.10 Deflection w (**a**, **b**) and Moments M_{XX} (**c**, **d**) for structured and unstructured meshes, note that M_{YY} is same but flipped over the domain diagonal

13.4.3 Rhombic Plate

The solution of Kirchhoff-Love plates can include singularities. Unlike the Reissner-Mindlin plates where singularities are associated with specific loading conditions, e.g. point loads, and certain geometrical conditions, e.g. reentrant corners, Kirchhoff-Love plates show singular behaviour only for particular geometries like obtuse and reentrant corners.

The simply supported rhombic plate, see Fig. 13.11a, depicts such singular behaviour. The rhombic plate is subjected to a uniform load. Its dimensions are $H = B = 8$ m. The angle α in Fig. 13.11a is selected as $\alpha = 30^{\circ}$ which creates an internal, obtuse angle of 150°. The material parameters can be found in Table 13.1.

Near the obtuse angles the solution is singular, belonging to $H^{\gamma-\epsilon}$, $\forall\epsilon > 0$ with $\gamma = 2 - \frac{\alpha}{\pi-\alpha}$. Therefore the asymptotic convergence rate for uniform meshes is given by $\frac{\alpha}{\pi-\alpha} = 0.2$ and the solution is governed by the singularity, see e.g. Babuska and Scapolla (1989).

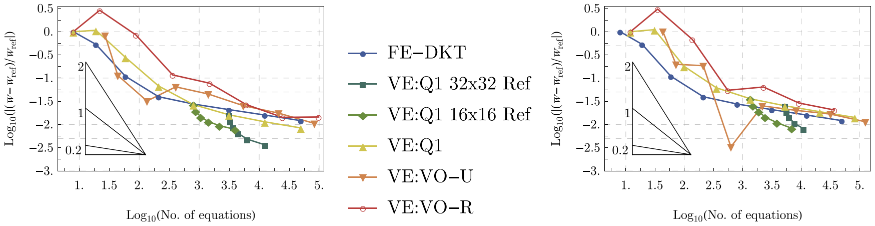


(a) Simply supported

(b) VO-R mesh

(c) Q1 adaptive refined mesh

Fig. 13.11 Rhombic plate geometry and meshes



(a) VE-1: deflection convergence

(b) VE-2: deflection convergence

Fig. 13.12 Rhombic plate with $\alpha = 30°$: Convergence study of the deflection w

In such cases it is well known that an adaptive refinement leads to an improvement of the asymptotic convergence, with the possibility to recover the optimal convergence rate, see Babuska and Scapolla (1989). Since a virtual plate element can have an arbitrary number of nodes, it is relatively simple to adaptively refine a regular mesh, see Fig. 13.11b, in a non uniform manner, see Fig. 13.11c. This allows the recovery of a better convergence rate. In this example the refinement is based on bisecting h_v as demonstrated in Fig. 13.11c. Only new nodes have to be added at the middle of the edge of an element since the number of nodes can be arbitrary and still yield a C^1-continuous mesh avoiding hanging nodes that appear when a finite element mesh is refined in the same manner. Five refinement steps were executed towards the obtuse corner to compute the solutions in Fig. 13.12 for elements 1 and 2 st-2b using regular, Voronoi, Fig. 13.7b, and adaptively refined meshes, Fig. 13.11c. The adaptive local refinement increases the convergence rate drastically when compared to a mesh refined uniformly by a factor of 2 as shown in Fig. 13.12. The simple bisecting refinement however is not sufficient to recover the optimal convergence rate which would need a node insertion as shown in Wang et al. (1984). But the results show clearly the sensitivity of the solution with respect to the mesh refinement at the singularity.

Here the criteria for adaptive refinement is based on the difference of the energy between element nodes. To keep this difference small, elements at the obtuse corners

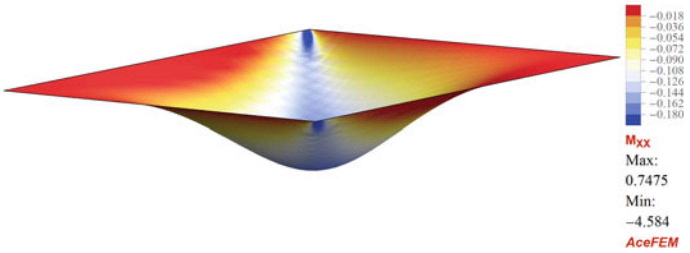


Fig. 13.13 Moment M_{XX} using VE-2 with 32 × 32 elements, St-1b

are automatically refined, as depicted in Fig. 13.11c. The energy of the previous refinement step is used in this convergence study as an indicator for the refinement in the next step.

An analytical solution for the rhombic plate can be found in Morley (1962) for different plate angles, however it is only provided up to three digits. Based on adaptive finite elements solutions convergence results for the rhombic plate were reported in Wang et al. (1984), Babuska and Scapolla (1989) with 3 digit accuracy. We note, that the asymptotic convergence rate of 0.2 (related to the singularity) is recovered for all element formulations with a uniformly refined mesh which is clearly depicted in Fig. 13.12.

Figure 13.13 shows the distribution of the bending moment M_{XX} for the rhombic plate using a uniform mesh with 32 × 32 VE-2 quadrilateral virtual elements which demonstrates the singularity of the bending moment at the obtuse corner.

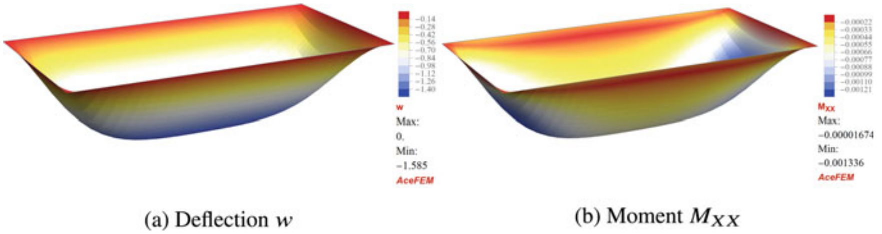
13.4.4 Rectangular Orthotropic Plate

The simplest version of anisotropy in a plate is provided by orthotropic behaviour. It is possible to obtain an analytical solution for such material as given in Timoshenko and Woinowsky-Krieger (1959) for a rectangular plate. Here the orthotropic directions coincide with the main axis of the plate. This example can be used to investigate the response of the virtual plate elements for a more complex material.

Using the same notation as in Timoshenko and Woinowsky-Krieger (1959) the problem is characterized by a simplified material stiffness matrix when compared to (13.69)

$$\hat{\mathbb{D}}_G = \begin{bmatrix} D_x & D_1 & 0 \\ D_1 & D_y & 0 \\ 0 & 0 & D_{xy} \end{bmatrix} = \frac{t^3}{12} \begin{bmatrix} E_x & \hat{E} & 0 \\ \hat{E} & E_y & 0 \\ 0 & 0 & G \end{bmatrix} . \tag{13.125}$$

In this relation t is the thickness of the plate with length $a = 2$ mm and width $b = 1$ mm. The plate is simply supported and subjected to a uniform load q. The material data can be found in Table 13.3.



(a) Deflection w

(b) Moment M_{XX}

Fig. 13.14 Simply supported plate, quantities plotted on the deformed configuration

Table 13.3 Material parameters for the orthotropic plate

Description	Symbol	Value	Unit
Young's modulus in fiber direction	E_x	10000	MPa
Transverse modulus	E_y	1000	MPa
Shear modulus	G	500	MPa
Modulus	$\hat{E}$	500	MPa
Thickness of the specimen	t	0.01	mm
Surface load	q	−0.1	MPa

By noting that $H = D_1 + 2D_{xy} = \hat{E} + 2G$, the deflection can be expressed by the double sum

$$w(X,Y) = \frac{16\,q}{\pi^6} \sum_{m=1,3..} \sum_{n=1,3..} \frac{1}{mn(\frac{m^4}{a^4} D_x + 2\frac{m^2 n^2}{a^2 b^2} H + \frac{n^4}{b^4} D_y)} \sin\left[\frac{m\pi}{a} X\right] \sin\left[\frac{n\pi}{b} Y\right]. \tag{13.126}$$

A convergence study of the deflection w at the center $(X, Y) = (\frac{a}{2}, \frac{b}{2})$ of the plate is performed for the orthotropic plate using element VE-2: st-2b for regular (Fig. 13.11a) and Voronoi (Fig. 13.7b) meshes.

The reference value w_{ref} is computed for the given data with *Mathematica* by using $m = n = 1, 3, \ldots, 31$ terms of the sum (13.126) which yields the converged result $w_{ref} = -1.5835858431216134$ mm. Deflection and distribution of the moment M_{XX} are depicted in Fig. 13.14.

Figure 13.15 depicts the error of deflection w in the middle of the plate. We observe a quadratic convergence rate for element 2 for Q1 and Voronoi meshes VO-R and VO-U. As in the previous example, Q1 meshes yield the best result. This examples underlines that the Kirchhoff-Love virtual elements converge for these more intricate materials with the same order as for isotropic materials. This was also shown for plates with anisotropic materials in Wriggers et al. (2021).

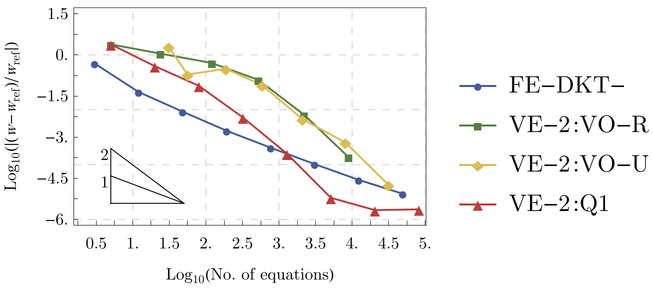


Fig. 13.15 Convergence study of the deflection w for the orthotropic plate

13.5 Stabilization Free C^1-Continuous Virtual Plate Elements for FEM Codes

Virtual plate elements can, of course, assume the simple form of a triangular or quadrilateral element. Such C^1-continuous plate element can be based on both virtual elements, VE-1 or VE-2, see Fig. 13.16. The VE-1:T1 to VE-2:Q1 elements, defined in Fig. 13.16, can be implemented in conventional finite element codes since their number of nodes and degrees of freedom coincide with the node numbers and degrees of freedom of classical finite plate elements. The so constructed C^1-continuous virtual plate elements do not need a high ansatz order within the element, like in the TUBA family of triangular elements in Argyris et al. (1968) or—as quadrilaterals—using a composition of four triangular elements, see De Veubeke (1968). The developed virtual element formulation provides the necessary C^1-continuity for Kirchhoff plates for elements depicted in Fig. 13.16. These have as VE-1:T1 and VE-1:Q1 three unknowns per node and thus in total 9 and 12 unknowns, respectively. As VE-2:T1 and VE-2:Q1 the elements have three unknowns at the vertices and one unknown at the midnodes along the edges summing up to 12 and 16 total unknowns per element, respectively.[10]

Stabilization free formulation. As discussed in Sect. 13.3.6 it is not possible to avoid stabilization for the standard construction of the virtual plate element. However when looking at the formula (13.115) we observe that a proper rank of the plate element can be obtained by raising the ansatz order for the projection w_π. Table 13.4 shows the necessary ansatz order for the projection w_π which avoids rank deficiency.

Thus, for element VE-1:T1 the cubic polynomial (13.98) can be used instead of the quadratic ansatz (13.83) to compute the projection using (13.113) with the ansatz w_h from (13.78) and θ_{ht} for element 1 in $(13.82)_1$. In the same way the projection can be changed for VE-2:T1 by using $w_\pi = \mathbf{H}_w^{(2,4)}\,\mathbf{a}$ instead of (13.98) where now $\mathbf{H}_w^{(2,4)}$ is a fourth order polynomial and $\mathbf{a}$ has 15 independent constants. The resulting projection is on one side less efficient since the equation system to be solved changes

[10] It is even possible to construct virtual plate elements with a higher continuity, see e.g. Brezzi and Marini (2013, 2021), by just introducing higher order ansatz functions at the edges.

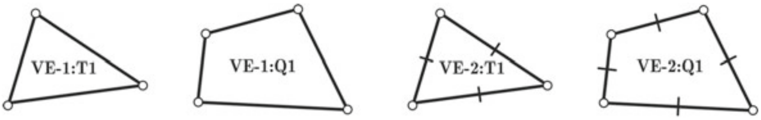


Fig. 13.16 Three and four noded virtual elements VE-1 and VE-2

Table 13.4 Ansatz order w_π of for correct rank of triangular plate

Element	VE-1:T1	VE-2:T1
Ansatz order w_π	3	4

in this case from (10×10) to (15×15) but on the other side the stabilization has not to be computed and the solution is no longer dependent on a stabilization parameter. When testing the rank of the stabilization free elements VE-1:T1 and VE-2:T1 it turns out that the eigenvalues are basically the same as with the stabilization and include the correct number of zero eigenvalues for the rigid body modes.

The VE-2:Q1 element would need a fifth order polynomial for proper rank, see Table 13.4. This leads to a local solution of a (21×21) matrix for the determination of the parameters $\mathbf{a}$. Hence it is more efficient to construct this element from two VE-2:T1 elements by local assembly and elimination of the unknown θ_t related to edge that defines the interface between the two triangles. The same argument hold for VE-1:Q1 which can be easily constructed from two VE-1:T1 elements by local assembly.

Since the virtual elements VE-1:T1 to VE-2:Q1 have C^1-continuity with a low order ansatz function it is of interest to investigate how these virtual plate elements perform on classical finite meshes which are used in engineering software. It is also of interest how the stabilization free triangular virtual elements perform in comparison to the stabilized ones. All test results will be compared with finite elements that approximate the curvature in the same way as the derived virtual elements VE-1:T1 to VE-2:Q1. For the virtual element VE-1 an equivalent Kirchhoff plate element with constant curvature is the nonconforming Morley element, see Morley (1968). For element VE-2 non-conforming triangular plate element, developed by Specht (1988), and the DKT element, see Batoz et al. (1980) can be selected which both rely on a linear curvature approximation.

We start the comparison with two benchmark examples of a clamped plate under uniform and point load, see e.g. Oñate (2013), and illustrate the performance of the virtual plate elements VE-1:T1 to VE-2:Q1 depicted in Fig. 13.16 for stabilizations (13.121), (13.122) and the stabilization free formulation.

13.5.1 Clamped Plate Under Uniform Load

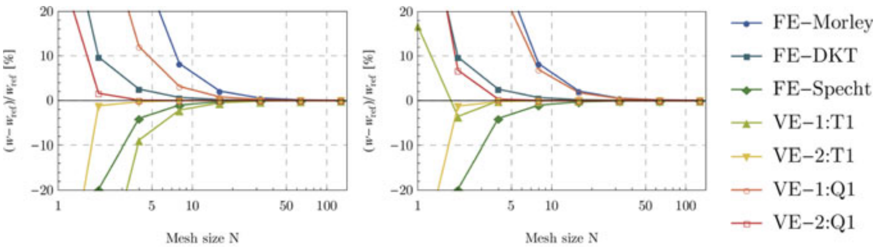
The clamped plate under uniform load $\bar{q}$ was already discussed in Sect. 13.4.2. Geometry and material data are the same as in Sect. 13.4.2 and the analytical solutions are given in Table 13.2. Results related to the convergence behaviour of a number of conforming and non-conforming plate elements are provided in Oñate (2013) for the case of a uniform load.

We compare the results of solutions obtained with finite elements from Morley, Specht and the DKT formulation with the virtual plate elements that have the same nodes as standard triangles and quadrilaterals, see Fig. 13.17. For the pointwise stabilization “st-1”, see (13.121), Fig. 13.17a demonstrates that VE-1 has a better coarse mesh accuracy than the Morley element despite the fact that both elements have a constant approximation of the curvature. The solution of the stabilization-free VE-1:T1 element is exactly the same as the one of element VE-1:T1 with stabilization “st-1”, and thus, it is not extra marked in Fig. 13.17a.

For the pointwise stabilization VE-2 outperforms the DKT and Specht elements, although all elements approximate the curvature with a linear polynomial. It is interesting that the triangular element VE-1 has an extremely good coarse mesh accuracy when using the continuous stabilization “st-2” in (13.122) as shown in Fig. 13.17b. The triangular VE-1 element with stabilization “st-2b” is simple, efficient, depicts a superior coarse grid accuracy and has only three nodes. Thus it qualifies as a C^1-continuous Kirchhoff plate element for legacy finite element codes.

Interestingly, the behaviour of VE-2:T1 is not influenced by the type of stabilization, also the stabilization-free version yields the same results as VE-2:T1 with stabilizations “st-1b” and ”st-2b”.

The rate of convergence is shown in Fig. 13.18 for the finite elements and the virtual element VE-2 with linear curvature. It can be clearly seen that VE-2 depicts a higher convergence order when compared to the DKT and Specht elements as demonstrated. Thus the conforming virtual element 2 as a triangular well as as a quadrilateral element is performing extremely well and thus a good candidate for



(a) Pointwise stabilisation (st-1b)

(b) Continuous stabilisation (st-2b)

Fig. 13.17 Convergence study of the deflection w under uniform load for VE-1 and VE-2, compared with finite elements

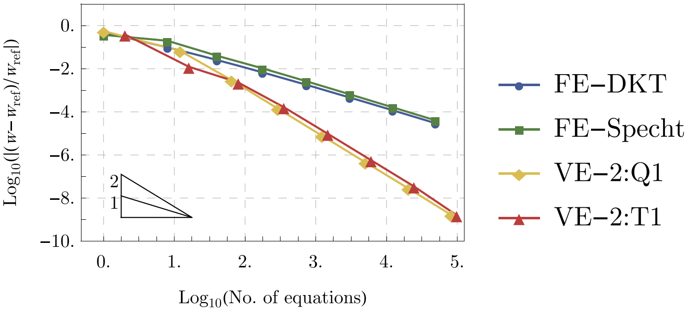


Fig. 13.18 Convergence of the deflection w for element 2, compared to Specht and DKT finite elements for a plate under uniform load

finite element software as well. This is especially true for the stabilization-free version since it does not rely on additional parameters.

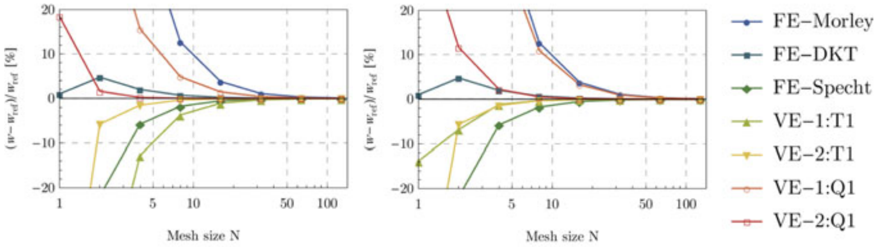
13.5.2 Clamped Plate Under Point Load

In a second example we apply the triangular and quadrilateral virtual element to the clamped plate under point load $F = 64$ kN. The geometrical and the material data are described in Sect. 13.4.2. The analytical solution is reported in Timoshenko and Woinowsky-Krieger (1959) as $w = -0.0896\frac{F}{D}$. The plot in Fig. 13.19 depicts the convergence behaviour of the different element formulations.

As shown in Fig. 13.19a the pointwise stabilization "st-1b" yields the best results for VE-2:Q1 while Fig. 13.19b depicts the superior coarse mesh accuracy of element VE-1:T1 for the continuous stabilization "st-2b", as in the previous example. For the triangular element VE-2:T1 there is no difference in the results for stabilization "st-1b", "st-2b" and the stabilization free version.

The asymptotic convergence behaviour can be observed in Fig. 13.20. Here again the coarse mesh accuracy of the VE-1:T1 element is demonstrated in Fig. 13.20a for the continuous stabilization "st-2b". Expectantly the same asymptotic convergence rate is achieved for VE-1: T1 and VE-1:Q1. Due to the point load the rate of convergence is lower for the higher order ansatz which can be seen in Fig. 13.20b. Here VE-2: Q1 depicts the best performance when using the pointwise stabilization.

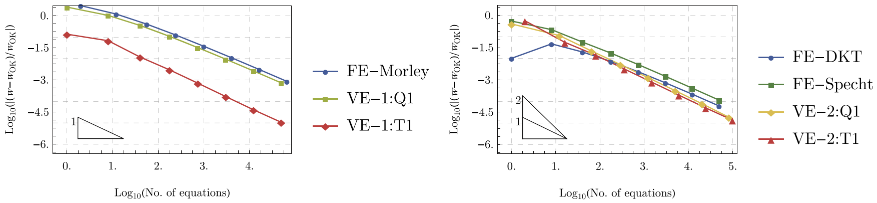
It is impressive to see in both examples, that the proposed virtual elements (VE-1:T1 st-2b and VE-2:Q2 st-1b) outperform the DKT element which, in the engineering literature, is known to be an excellent element. This qualifies both virtual plate elements as candidates for engineering software related to the analysis of thin plates. Especially the very simple triangular element VE-1:T1 "st-2b" is a good candidate since it fits easily in existing software packages, having only three nodes and the same number of unknowns at each node which is actually equivalent to using the



(a) Pointwise stabilisation (st-1b)

(b) Continuous stabilisation (st-2b)

Fig. 13.19 Convergence study of the deflection *w* under point load for VE-1 and -2, compared with finite elements



(a) VE-1:st-2b versus Morley

(b) VE-2:st-2b versus DKT and Specht

Fig. 13.20 Convergence of the deflection *w* for Elements 1 and 2, compared to Morley and DKT finite elements for plate under point load

Specht or DKT element. On the other side, the stabilization free form of the VE-2:T1 element is very accurate and has a superior convergence rate.

13.5.3 L-Shaped Plate

An example that is more demanding is the "L-shaped" plate, see Fig. 13.21. The plate is loaded by a uniform load $q = -1$ and simply supported at all sides ($w = 0$).

The geometrical data are provided in Fig. 13.21 with $H_1 = H_2 = 4$ and $W_1 = W_2 = 4$. Young's modulus is given by $E = 10^8$, the Poisson ratio by $\nu = 0.3$ and the plate thickness is $t = 0.01$. Here a dimensionless representation is provided, assuming that all geometric and material data match.

The solution of the plate is governed by a singularity at the re-entrant corner. For the case of all sides being simply supported the regularity in the critical corner is such that $W \in H^{7/3}$, see Beirão da Veiga et al. (2008), which results in a convergence rate of $O(h^\beta)$ with $\beta = \min(\frac{1}{3}, k) = \frac{1}{3}$ where k is the ansatz order of the element.

For the numerical simulations the virtual elements VE-1 as well as VE-2 are selected and compared with results obtained by the DKT element. The plate is discretized using uniform meshes based on triangular and quadrilateral elements.

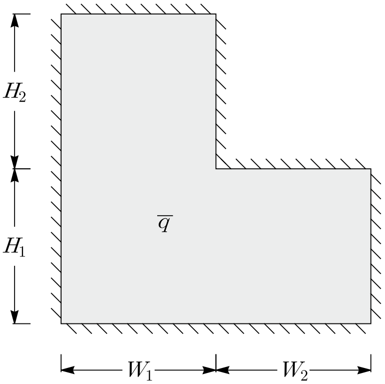
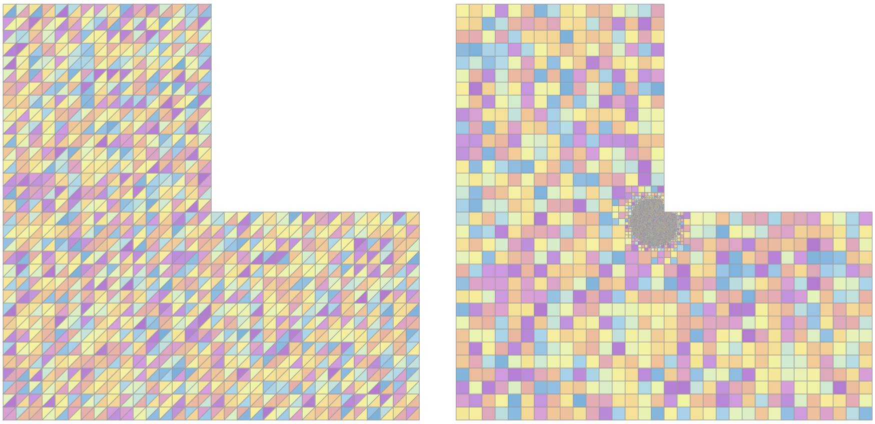


Fig. 13.21 Simply supported L-shaped plate



(a) uniform triangular mesh

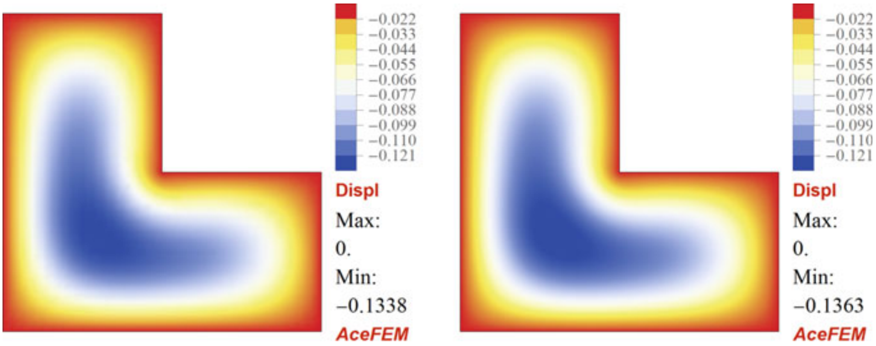
(b) locally refined Q1-mesh

Fig. 13.22 Different meshes for the L-shaped plate

Furthermore, a mesh is employed that is refined at the re-entrant corner, see Fig. 13.22b. The mesh in Fig. 13.22a consists of 3 areas that are discretized by 2 16 × 16 elements while the mesh in Fig. 13.22b is constructed using 3 areas of 16 × 16 quadrilaterals and a refinement based on a bi-section approach at the re-entrant corner which halves the element size 5 times in the refined zone.

Figure 13.23a shows the deflection obtained with the locally refined mesh of Fig. 13.22b while Fig. 13.23b depicts the result obtained with a 64 × 64 mesh per area. Here both discretizations yield basically the same results.

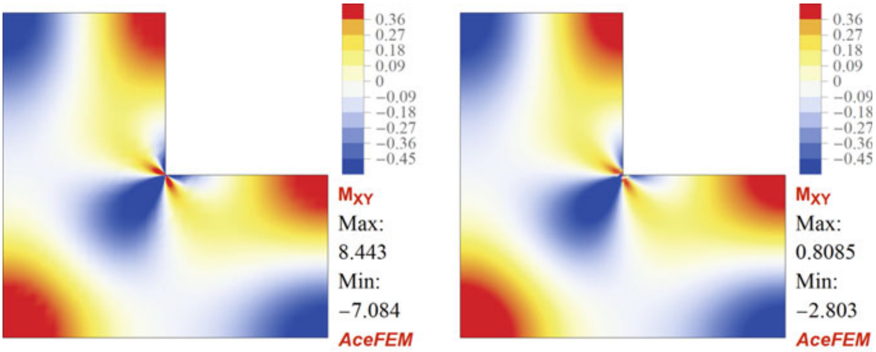
The difference of the accuracy in predicting the bending moments is demonstrated for M_{XY} in Fig. 13.24. It is easily observed, by looking at the minimum and maximum values for the bending moment M_{XY}, that the uniform mesh using the DKT



(a) locally refined mesh

(b) uniform DKT mesh

Fig. 13.23 Deflection of the plate



(a) locally refined mesh

(b) uniform DKT mesh

Fig. 13.24 Distribution of bending moments M_{XY}

element, see Fig. 13.24b, does not produce the same accurate results than virtual element solution based in the locally refined mesh, see 13.24a. The latter picks up the singularity in a much clearer way.

It is obvious that this type of problem requires a mesh refinement at the areas where singularities occur. Ideally the refinement has to reflect the order of the singularity which then would lead to optimal convergence rates, for details see e.g. Wang et al. (1984), Babuska and Scapolla (1989).

References

Antonietti, P.F., G. Manzini, and M. Verani. 2018. The fully nonconforming virtual element method for biharmonic problems. *Mathematical Models and Methods in Applied Sciences* 28 (02): 387–407.

Argyris, J.H., I. Fried, and D.W. Scharpf. 1968. The TUBA family of plate elements for the matrix displacement method. *The Aeronautical Journal* 72 (692): 701–709.

Babuska, I., and T. Scapolla. 1989. Benchmark computation and performance evaluation for a rhombic plate bending problem. *International Journal for Numerical Methods in Engineering* 28: 155–179.

Batoz, J.L., and M.B. Tahar. 1982. Evaluation of a new quadrilateral thin plate bending element. *International Journal for Numerical Methods in Engineering* 18 (11): 1655–1677.

Batoz, J.L., K.J. Bathe, and L.W. Ho. 1980. A study of three-node triangular plate bending elements. *International Journal for Numerical Methods in Engineering* 15 (12): 1771–1812.

Bazeley G.P., Y.K. Cheung, B.M. Irons, and O. Zienkiewicz. 1965. Triangular elements in plate bending-conforming and non-conforming solutions. In *Conference on matrix methods in structural mechanics*. Ohio: Air Force Institute of Technology, Wright–Patterson.

Beirão da Veiga L., D. Mora, and G. Rivera. 2019c. Virtual elements for a shear-deflection formulation of Reissner–Mindlin plates. *Mathematics of Computation* 88(315): 149–178 .

Beirão da Veiga L., J. Niiranen, and R. Stenberg. 2008. A family of C^0 finite elements for Kirchhoff plates II: Numerical results. *Computer Methods in Applied Mechanics and Engineering* 197: 1850–1864.

Bell, K. 1969. A refined triangular plate bending finite element. *International Journal for Numerical Methods in Engineering* 1 (1): 101–122.

Brezzi, F., and L.D. Marini. 2013. Virtual element methods for plate bending problems. *Computer Methods in Applied Mechanics and Engineering* 253: 455–462.

Brezzi, F., and L.D. Marini. 2021. Finite elements and virtual elements on classical meshes. *Vietnam Journal of Mathematics* 49: 871–899.

Bufler, H., and E. Stein. 1970. Zur Plattenberechnung mittels finiter Elemente. *Ingenieurarchiv* 39: 248–260.

Chinosi, C., and L.D. Marini. 2016. Virtual element method for fourth order problems: L2-estimates. *Computers & Mathematics with Applications* 72 (8): 1959–1967.

Clough, R.W., J.L. Tocher. 1965. Finite element stiffness matrices for analysis of plate bending. In *Proceedings of the first conference on matrix methods in structural mechanics*, 515–546.

De Veubeke, B.F. 1968. A conforming finite element for plate bending. *International Journal of Solids and Structures* 4 (1): 95–108.

Hughes, T.J.R. 1987. *The finite element method*. Englewood Cliffs, New Jersey: Prentice Hall.

Hughes, T.J.R. 2012. *The finite element method: Linear static and dynamic finite element analysis*. Courier Corporation.

Hughes, T.J.R., R.L. Taylor, and W. Kanoknukulchai. 1977. A simple and efficient finite element for plate bending. *International Journal for Numerical Methods in Engineering* 11: 1529–1547.

Korelc, J., P. Wriggers. 2016. *Automation of finite element methods*. Berlin: Springer.

Melosh, R.J. 1961. A stiffness matrix for the analysis of thin plates in bending. *Journal of the Aerospace Sciences* 28 (1): 34–42.

Meng, J., and L. Mei. 2020. A linear virtual element method for the Kirchhoff plate buckling problem. *Applied Mathematical Letters* 103.

Mora, D., and I. Velásquez. 2020. Virtual element for the buckling problem of Kirchhoff-love plates. *Computer Methods in Applied Mechanics and Engineering* 360: 112687.

Mora D., G. Rivera, and I. Velásquez. 2018. A virtual element method for the vibration problem of Kirchhoff plates. *ESAIM: Mathematical Modelling and Numerical Analysis* 52(4): 1437–1456.

Morley, L. 1962. Bending of a simply supported rhombic plate under uniform normal loading. *The Quarterly Journal of Mechanics and Applied Mathematics* 15 (4): 413–426.

Morley, L.S.D. 1968. The triangular equilibrium element in the solution of plate bending problems. *Aeronautical Quarterly* 19 (2): 149–169.

Oñate, E. 2013. *Structural analysis with the finite element method, Vol. 2: Beams, plates and shells*. Springer.

Reddy, J.N. 1999. *Theory and analysis of elastic plates and shells*. CRC Press.

Reddy, J.N. 2004. *Mechanics of laminated composite plates and shells: Theory and analysis*. CRC Press.

Specht, B. 1988. Modified shape functions for the three-node plate bending element passing the patch test. *International Journal for Numerical Methods in Engineering* 26: 705–715.

Taylor, R.L., and S. Govindjee. 2004. Solution of clamped rectangular plate problems. *Communications in Numerical Methods in Engineering* 20 (10): 757–765.

Timoshenko, S., and S. Woinowsky-Krieger. 1959. *Theory of plates and shells*, vol. 2. New York: McGraw-hill.

Tong, P. 1969. Exact solution of certain problems by the finite element method. *AIAA Journal* 7 (1): 179–180.

Wang, D., I. Katz, and B. Szabo. 1984. h-and p-version finite element analyses of a rhombic plate. *International Journal for Numerical Methods in Engineering* 20 (8): 1399–1405.

Withum D. Berechnung von Platten nach dem Ritz'schen Verfahren mit Hilfe dreieckförmiger Maschennetze. *Technical Report 9*, Mitteilungen des Instiituts für Statik der Technischen Hochschule Hannover (1966).

Wriggers, P., B. Hudobivnik, and F. Aldakheel. 2021. NURBS-based geometries: A mapping approach for virtual serendipity elements. *Computer Methods in Applied Mechanics and Engineering* 378: 113732.

Wriggers, P. 2022. On a virtual element formulation for trusses and beams. *Archive of Applied Mechanics* 92 (6): 1655–1678.

Wriggers, P., B. Hudobivnik, and O. Allix. 2022. On two simple virtual Kirchhoff-Love plate elements for isotropic and anisotropic materials. *Computational Mechanics* 69: 615–637.

Yu, Y. 2022. mVEM: A MATLAB software package for the virtual element methods. arXiv:2204.01339.

Zienkiewicz, O.C., R.L. Taylor, and J.M. Too. 1971. Reduced integration technique in general analysis of plates and shells. *International Journal for Numerical Methods in Engineering* 3: 275–290.

Appendix A
Formulae in Virtual Element Formulations

This appendix provides some useful formulae that help to derive efficient integration schemes that can be applied within the discretization process of virtual elements.

A.1 Integration Over Polygons

A general polygon occurs often in two-dimensions often as a virtual element, see Fig. A.1. The following formulae can be used to compute certain integrals that appear during the derivation of virtual elements. Generally they follow from Green's theorem

$$\int\limits_{\Omega} \left[\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right] \mathrm{d}x \, \mathrm{d}y = \int\limits_{\Gamma} P(x, y) \, \mathrm{d}x + \int\limits_{\Gamma} Q(x, y) \, \mathrm{d}y \tag{A.1}$$

where Γ is the boundary of Ω. Based on this theorem the following relations can be derived which allow to compute the integrals by just using the vertices of the polygon.

The area of the polygon is given by

$$A = \int\limits_{\Omega} d\Omega = \frac{1}{2} \sum_{i=1}^{n} (x_i \, y_{i+1} - x_{i+1} \, y_i) \tag{A.2}$$

Higher order integrals, up to polynomial order $n = 2$:

$$\int\limits_{\Omega} x \, d\Omega = \frac{1}{6} \sum_{i=1}^{n} (x_i \, y_{i+1} - x_{i+1} \, y_i)(x_i + x_{i+1}) \tag{A.3}$$

P. Wriggers et al., *Virtual Element Methods in Engineering Sciences*,
https://doi.org/10.1007/978-3-031-39255-9

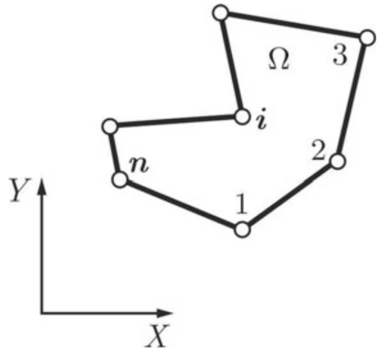


Fig. A.1 General polygon that describes a virtual element

$$\int_\Omega y \, d\Omega = \frac{1}{6} \sum_{i=1}^{n} (x_i \, y_{i+1} - x_{i+1} \, y_i)(y_i + y_{i+1}) \tag{A.4}$$

$$\int_\Omega x^2 \, d\Omega = \frac{1}{12} \sum_{i=1}^{n} (x_i \, y_{i+1} - x_{i+1} \, y_i)(x_i^2 + x_i \, x_{i+1} + x_{i+1}^2) \tag{A.5}$$

$$\int_\Omega y^2 \, d\Omega = \frac{1}{12} \sum_{i=1}^{n} (x_i \, y_{i+1} - x_{i+1} \, y_i)(y_i^2 + y_i \, y_{i+1} + y_{i+1}^2) \tag{A.6}$$

$$\int_\Omega x \, y \, d\Omega = \frac{1}{24} \sum_{i=1}^{n} (x_i \, y_{i+1} - x_{i+1} \, y_i)(x_i \, y_{i+1} + 2x_i \, y_i + 2 \, x_{i+1} \, y_{i+1} + x_{i+1} \, y_i) \tag{A.7}$$

In these relations, x_i, y_i are the coordinates of the ith polygon vertex, for $1 \leq i \leq n$. Also, x_{n+1}, y_{n+1} are assumed to be equal to the coordinates of the first vertex, i.e., $x_{n+1} = x_1$ and $y_{n+1} = y_1$.

For higher order polynomials the following relation holds

$$\int_\Omega x^p \, y^q \, \mathrm{d}x \, \mathrm{d}y = \frac{1}{(p+q+2)(p+q+1) \begin{pmatrix} p+q \\ p \end{pmatrix}} \times \tag{A.8}$$

$$\sum_{i=1}^{n} (x_i \, y_{i+1} - x_{i+1} \, y_i) \sum_{k=0}^{p} \sum_{l=0}^{q} \begin{pmatrix} k+l \\ l \end{pmatrix} \begin{pmatrix} p+q-k-l \\ q-l \end{pmatrix} x_{i+1}^k \, x_i^{p-k} \, y_{i+1}^l \, y_i^{q-l}$$

see Steger (1996). The expression $\begin{pmatrix} n \\ k \end{pmatrix}$ is the binomial coefficient which follows from the product

$$\binom{n}{k} = \prod_{i=1}^{k} \frac{n+1+i}{i}$$

for integer values of n and $k > 0$. For $k = 0$ the value of $\binom{n}{0}$ is always 1.

Special form for linear functions. The barycenter $\boldsymbol{x}_b = (x_b, y_b)$ of a two-dimensional domain Ω is given by

$$x_b = \frac{1}{\Omega} \int\limits_{\Omega} x \, \mathrm{d}\Omega \qquad \text{and } y_b = \frac{1}{\Omega} \int\limits_{\Omega} y \, \mathrm{d}\Omega \,. \tag{A.9}$$

Thus the integration of a linear function $f(x, y) = a + bx + cy$ over the domain can be expressed by

$$\int\limits_{\Omega} f(x, y) \, \mathrm{d}\Omega = f(x_b, y_b) \, \Omega \,. \tag{A.10}$$

This results is also valid in the three-dimensional case

$$\int\limits_{\Omega} f(\boldsymbol{x}) \, \mathrm{d}\Omega = f(\boldsymbol{x}_b) \, \Omega \tag{A.11}$$

where $\boldsymbol{x}_b$ is the barycenter, Ω the volume and $f(\boldsymbol{x}) = a + bx + cy + dz$.

A.2 Computation of Volume by Surface Integrals

Classically the divergence or Gauss theorem allows the evaluation of a volume integral (Ω) over a closed surface Γ with outward normal $\boldsymbol{n}$. Here different variants can be listed depending on the type of integrands. These are summarized in the list below for a vector $\boldsymbol{u}$ and a tensor $\boldsymbol{T}$

$$\int\limits_{\Omega} \mathrm{Div}\, \boldsymbol{u} \, \mathrm{d}\Omega = \int\limits_{\Gamma} \boldsymbol{u} \cdot \boldsymbol{n} \, \mathrm{d}\Gamma \,, \tag{A.12}$$

$$\int\limits_{\Omega} \mathrm{Div}\, \boldsymbol{T} \, \mathrm{d}\Omega = \int\limits_{\Gamma} \boldsymbol{T} \, \boldsymbol{n} \, \mathrm{d}\Gamma \,, \tag{A.13}$$

$$\int\limits_{\Omega} \mathrm{Rot}\, \boldsymbol{u} \, \mathrm{d}\Omega = \int\limits_{\Gamma} \boldsymbol{u} \times \boldsymbol{n} \, \mathrm{d}\Gamma \,. \tag{A.14}$$

For the gradient of a scalar function $\alpha(\boldsymbol{x})$ one can derive from the Gauss theorem

$$\int_\Omega \nabla\alpha(\boldsymbol{x})\, \mathrm{d}\Omega = \int_\Omega \mathrm{Grad}\,\alpha(\boldsymbol{x})\, \mathrm{d}\Omega = \int_\Gamma \alpha(\boldsymbol{x})\,\boldsymbol{n}\, \mathrm{d}\Gamma\,. \tag{A.15}$$

Using these results, products of vectors and tensors can be transformed partly to surface integrals. Examples are the rules below

$$\mathrm{Div}\,(\alpha\,\boldsymbol{u}) = \boldsymbol{u}\cdot\mathrm{Grad}\,\alpha + \alpha\mathrm{Div}\,\boldsymbol{u}\,, \tag{A.16}$$

$$\mathrm{Div}\,(\alpha\,\boldsymbol{T}) = \boldsymbol{T}\,\mathrm{Grad}\,\alpha + \alpha\mathrm{Div}\,\boldsymbol{T}\,, \tag{A.17}$$

$$\mathrm{Div}\,(\boldsymbol{T}\,\boldsymbol{u}) = \boldsymbol{T}^T\cdot\mathrm{Grad}\,\boldsymbol{u} + \mathrm{Div}\,\boldsymbol{T}^T\cdot\boldsymbol{u}\,, \tag{A.18}$$

$$\mathrm{Div}\,(\boldsymbol{u}\otimes\boldsymbol{w}) = (\mathrm{Grad}\,\boldsymbol{u})\,\boldsymbol{w} + (\mathrm{Div}\,\boldsymbol{w})\,\boldsymbol{u} \tag{A.19}$$

which then yields e.g. for (A.18) together with (A.12)

$$\int\limits_\Omega \boldsymbol{T}^T\cdot\mathrm{Grad}\,\boldsymbol{u}\;\mathrm{d}\Omega + \int\limits_\Omega \mathrm{Div}\,\boldsymbol{T}^T\cdot\boldsymbol{u}\;\mathrm{d}\Omega = \int\limits_\Gamma (\boldsymbol{T}\,\boldsymbol{u})\cdot\boldsymbol{n}\;\mathrm{d}\Gamma\,. \tag{A.20}$$

Application of the divergence theorem. Another way to compute an area or volume integral can be derived by using the divergence theorem which reduces the integral by one-dimension By noting that a function can be written in the two-dimensional case as

$$f(x,y) = \frac{1}{2}\,\mathrm{div}\left\{\int f(x,y)\mathrm{d}x\,,\int f(x,y)\mathrm{d}y\right\} \tag{A.21}$$

an area integral can be transformed to a line integral

$$\int\limits_\Omega f(x,y)\,\mathrm{d}\Omega = \frac{1}{2}\int\limits_\Gamma \left\{\int f(x,y)\mathrm{d}x\,,\int f(x,y)\mathrm{d}y\right\}\cdot\boldsymbol{n}\,\mathrm{d}\Gamma \tag{A.22}$$

where $\boldsymbol{n}$ is the outward normal vector of the area Ω. In the same way we obtain for three dimensions

$$\int\limits_\Omega f(x,y,z)\,\mathrm{d}\Omega = \frac{1}{3}\int\limits_{\Gamma_f} \left\{\int f(x,y,z)\mathrm{d}x\,,\int f(x,y,z)\mathrm{d}y\,,\int f(x,y,z)\mathrm{d}z\right\}\cdot\boldsymbol{n}\,\mathrm{d}\Gamma \tag{A.23}$$

where now $\boldsymbol{n}$ is the outward normal to the surface Γ_f of the volume Ω.

This formulation is especially useful for the integration of polynomials over polygons or polyhedra. In the two-dimensional case it follows

$$\int\limits_\Omega x^p y^q \mathrm{d}\Omega = \frac{1}{2}\int\limits_\Gamma \left[\frac{x^{p+1}y^q}{p+1}n_x + \frac{x^p y^{q+1}}{q+1}n_y\right]\mathrm{d}\Gamma \tag{A.24}$$

and in the three-dimensional case one can write

$$\int\limits_{\Omega} x^p y^q z^r \mathrm{d}\Omega = \frac{1}{3} \int\limits_{\Gamma_f} \left[\frac{x^{p+1} y^q z^r}{p+1} n_x + \frac{x^p y^{q+1} z^r}{q+1} n_y + \frac{x^p y^q z^{r+1}}{r+1} n_z \right] \mathrm{d}\Gamma \quad \text{(A.25)}$$

where p, q and r is the polynomial degree.[1]

Using the barycenter. When computing the volume integral over a scalar function p, which gradient is constant, Grad $p = const.$, one can shift the integration to the boundary using the relation (A.16) and (A.12). First we reformulate

$$p\,\mathrm{Div}\boldsymbol{x} = \mathrm{Div}(p\,\boldsymbol{x}) - \mathrm{Grad}\,p \cdot \boldsymbol{x}\,. \quad \text{(A.26)}$$

Then with a change of coordinates $\boldsymbol{x}_b = \boldsymbol{x} - \boldsymbol{b}$, where $\boldsymbol{b}$ denotes the coordinates of the barycenter of the virtual element, we can write with $\mathrm{Div}(\boldsymbol{x}_b) = \mathrm{Div}(\boldsymbol{x} - \boldsymbol{b}) = \mathrm{Div}(\boldsymbol{x})$

$$\int_{\Omega} p\,\mathrm{Div}\boldsymbol{x}\; d\Omega = \int_{\Omega} \mathrm{Div}(p\,\boldsymbol{x}_b)\; d\Omega - \int_{\Omega} \mathrm{Grad}\,p \cdot \boldsymbol{x}_b\; d\Omega\,. \quad \text{(A.27)}$$

Now with $\mathrm{Div}\boldsymbol{x} = 2$ in 2d and $\mathrm{Div}\boldsymbol{x} = 3$ in 3d, and taking into account Grad $p = const.$ we can write for the two-dimensional case

$$\int_{\Omega} p\; d\Omega = \frac{1}{2} \left[\int_{\Omega} \mathrm{Div}(p\,\boldsymbol{x}_b)\; d\Omega - \mathrm{Grad}\,p \cdot \int_{\Omega} \boldsymbol{x}_b\; d\Omega \right]. \quad \text{(A.28)}$$

Since $\int_{\Omega} \boldsymbol{x}_b\; d\Omega = 0$ we arrive finally at

$$\int_{\Omega} p\; \mathrm{d}\Omega = \frac{1}{2} \int_{\Omega} \mathrm{Div}(p\,\boldsymbol{x}_b)\; \mathrm{d}\Omega \quad \text{(A.29)}$$

where the latter surface integral can be transformed to a contour integral

$$\int_{\Omega} p\; \mathrm{d}\Omega = \frac{1}{2} \int_{\Gamma} p\,\boldsymbol{x}_b \cdot \boldsymbol{n}\; \mathrm{d}\Gamma \quad \text{(A.30)}$$

with $\boldsymbol{n}$ being the outward normal vector related to the contour.

In the three dimensional case the same result holds for transformation of a volume to a surface integral, only the factor changes

$$\int_{\Omega} p\; \mathrm{d}\Omega = \frac{1}{3} \int_{\Gamma_f} p\,\boldsymbol{x}_b \cdot \boldsymbol{n}\; \mathrm{d}\Gamma \quad \text{(A.31)}$$

[1] The right hand side of Eq. (A.25) can be transformed to a line integral over the edges of the polygon using (A.24). However this yields a quite complex formulation that does not lead to an efficient evaluation of the resulting line integral. The complexity follows from the fact that z has to be expressed as a function of x and y using the normal form of face Γ_f.

and the integral on the right hand side has to be computed over the surface Γ_f.

This result can be extended to a vector valued function $\boldsymbol{w}$ with $\text{Grad}\,\boldsymbol{w} = const.$ by using the identity, see (A.19),

$$\text{Div}(\boldsymbol{w} \otimes \boldsymbol{x}_b) = (\text{Grad}\,\boldsymbol{w})\,\boldsymbol{x}_b + (\text{Div}\boldsymbol{x}_b)\boldsymbol{w}\,. \tag{A.32}$$

This leads for the two-dimensional case to

$$\int_\Omega \boldsymbol{w}\;\mathrm{d}\Omega = \frac{1}{2}\int_\Omega \text{Div}(\boldsymbol{w} \otimes \boldsymbol{x}_b)\;\mathrm{d}\Omega = \frac{1}{2}\int_\Gamma (\boldsymbol{w} \otimes \boldsymbol{x}_b)\mathbf{n}\;\mathrm{d}\Gamma \tag{A.33}$$

and hence the integral over the surface Ω can be transformed in a line integral over the contour Γ

$$\int_\Omega \boldsymbol{w}\;\mathrm{d}\Omega = \frac{1}{2}\int_\Gamma \boldsymbol{w}\,(\boldsymbol{x}_b \cdot \boldsymbol{n})\;\mathrm{d}\Gamma \tag{A.34}$$

where $\boldsymbol{n}$ is the outward normal to the contour Γ.

The same is true in three-dimensional applications where now a volume integral is transformed to a surface integral

$$\int_\Omega \boldsymbol{w}\;\mathrm{d}\Omega = \frac{1}{3}\int_{\Gamma_f} \boldsymbol{w}\,(\boldsymbol{x}_b \cdot \boldsymbol{n})\;\mathrm{d}\Gamma \tag{A.35}$$

where $\boldsymbol{n}$ is the outward normal to the surface Γ_f.

Appendix B
Definition and Labeling of Different Mesh Types

Meshes of complex geometries can be employed within the virtual element method. These meshes include convex and non-convex shaped elements as well as Voronoi cells. A precise naming of the different mesh types, here abbreviated by ⊞, is introduced such that numerical results in the different applications of VEM are clearly distinguishable:

- Virtual element: **VE**—⊞,
- Finite element: **FE**—⊞.

Virtual elements VE—⊞ are employed to numerically solve the examples of each chapter. The abbreviation ⊞ stands for different types of meshes used in the analysis. In the two-and three-dimensional case we set

- **Q1** for a mesh consisting of quadrilateral elements with 4 nodes, see left part of Fig. B.1,
- **Q2S** for a mesh consisting of quadrilateral elements with 8 nodes and straight edges, see middle and right part of Fig. B.1, which can have arbitrary shape,
- **T1**: a two-dimensional triangular element with linear shape functions and 3 nodes, see left part of Fig. B.4,
- **T2**: a two-dimensional triangular element with quadratic shape functions and 6 nodes and straight edges, see middle and right part of Fig. B.4, which can have arbitrary shape as long as the edges are straight,
- **H1** for a mesh consisting of hexahedral elements with 8 nodes, see left part of Fig. B.2,
- **H2S** for a mesh consisting of hexahedral elements with 20 nodes, see middle part of Fig. B.2. Note that the edges between nodes of the element on the right side of Fig. B.2 have to be straight,
- **H2** for a mesh consisting of hexahedral elements with 26 nodes which is basically the same as the H2S mesh, but has an additional mid node at each of the 6 surfaces,
- **O1**: a three-dimensional tetrahedral element with linear shape functions and four nodes, see left part of Fig. B.5,

P. Wriggers et al., *Virtual Element Methods in Engineering Sciences*,
https://doi.org/10.1007/978-3-031-39255-9

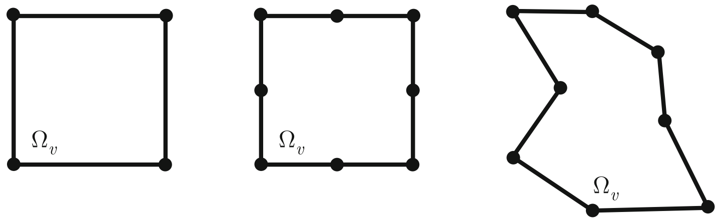


Fig. B.1 Quadrilateral and polygonal elements with 4 (Q1) and 8 (Q2S) nodes

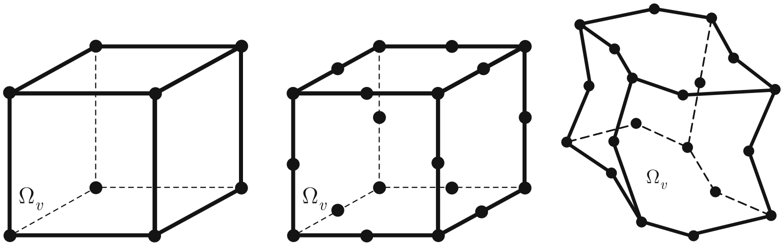


Fig. B.2 Hexahedral elements with 8 (H1) and 20 (H2S) nodes

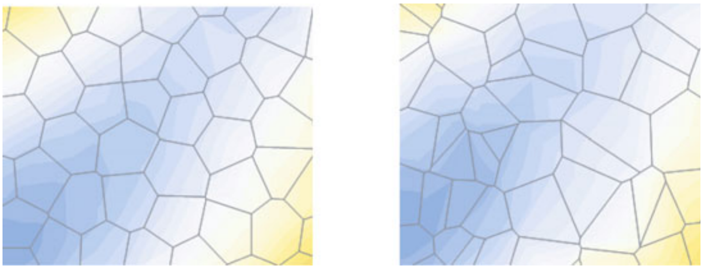


Fig. B.3 Voronoi mesh with uniformly (VOU) and randomly size distribution (VOR)

- **O2**: a three-dimensional tetrahedral element with quadratic shape functions and ten nodes, see right part of Fig. B.5. Note that the edges between nodes of the element on the right side of Fig. B.5 have to be straight,
- **VOU** for a mesh consisting of Voronoi cells that have uniform size, see left part of Fig. B.3, and
- **VOR** for a mesh consisting of Voronoi cells that have randomly distributed size, see right part of Fig. B.3.

Additionally the following finite elements FE-⊞ are used where the abbreviation ⊞ stands for:

- **T1**: a two-dimensional triangular element with linear shape functions and three nodes, see left part of Fig. B.4,

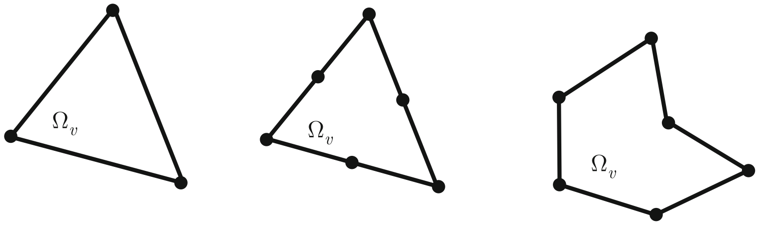


Fig. B.4 Triangular elements with 3 (T1) and 6 (T2) nodes

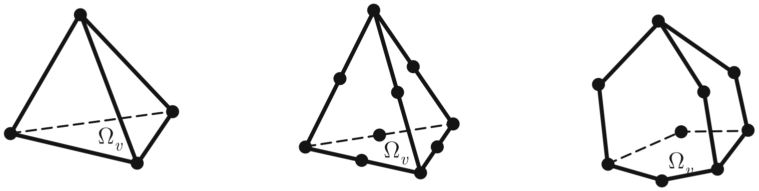


Fig. B.5 Tetrahedral elements with 4 (T1) and 10 (T2) nodes

- **T2**: a two-dimensional triangular element with quadratic shape functions and six nodes, see middle part of Fig. B.4,
- **Q1**: a two-dimensional quadrilateral finite element with linear shape functions and 4 nodes, see left part of Fig. B.1,
- **Q2** : a two-dimensional quadrilateral finite element with quadratic shape functions and nine nodes,
- **Q2S**: a two-dimensional quadrilateral serendipity finite element with quadratic shape functions with eight nodes, see middle part of Fig. B.1,
- **O1**: a three-dimensional tetrahedral element with linear shape functions and four nodes, see left part of Fig. B.5,
- **O2**: a three-dimensional tetrahedral element with quadratic shape functions and ten nodes, see middle part of Fig. B.5,
- **H1**: a three-dimensional hexahedral finite element with linear shape functions and eight nodes, see left part of Fig. B.2,
- **H2**: a three-dimensional hexahedral finite element with quadratic shape functions and 27 nodes,
- **H2S**: a three-dimensional hexahedral serendipity finite element with quadratic shape functions and 20 nodes, see middle part of Fig. B.2.

Generally the element types of the finite element discretization can also be employed in a virtual element discretization as long as they do not have nodes inside the element such as the Q2 and H2 finite elements. However the finite elements cannot assume geometrical shapes, like the elements on the right parts of Figs. B.1, B.4, B.2 and B.5, since the associated Q2, Q2S, T2, H2S, H2 and O2 elements have quadratic shape functions that lead within the isoparametric concept to quadratic surface representations.

Reference

Steger, C. 1996. On the calculation of arbitrary moments of polygons. Technical report, Forschungsgruppe Bildverstehen, Informatik IX, TU München. Technical report FGBV-95-05.