ORIGINAL PAPER

A virtual element formulation for general element shapes

P. Wriggers[1] · B. Hudobivnik[1] · F. Aldakheel[1]

Abstract

The virtual element method is a lively field of research, in which considerable progress has been made during the last decade and applied to many problems in physics and engineering. The method allows ansatz function of arbitrary polynomial degree. However, one of the prerequisite of the formulation is that the element edges have to be *straight*. In the literature there are several new formulations that introduce curved element edges. These virtual elements allow for specific geometrical forms of the course of the curve at the edges. In this contribution a new methodology is proposed that allows to use general mappings for virtual elements which can model arbitrary geometries.

Keywords Virtual element method · Stabilization · Bezier splines · Isoparametric maps

1 Introduction

The many different approaches to the approximate solution of problems involving partial differential equations include finite difference schemes, finite elements, finite volume techniques, boundary elements, and particle methods. Within the finite element method there have been various significant developments, including for example classical isoparametric mapping, see e.g. Hughes [15] and Zienkiewicz and Taylor [35], but also isogeometric schemes, see Cottrell et al. [10].

Research continues to be motivated by the goal of developing stable, efficient and robust discretization schemes for finite deformation applications in solid mechanics. Within this line also the virtual element technology is further refined and applied to nonlinear problems in mechanics, see e.g. Beirão da Veiga et al. [6], Chi et al. [9], Wriggers et al. [34], Artioli et al. [3], De Bellis et al. [11], Wriggers and Hudobivnik [33], Aldakheel et al. [1], Hussein et al. [17] and De Bellis et al. [12]. So far the assumptions—even for higher order virtual elements—contain the restriction to straight edges of the elements that are directly defined in the physical space, see e.g. Beirão da Veiga et al. [5]. This makes the definition of virtual elements having a general geometric shape more complicated which is due to the fact that mappings like the isoparametric map for finite elements or NURBS type maps in isogeometric analysis are not easily applicable. New formulations that introduce curved element edges allow specific geometrical forms of the course of the curve at the edges were discussed in Beirão da Veiga et al. [7], Artioli et al. [4] and Aldakheel et al. [2].

This paper introduces a possibility to employ general mappings also within the virtual element formulation. The idea is to map a virtual element, defined at a reference configuration, to a general shape in the initial (physical) configuration. With such a map *general* shapes of virtual elements can be created in the initial configuration while *preserving* the straight edges in the reference configuration. This type of mapping was investigated for shell problems in e.g. Pimenta and Campello [27], but so far has not been used in two- and three-dimensional solid mechanics and within the virtual element methodology.

The method is developed for hyperelastic materials undergoing finite strains. However it can as well be applied for small strain cases as for material nonlinearities such as plasticity. Here we use the neo-Hookean strain energy as a model. The mapping from the reference to the initial configuration is performed either with an *isoparametric map* based on a quadratic ansatz for the geometry or a *Bezier type map* using NURBS: Both formulations perform extremely well in a series of benchmark tests involving regular and Voronoi meshes.

After presenting the governing equations for nonlinear elasticity in Sect. 2. Section 3 is devoted to the general mapping procedure, followed by the formulation of a low order

✉ P. Wriggers
  wriggers@ikm.uni-hannover.de

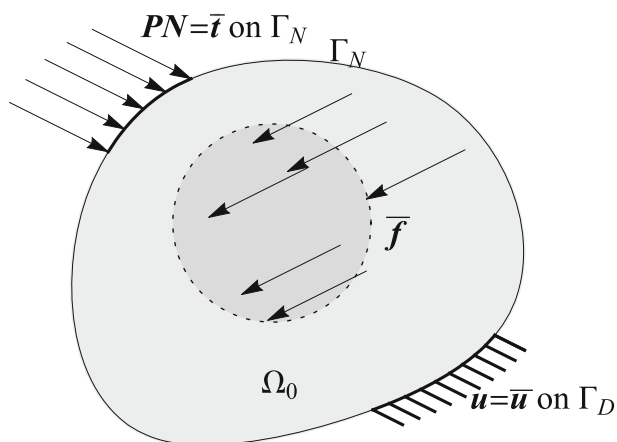[1] Institute for Continuum Mechanics, Leibniz Universität, Hannover, Germany

Fig. 1 Solid with boundary conditions

virtual element method in Sect. 4 that takes into account the general mapping. A number of numerical examples are presented and discussed in Sect. 5, and in Sect. 6 some concluding remarks are presented.

2 Governing equations for finite elasticity

Consider an elastic body that occupies the bounded domain $\Omega_0 \subset \mathbb{R}^2$. The body Ω_0 has a boundary Γ which comprises non-overlapping sections Γ_D and Γ_N such that $\Gamma_D \cup \Gamma_N = \Gamma$ (Fig. 1).

The position x of a material point initially at X is given by the motion

$$x = \varphi(X, t) = X + u(X, t) \tag{1}$$

where u is the displacement field that depends generally on the time t. We also define the deformation gradient F by

$$F(u) = 1 + \text{Grad}\, u(X, t), \tag{2}$$

the gradient being evaluated with respect to X. The body satisfies for the static case, on Ω_0 the equation of equilibrium

$$\text{Div}\, P + \bar{f} = 0, \tag{3}$$

with the body force $\bar{f}$ and the first Piola–Kirchhoff stress P. The Dirichlet and Neumann boundary conditions are respectively

$$u = \bar{u} \quad \text{on } \Gamma_D, \tag{4}$$

$$PN = \bar{t} \quad \text{on } \Gamma_N, \tag{5}$$

with N the outward unit normal vector, $\bar{u}$ the prescribed displacement, and $\bar{t}$ the prescribed surface traction on Γ_N.

By introducing a strain energy function $\Psi(u)$ for elastic problems the first Piola–Kirchhoff stresses follow from

$$P(u) = \frac{\partial \Psi(u)}{\partial F}. \tag{6}$$

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

$$\begin{aligned} \Psi(u) = &\frac{\lambda}{4}(J(u)^2 - 1 - 2\ln J(u)) \\ &+ \frac{\mu}{2}(\text{tr}\, C(u) - 2 - 2\ln J(u)) \end{aligned} \tag{7}$$

in which λ and μ are the Lamé constants. This strain energy is known as Neo-Hookean model. The right Cauchy-Green tensor $C(u)$ is defined as $C = F^T F$ and the Jacobian $J(u)$ of the deformation is given as $J = \det F > 0$.

In case of a hyperelastic material the development of a virtual element can start from the potential energy function directly instead of using the weak form, see e.g. Wriggers et al. [34]. In that case the potential energy can be written with (7) as

$$U(u) = \int_{\Omega_0} \left[\Psi(u) - \bar{f} \cdot u \right] d\Omega - \int_{\Gamma_N} \bar{t} \cdot u \, d\Gamma \tag{8}$$
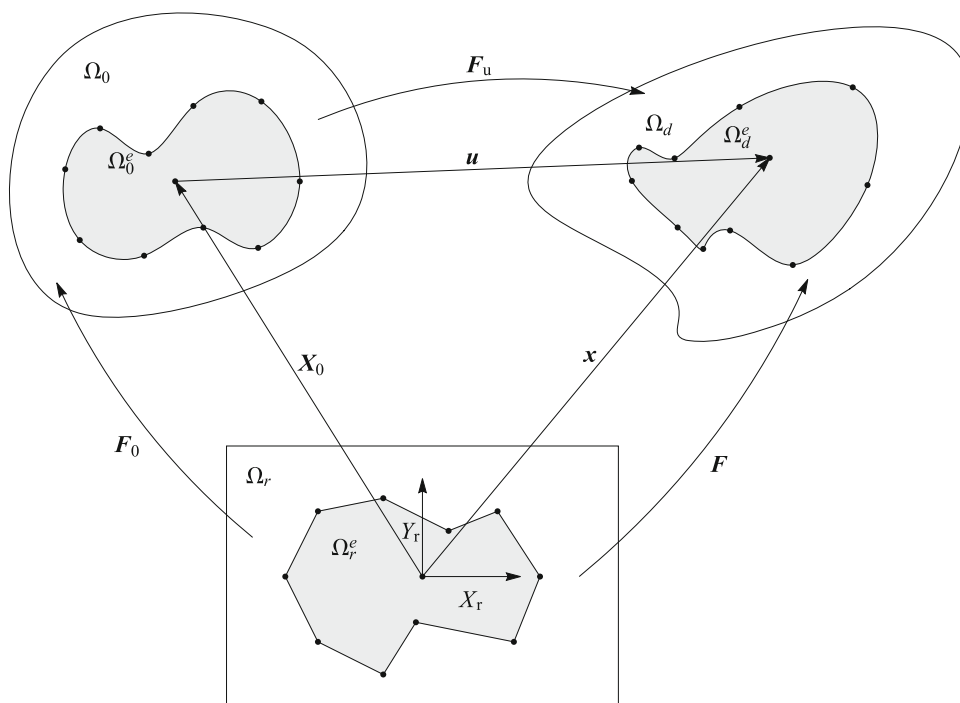
3 General mapping

The idea for the construction of general shaped virtual elements is to use a function that maps the coordinates X_r of a virtual element from a reference configuration Ω_r to the coordinates X_0 of the initial configuration Ω_0, see Fig. 2. The deformation is then given by the vector x which describes the current configuration Ω_d that a solid assumes under loading.

The mapping from the reference to the initial configuration can be defined in an arbitrary way using either an isoparametric map or e.g. NURBS functions. For isoparametric maps we can use shape function N_I that describe a polynomial basis in the reference configuration. The map is given by

$$X_0(X_r, Y_r) = \sum_{I=1}^{n_{iso}} N_I(X_r, Y_r) X_I^0 \tag{9}$$

In this paper we use a quadratic function for the isoparametric map, explicit expressions for the ansatz functions can be found in e.g. Hughes [15], Wriggers [32] and Onate [25]. The

Fig. 2 Configurations of a solid including map to initial configuration

coordinates of the reference configuration are here selected as $(X_r, Y_r) \in [-1, 1] \times [-1, 1]$.[1]

A more general mapping that includes also mapping onto domains that cannot be exactly defined by polynomials is provided by NURBS, see e.g. Farin [13] and Piegl and Tiller [26]. These Non-Uniform Rational B-Splines are obtained for a two-dimensional map by a projection of a B-spline quantity from three-dimensional space which allows exact constructions of geometrical objects such as circles and ellipses. The NURBS projection maps from a parameter space Ω_r with $0 \le \eta^1 \le 1$ and $0 \le \eta^2 \le 1$ to the initial configuration Ω_0 follows as

$$X_0(\eta^1, \eta^2) = \sum_{I=0}^{m} \sum_{J=0}^{n} R_{IJ}(\eta^1, \eta^2) P_{IJ}^0 \tag{10}$$

where P_{IJ}^0 are the control points that define the domain in the initial configuration. The functions $R_{IJ}(\eta^1, \eta^2)$ are the rational B-spline basis functions. They are explicitly given by

$$R_{IJ}(\eta^1, \eta^2) = \frac{w_{IJ} N_I^1(\eta^1) N_J^2(\eta^2)}{\sum_{I=0}^{m} \sum_{J=0}^{n} w_{IJ} N_I^1(\eta^1) N_J^2(\eta^2)} \tag{11}$$

with the weights w_{IJ}. For more details related to the order of the B-spline basic functions we refer to Piegl and Tiller

[1] When using different arbitrary rectangular domains as reference domain one would need an extra linear mapping $(X_r, Y_r) \in [X_{min}, X_{max}] \times [Y_{min}, Y_{max}]$.

[26], Hughes et al. [16] and Cottrell et al. [10]. The latter used this parametrization to develop the discretization technique known as isogemetrical analysis. We note that the NURBS functions can also be linearly mapped to $(X_r, Y_r) \in [X_{min}, X_{max}] \times [Y_{min}, Y_{max}]$. Thus we will write the mapping for both, isoparametric and NURBS, mapping as being a function of the coordinates $X_r = (X_r, Y_r)$.

According to Fig. 2 the deformation gradient in (2) can be computed via

$$F = F_u F_0 \tag{12}$$

with

$$F_0 = \text{Grad } X_0 \tag{13}$$

following from the mapping functions (9) and (10). The Jacobian of this mapping is given by $J_0 = \det F_0$. Furthermore the volume element in the initial configuration is connected to a volume element in the reference configuration by $d\Omega_0 = J_0 \, d\Omega_r$. Depending on the used mapping we either have to differentiate in (13) with respect to (X_r, Y_r) or (η^1, η^2).

The deformation gradient F_u that describes the deformation from the initial to the current configuration yields

$$F_u = F F_0^{-1}. \tag{14}$$

Based on these general kinematical quantities we can compute the right Cauchy Green tensor in terms of the deformation gradient F_u as

Fig. 3 VEM element and boundary discretization in Ω_r
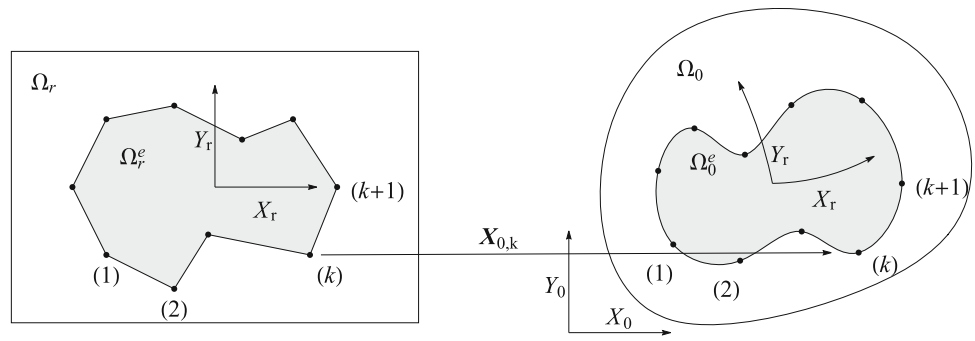
$$C_u = F_u^T F_u = F_0^{-T} F^T F F_0^{-1}. \tag{15}$$

Now the potential (8) can be reformulated using (15)

$$U(\boldsymbol{u}) = \int\limits_{\Omega_r} \left[\Psi(C_u) - \bar{\boldsymbol{f}} \cdot \boldsymbol{u} \right] J_0 \, d\Omega_r - \int\limits_{\Gamma_{Nr}} \bar{\boldsymbol{t}} \cdot \boldsymbol{u} \, \Gamma_0 \, d\Gamma \tag{16}$$

where in the general three-dimensional case

$$\Gamma_0 = \| \boldsymbol{X}_{0,X_r} \times \boldsymbol{X}_{0,Y_r} \| \tag{17}$$

is the mapping of the area elements of the reference to the initial configuration, either with respect to the coordinates of the isoparametric or the NURBS mapping, respectively. Furthermore, the Jacobian J describing the volume change in the strain energy (7) is given by $J_u = J J_0^{-1}$.

In the two-dimensional case one can compute the gradient of the mapping from the reference to the initial configuration directly from the isoparametric or the NURBS map as

$$F_0^{iso} = \sum_{I=1}^{n_{iso}} \begin{bmatrix} N_{I,x} X_I^0 & N_{I,y} X_I^0 \\ N_{I,x} Y_I^0 & N_{I,y} Y_I^0 \end{bmatrix} \quad \text{or}$$

$$F_0^N = \sum_{I=0}^{m} \sum_{J=0}^{n} \begin{bmatrix} R_{IJ,\eta^1} P_{xIJ}^0 & R_{IJ,\eta^2} P_{xIJ}^0 \\ R_{IJ,\eta^1} P_{yIJ}^0 & R_{IJ,\eta^2} P_{yIJ}^0 \end{bmatrix}. \tag{18}$$

The differentiations in (18) have to be performed with respect to the reference coordinates. Note that the computation of the derivatives of the rational B-splines is more involved due to the definition of the B-spline in (11).

Note that description of a solid may require more than one reference configurations. In this case it is possible to combine several configurations to model the solid which will be shown in the numerical example section.

4 Formulation of the virtual element method

The main idea of the virtual element method (VEM) is a projection of the deformation onto a specific ansatz space. In classical analysis using virtual elements a domain Ω is partitioned into non-overlapping polygonal elements which need not be convex. Here we extend this possibility by mapping the polygons into other geometrical objects using a nonlinear map as discussed in the last section. In this contribution we restrict ourselves to two dimensional solids and to a low-order approach using linear ansatz functions. Due to the construction of the mappings the developed methodology works without any changes for higher order ansatz spaces of virtual elements.

The construction of a low order virtual element is based on a linear ansatz space $\boldsymbol{u}_h$, see e.g. Beirão da Veiga et al. [5], that has unknown displacements $\boldsymbol{u}_k$ at the vertices k of the polygon, a linear ansatz for the displacement field $\boldsymbol{u}_h$ at the edges of the polygon and the property that $\mathrm{Div}(\nabla \boldsymbol{u}_h) = 0$. The polygonal elements can have arbitrary number of vertices where n_V is the total number. Furthermore, the element shape is restricted to straight edges, see Fig. 3, in the reference configuration Ω_r^e.

4.1 Ansatz functions

Generally the virtual element method relies on the split of the ansatz space for the displacement field $\boldsymbol{u}_h$ into a part $\boldsymbol{u}_\pi$ and a remainder

$$\boldsymbol{u}_h = \boldsymbol{u}_\pi + (\boldsymbol{u}_h - \boldsymbol{u}_\pi) \tag{19}$$

Classically the projection $\boldsymbol{u}_\pi$ is defined at element level by a function which is directly formulated in the coordinates that describe the real geometry of the domain. Hence the ansatz for a linear polynomial function yields for the two dimensional case in the initial configuration

$$\boldsymbol{u}_\pi = \mathbf{H}^{2d} \mathbf{a} = \begin{bmatrix} 1 & 0 & X_0 & 0 & Y_0 & 0 \\ 0 & 1 & 0 & X_0 & 0 & Y_0 \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ \dots \\ a_6 \end{Bmatrix} \tag{20}$$
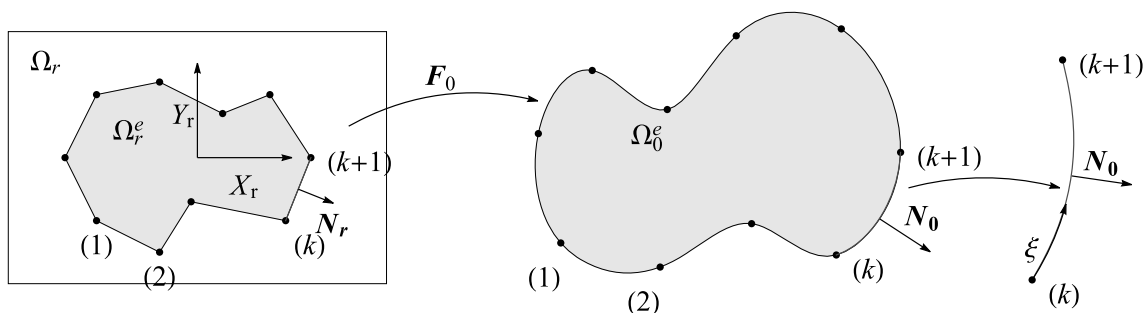
Fig. 4 VEM element and boundary discretization in Ω_0 and Ω_r using surface coordinate ξ

4.2 Computation of the projection

There exist several methods to obtain the parameters a_i in terms of the unknown nodal values of a virtual element Ω^e as depicted in Fig. 3. Here we use a Galerkin projection of the gradients to project u_π onto the ansatz u_h. Since this formulation does not rely on the mechanical weak form it leads to the same projection for linear and nonlinear elasticity problems. The Galerkin projection of the gradient ∇u_π is carried out in the initial configuration Ω_0^e of the virtual element. It can be formulated as

$$\int_{\Omega_0^e} (\nabla_0 u_h - \nabla_0 u_\pi) \cdot \nabla_0 N_\pi \, d\Omega_0 = 0. \tag{21}$$

Here the index "0" in ∇ refers to the differentiation with respect to the initial coordinates X_0. N_π is the vector containing a weighting polynomial that has the same order as u_π. For the chosen linear ansatz the gradient $\nabla_0 N_\pi$ is constant when we select $N_\pi = \{1, X_0, Y_0\}$. Thus we can write (21) as

$$\int_{\Omega_0^e} \nabla_0 u_\pi \, d\Omega_0 = \int_{\Omega_0^e} \nabla_0 u_h \, d\Omega_0. \tag{22}$$

Since $\nabla_0 u_\pi$ is constant in Ω_0^e for the linear ansatz in (20) we can write for the left side of the integral above as

$$\int_{\Omega_0^e} \nabla_0 u_\pi \, d\Omega_0 = \Omega_0^e \, \nabla_0 u_\pi. \tag{23}$$

Using now the Gauss theorem the second integral in (22) can be transformed to a boundary integral

$$\int_{\Omega_0^e} \nabla_0 u_h \, d\Omega_0 = \int_{\Gamma_0^e} u_h \otimes N_0 \, d\Gamma_0 \tag{24}$$

with N_0 being the outward normal to the element area Ω_0 in the initial configuration.

The integral on the right hand side in (24) can be transformed to the reference domain directly. The question is how to compute the normal vector N_0 at an edge $\gamma \in \Gamma_0$ where γ stands for an edge between two vertices and Γ_0 is the entire boundary of the virtual element. Two different possibilities are available for the transformation:

- We can introduce along an edge γ the coordinate ξ, see Fig. 4. The normal vector N_0 can then be discretized directly using the map (think of convective coordinates)

$$N_0 = \frac{X_{0,\xi}^\gamma \times E_3}{\|X_{0,\xi}^\gamma\|} = \frac{1}{\|X_{0,\xi}^\gamma\|} \left\{ \begin{array}{c} -Y_{0,\xi} \\ X_{0,\xi} \end{array} \right\} \tag{25}$$

Where the edge γ_0 is defined through the reference configuration by

$$X_r^\gamma = \frac{1}{2}(1 - \xi) X_k + \frac{1}{2}(1 + \xi) X_{k+1} \tag{26}$$

which yields the coordinate at the edge $X_0(X(\xi))$ as a function of ξ.

Now we define a linear ansatz for u_h at the edge γ_0 as

$$u_h(\xi) = \frac{1}{2}(1 - \xi) u_k + \frac{1}{2}(1 + \xi) u_{k+1}. \tag{27}$$

With $d\Gamma_0^\gamma = \|X_{0,\xi}^\gamma\| d\xi$ we can write the integral as a sum over all n_γ edges $\gamma \in \Gamma_0$ of the virtual element

$$\int_{\Omega_0^e} \nabla_0 u_h \, d\Omega_0 = \sum_{\gamma=1}^{n_\gamma} \int_\xi u_h(\xi) \otimes N_0(\xi) \|X_{0,\xi}^\gamma\| \, d\xi$$

$$= \sum_{\gamma=1}^{n_\gamma} \int_\xi u_h(\xi) \otimes \left\{ \begin{array}{c} -Y_{0,\xi} \\ X_{0,\xi} \end{array} \right\}^\gamma d\xi \tag{28}$$

Note that the coordinate ξ is related to a line in Ω_r that is straight, see right side in Fig. 4.

For a specific map $\boldsymbol{X}_0$ the integration of the right hand side in (28) follows by using an integration rule, e.g. Gauss integration,

$$\sum_{\gamma=1}^{n_\gamma} \int_\xi \boldsymbol{u}_h(\xi) \otimes \left\{ \begin{matrix} -Y_{0,\xi} \\ X_{0,\xi} \end{matrix} \right\}^\gamma d\xi$$

$$= \sum_{\gamma=1}^{n_\gamma} \sum_{g=1}^{n_g} w_g \left[\boldsymbol{u}_k \otimes \left\{ \begin{matrix} -Y_{0,\xi}(\xi_g) \\ X_{0,\xi}(\xi_g) \end{matrix} \right\}^\gamma \right] \tag{29}$$

where w_g are the weights and ξ_g the integration points with respect to the reference configuration. This integration holds for isoparametric and NURBS mappings. The integration order in (29) depends on the mapping function, e.g for a quadratic isoparametric map the integrand in (29) is represented by a second order polynomial, thus a two point Gauss integration is sufficient. But for a Bezier type mapping the number of Gauss points has to be increased.

- Another way to integrated the right hand side in (24) is to use Nanson's formula. Then the integral can be transferred to the reference configuration by

$$\int_{\Gamma_0^e} \boldsymbol{u}_h \otimes \boldsymbol{N}_0 \, d\Gamma_0 = \sum_{\gamma=1}^{n_\gamma} \int_{\gamma_r} \boldsymbol{u}_h \otimes \boldsymbol{F}_0^{-T} \boldsymbol{N}_r^\gamma \, \det \boldsymbol{F}_0 \, d\gamma_r \tag{30}$$

where now $\boldsymbol{N}_r$ is the normal related to the straight edges of the virtual element in the reference configuration. The inverse of this gradient is given for the isoparametric map in the two-dimensional case simply by

$$\boldsymbol{F}_0^{-T} = \frac{1}{\det \boldsymbol{F}_0} \sum_{I=1}^{n_{iso}} \begin{bmatrix} N_{I,y} Y_I^0 & -N_{I,x} X_I^0 \\ -N_{I,y} Y_I^0 & N_{I,x} X_I^0 \end{bmatrix} \tag{31}$$

with $\{X_I^0, Y_I^0\} = \boldsymbol{X}_0$. By defining

$$\boldsymbol{G}_0^T = \boldsymbol{F}_0^{-T} \det \boldsymbol{F}_0 = \sum_{I=1}^{n_{iso}} \begin{bmatrix} N_{I,y} Y_I^0 & -N_{I,x} X_I^0 \\ -N_{I,y} Y_I^0 & N_{I,x} X_I^0 \end{bmatrix} \tag{32}$$

the integral in (30) yields

$$\int_{\Gamma_0^e} \boldsymbol{u}_h \otimes \boldsymbol{N}_0 \, d\Gamma_0 = \sum_{\gamma=1}^{n_\gamma} \int_{\gamma_r} \boldsymbol{u}_h \otimes \boldsymbol{G}_0^T \boldsymbol{N}_r^\gamma \, d\gamma_r \tag{33}$$

One can combine (23) and (33) to finally obtain the projection of the gradient

$$\nabla_0 \boldsymbol{u}_\pi = \frac{1}{\Omega_0} \sum_{\gamma=1}^{n_\gamma} \int_{\gamma_r} \boldsymbol{u}_h \otimes \boldsymbol{G}_0^T \boldsymbol{N}_r^\gamma \, d\gamma_r, \tag{34}$$

or use (23) and (28) which yields the alternative form

$$\nabla_0 \boldsymbol{u}_\pi = \frac{1}{\Omega_0} \sum_{\gamma=1}^{n_\gamma} \int_\xi \boldsymbol{u}_h(\xi) \otimes \left\{ \begin{matrix} -Y_{0,\xi} \\ X_{0,\xi} \end{matrix} \right\}^\gamma d\xi. \tag{35}$$

In (34) and (35) the area Ω_0 in the initial configuration is computed using the reference configuration of the virtual element. The integral can be evaluated over the edges using partial integration and Gauss integration

$$\Omega_0 = \int_{\Omega_r} J_0 \, d\Omega_r = \frac{1}{2} \sum_{\gamma=1}^{n_\gamma} \int_{\gamma_r} \boldsymbol{X}_r \cdot [\boldsymbol{G}_0^T \boldsymbol{N}_r^\gamma] d\gamma_r \tag{36}$$

$$= \frac{1}{2} \sum_{\gamma=1}^{n_\gamma} \sum_{g=1}^{n_g} l_r^\gamma w_g \boldsymbol{X}_r(\xi_g) \cdot \left[\boldsymbol{G}_0^T(\xi_g) \boldsymbol{N}_r^\gamma(\xi_g) \right]. \tag{37}$$

All quantities are related to the reference configuration, see Fig. 4. These are the coordinates $\boldsymbol{X}_r$, the normal $\boldsymbol{N}_r^\gamma$ at edge γ, the Gauss points ξ_g and associated weights w_g and the length of an edge l_r^γ. Note that also the first formulation leading to (35) can be applied to compute Ω_0.

Finally the projection in (34) can be evaluated by inserting ansatz (27) into this equation. Again Gauss integration is employed

$$\nabla_0 \boldsymbol{u}_\pi = \frac{1}{\Omega_0} \sum_{\gamma=1}^{n_\gamma} \sum_{g=1}^{n_g} l_r^\gamma w_g \boldsymbol{u}_h(\xi_g) \otimes \left[\boldsymbol{G}_0^T(\xi_g) \boldsymbol{N}_r^\gamma(\xi_g) \right] \tag{38}$$

which yields the parameters a_3 to a_6 in (20) that now depend on the displacements $\boldsymbol{u}_k$ at the vertices k of a virtual element. Again (35) can also be used to compute the projected gradient.

In the projection in (38) only the gradient plays a role. This allows to compute the parameters a_3 to a_6 of (20) directly. Hence the constant parts of the ansatz disappear. Thus the parameters a_1 and a_2, related to the constant parts, have to be obtained in a different way. Here the average displacement of the virtual element in the initial configuration along the edge Γ_0 is used. The equivalence

$$\frac{1}{\Gamma^e} \int_{\Gamma_0^e} \boldsymbol{u}_\pi \, d\Gamma_0 = \frac{1}{\Gamma_e} \int_{\Gamma_0^e} \boldsymbol{u}_h \, d\Gamma_0 \tag{39}$$

includes the parameters a_1 and a_2 and from (39) all parameters in (20) can be computed. A simplified expression follows, if the integrals are evaluated by the Gauss–Lobatto rule. In that case the nodal values of $\boldsymbol{u}_\pi$ and $\boldsymbol{u}_h$ appear in a sum over all vertices. This leads to

$$\frac{1}{n_v} \sum_{k=1}^{n_v} \boldsymbol{u}_\pi(\boldsymbol{X}_{0\,k}) = \frac{1}{n_v} \sum_{k=1}^{n_v} \boldsymbol{u}_k \qquad (40)$$

Ansatz (20) can now be inserted into (40) and evaluated at $\boldsymbol{X}_{0k} = \sum_{I=1}^{n_{iso}} N_I(X_{rk}, Y_{rk}) \boldsymbol{X}_I^0$ for the left hand side, while on the right hand side only the nodal values of $\boldsymbol{u}_h$ have to be used. This finally leads to two equations for the parameters a_1 and a_2.

4.3 Construction of the virtual element

In this contribution a linear ansatz is employed for the discretization using virtual elements. Thus the gradient of the displacement field is approximated by a constant part as a result of the projection in the last section. A construction of a virtual element which is based only on this projection would lead to a rank deficient element once the number of vertices is greater than 3. Thus the formulation has to be stabilized, as in the case of the classical one-point integrated elements developed by Flanagan and Belytschko [14], Belytschko and Bindeman [8], Reese et al. [30], Reese and Wriggers [29], Reese [28], Mueller-Hoeppe et al. [24], Korelc et al. [21] and Krysl [22], to mention some key contributions.

The formulation is provided here for finite deformation formulations. The potential U is a split into a constant part of the deformation gradient and an associated stabilization term. In this contribution we employ the hyperelastic potential function (16) as basis for the virtual element. By summing up all element contributions for the n_e virtual elements that discretize a domain Ω_0 one obtains

$$U(\boldsymbol{u}) \simeq \mathop{\mathbf{A}}_{e=1}^{n_e} \left[U_c^e(\boldsymbol{u}_\pi) + U_{stab}^e(\boldsymbol{u}_h - \boldsymbol{u}_\pi) \right]. \qquad (41)$$

In the following we will first discuss the formulation of the consistency part $U_c^e(\boldsymbol{u}_\pi)$ that stems from the projection, see last section. Furthermore, a possibility for the stabilization U_{stab}^e of the virtual element method will be considered.

4.3.1 Constant part due to projection

The consistency part in the potential (41) can be evaluated by inserting the results obtained in the last section. This yields

with respect to the initial configuration for a virtual element Ω_0^e

$$U_c^e(\boldsymbol{u}_\pi) = \int_{\Omega_0^e} \left[\Psi(\boldsymbol{C}_u) - \bar{\boldsymbol{f}} \cdot \boldsymbol{u}_\pi \right] d\Omega_0 - \int_{\Gamma_0^e} \bar{\boldsymbol{t}} \cdot \boldsymbol{u}_\pi \, d\Gamma_0. \qquad (42)$$

Due to the fact that we are able to compute the projection of the gradient $\nabla_0 \boldsymbol{u}_\pi$ in the initial configuration directly in (38) we obtain the deformation gradient describing the map from the initial to the current configuration as

$$\boldsymbol{F}_u = \boldsymbol{1} + \nabla_0 \boldsymbol{u}_\pi \qquad (43)$$

which is constant. Note that here also (35) could have been inserted for $\nabla_0 \boldsymbol{u}_\pi$ which leads to exactly the same results. Now the right Cauchy-Green tensor is given by

$$\boldsymbol{C}_u = \boldsymbol{F}_u^T \boldsymbol{F}_u \qquad (44)$$

which is also constant within the virtual element Ω_e but depends in a nonlinear fashion on the nodal displacements.

The strain energy of the hyperelastic material

$$\Psi(\boldsymbol{u}_\pi) = \frac{\lambda}{4} [\, J_u^2 - 1 - 2 \ln J_u \,] + \frac{\mu}{2} [\, \mathrm{tr}\, \boldsymbol{C}_u - 2 - 2 \ln J_u \,] \qquad (45)$$

can now be computed in terms of the projected deformation measures which are the right Cauchy-Green tensor $\boldsymbol{C}_u$ and the Jacobian J_u. The latter is given by

$$J_u = \det \boldsymbol{F}_u = \det(\boldsymbol{1} + \nabla_0 \boldsymbol{u}_\pi). \qquad (46)$$

Hence all quantities in (45) depend only the projection $\nabla_0 \boldsymbol{u}_\pi$.

The first integral in (42) can easily be computed since $\Psi(\boldsymbol{C}_u)$ is constant in (X_0, Y_0) and Ω_0^e known from (37). The two loading terms have to be transformed for evaluation to the reference configuration, see also (28),

$$U_c^e(\boldsymbol{u}_\pi) = \Omega_0^e \, \Psi(\boldsymbol{C}_u) - \int_{\Omega_r^e} \bar{\boldsymbol{f}} \cdot \boldsymbol{u}_\pi \, \det \boldsymbol{F}_0 \, d\Omega_r$$

$$- \sum_{\gamma=1}^{n_\gamma} \int_\xi \bar{\boldsymbol{t}}_\gamma \cdot \boldsymbol{u}_\pi \, \| X_{0,\xi}^\gamma \| \, d\xi. \qquad (47)$$

where γ denotes the edges of the virtual element that are loaded by surface traction $\bar{\boldsymbol{t}}_\gamma$.

Once the integration over the reference coordinate is finalized, the derivations of the potential (47) with respect to the n_V unknown nodal displacements at the vertices of the virtual element $\boldsymbol{u}_e = \{ \boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{n_V} \}$ can be carried out. This approach exploits the fact that $\nabla_0 \boldsymbol{u}_\pi$ depends directly
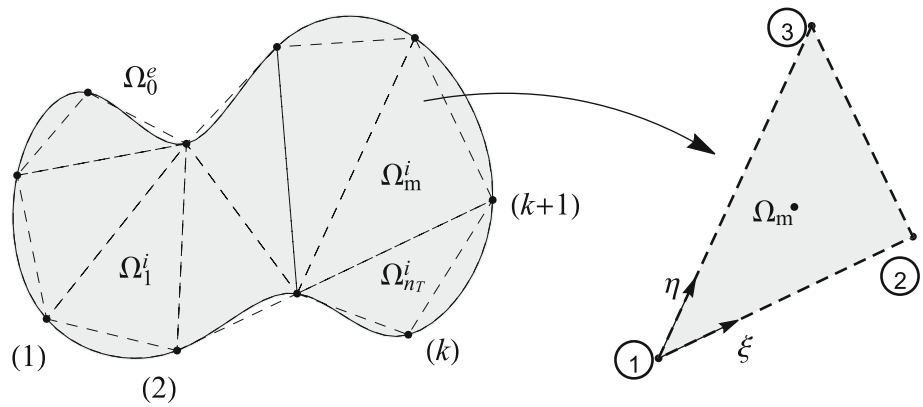
on the nodal displacements, see (35) and (38), and the displacements $\boldsymbol{u}_\pi$ are linked via (20) and (40) to the nodal displacements as well.

All derivations, leading to the residual vector $\boldsymbol{R}_e^c$ and the tangent matrix $\boldsymbol{K}_{Te}^c$, were performed with the symbolic tool AceGen, see Korelc and Wriggers [20]. This yields for (47)

$$\boldsymbol{R}_e^c = \frac{\partial U_c^e(\nabla_0 \boldsymbol{u}_\pi(\boldsymbol{u}_e), \boldsymbol{u}_\pi(\boldsymbol{u}_e))}{\partial \boldsymbol{u}_e} \quad \text{and} \quad \boldsymbol{K}_{Te}^c = \frac{\partial \boldsymbol{R}_e^c(\boldsymbol{u}_e)}{\partial \boldsymbol{u}_e} \tag{48}$$

where $\boldsymbol{u}_e$ are the nodal displacements of the virtual element Ω_e at the vertices.

4.3.2 Stabilization techniques for nonlinear virtual elements

In the literature on virtual element technologies basically two different stabilization techniques were discussed that work well for classical solid mechanics problems. The first stabilization is directly based on the degrees of freedom. It introduces a point wise error measure between the nodal values $\boldsymbol{u}_k$ and the approximation function $\boldsymbol{u}_\pi$ evaluated at the vertices $\boldsymbol{X}_k$, see e.g. Beirão da Veiga et al. [5], Beirão da Veiga et al. [6] and Chi et al. [9].

Here we use the stabilization approach developed in Wriggers et al. [34] for virtual elements. The essence of the approach is to introduce a new, positive definite strain energy $\hat{\Psi}$ and to define the stabilization contribution to the strain energy by

$$U_{stab}(\boldsymbol{u}_h - \boldsymbol{u}_\pi) = \hat{U}(\boldsymbol{u}_h) - \hat{U}(\boldsymbol{u}_\pi). \tag{49}$$

Such a stabilization was also used in Krysl [22] for stabilized mean strain formulations of finite elements. The second term on the right side ensures the consistency of the total potential energy, which is now given by

$$U(\boldsymbol{u}_h) \simeq U_c(\boldsymbol{u}_\pi) + \hat{U}(\boldsymbol{u}_h) - \hat{U}(\boldsymbol{u}_\pi). \tag{50}$$

Actually $\hat{U}$ can be selected differently from the strain energy that describes the constitutive behavior of the solid, see e.g. Wriggers et al. [34], since it is only for stabilization. Note that $\hat{U}(\boldsymbol{u}_h) - \hat{U}(\boldsymbol{u}_\pi)$ disappears for element sizes $\Omega_0^e \to 0$. In this paper we use the same energy Ψ as in the consistency part U_c, however with different Lamé parameters $\hat{\lambda}$ and $\hat{\mu}$. The stabilization energy follows then by assembly over all virtual elements

$$\hat{U}(\boldsymbol{u}_h) = \overset{n_e}{\underset{e=1}{\mathbf{A}}} \int_{\Omega_0^e} \hat{\Psi}(\boldsymbol{u}_h) \, d\Omega_0. \tag{51}$$

The computation of the consistency part $U_c(\boldsymbol{u}_\pi)$ was already discussed above. The last part of the potential energy $\hat{U}(\boldsymbol{u}_\pi)$ depends also on $\boldsymbol{u}_\pi$, thus it can be evaluated as the consistency part, see Sec. 4.3.1.

The question is now how to compute the stabilization term $\hat{U}(\boldsymbol{u}_h)$ since $\boldsymbol{u}_h$ is not known within the virtual element. The idea is to use an internal triangular mesh. This inscribed triangular finite element mesh, see Fig. 5, consists of n_{int} linear three-noded triangles that are connected to the nodes of the virtual element. Hence this internal mesh does not creat new nodal unknowns. The ansatz functions for the triangles are then assumed to approximate $\boldsymbol{u}_h$ in (19).

For this discretization of the potential $\hat{U}^e(\boldsymbol{u}_h)$ within a virtual element Ω_0^e one has to add all contributions of the internal triangles Ω_m^i

$$\hat{U}^e(\boldsymbol{u}_h) = \sum_{m=1}^{n_{int}} \int_{\Omega_{0m}^i} \left[\frac{\hat{\lambda}}{4} [\, J_m^2 - 1 - 2 \ln J_m \,] \right.$$
$$\left. + \frac{\hat{\mu}}{2} [\, \text{tr}\, \boldsymbol{C_m} - 2 - 2 \ln J_m \,] \right] d\Omega_0, \tag{52}$$

which can be directly formulated with respect to the initial configuration Ω_{0m}.

To fulfill the condition that $\hat{U}(\boldsymbol{u}_h) - \hat{U}(\boldsymbol{u}_\pi) \to 0$ for small virtual elements $\Omega_0^e \to 0$ one has to compute the poten-

tial $\bar{U}(\boldsymbol{u}_\pi)$ using the area related to the triangular elements $\Omega_{OT}^e = \sum_{m=1}^{n_{\text{int}}} \Omega_{0m}^i$ which leads to

$$\bar{U}(\boldsymbol{u}_\pi) = \Omega_{OT}^e \, \hat{\Psi}(\boldsymbol{C}_u). \tag{53}$$

Due to the straight edges of the inscribed triangular mesh, see Fig. 5, this slightly differs from the way the element area is computed in the consistency part in (47). However such approximation can be admitted since (53) is only the stabilization energy which approaches zero for fine meshes, see above.

All further derivations leading to the residual vector $\boldsymbol{R}_e^s$ and the tangent matrix $\boldsymbol{K}_{Te}^s$ were performed in the same manner as in (48) with the symbolic tool ACEGEN, see Korelc and Wriggers [20]. This yields for (49)

$$\boldsymbol{R}_e^s = \frac{\partial U_{stab}(\boldsymbol{u}_e)}{\partial \boldsymbol{u}_e} \quad \text{and} \quad \boldsymbol{K}_{Te}^s = \frac{\partial \boldsymbol{R}_e^s(\boldsymbol{u}_e)}{\partial \boldsymbol{u}_e}. \tag{54}$$

Thus the final residual and tangent matrix of the virtual element are given by the sum of expressions (48) and (54): $\boldsymbol{R}_e = \boldsymbol{R}_e^c + \boldsymbol{R}_e^s$ and $\boldsymbol{K}_{Te} = \boldsymbol{K}_{Te}^c + \boldsymbol{K}_{Te}^s$.

The values of the Lamé parameters in the strain energy (52) have to be selected in a proper way. Krysl [23] suggested for cuboid finite elements a procedure that is based on a comparison of the bending energy of a thick beam and the bending energy of the finite element. This yields Lamé parameters $\hat{\lambda}$ and $\hat{\mu}$ that enhance the bending behavior of the element. Since this procedure is not directly applicable to arbitrary virtual elements we suggest a simple and computationally efficient way to compute the parameters from the basic geometric data of the virtual element in the initial configuration Ω_0^e, see Fig. 6. The procedure was described in detail in Wriggers et al. [34] and will not be reported here. We just state the final results. The Young's modulus is corrected by

$$\hat{E} = \frac{\beta}{1 + \beta} \, E = \phi \, E \tag{55}$$

while the Poisson ratio $\hat{\nu}$ is kept constant as $\hat{\nu} = 0.3$ since the Poisson ratio does not influence the convergence behavior of the element and avoids locking in the stabilization term for incompressible problems. The factor β depends on the height to length ratio of the brick element. For a virtual element with arbitrary shape this ratio can be approximated by using the inner and outer radii, R_i^2, R_a^2 respectively, see Fig. 6, to obtain

$$\beta = 2\sqrt{2} \, (1 + \nu) \frac{R_i^2}{R_a^2 - R_i^2}. \tag{56}$$
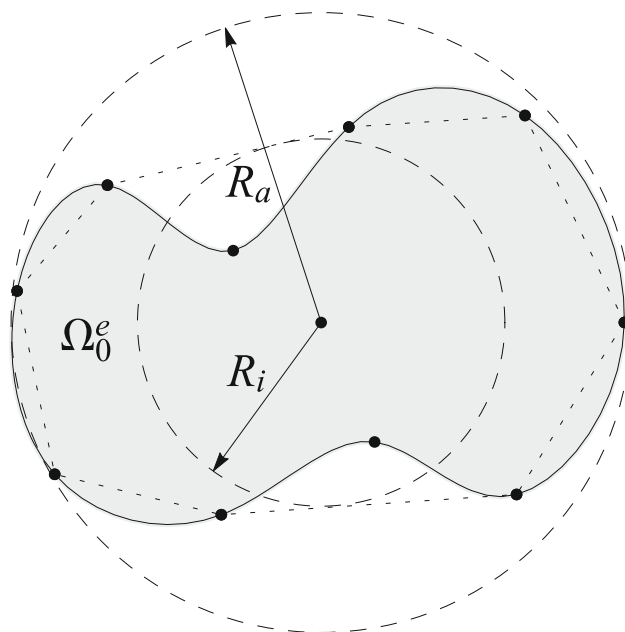

Fig. 6 Inner and outer radius of a virtual element

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. Another way to obtain β can be found in van Huyssteen and Reddy [31] which is based on employing ellipses instead of the circles in Fig. 6. Once the corrected Young's modulus and Poisson ratio are determined the Lamé parameters follow from

$$\hat{\lambda} = \frac{\hat{E} \, \hat{\nu}}{(1 + \hat{\nu})(1 - 2\hat{\nu})} \quad \hat{\mu} = \frac{\hat{E}}{2(1 + \hat{\nu})} \tag{57}$$

Remark From (55) the Lamé parameters can be written as $\hat{\lambda} = \phi \, \lambda$ and $\hat{\mu} = \phi \, \mu$, and with that also the strain energy as $\hat{\Psi} = \phi \, \Psi$. This means that we can re-write the strain energy parts in (41) for a virtual element Ω_0^e with (47) and (53) as

$$U_s^e(\boldsymbol{u}_h) = (\Omega_0^e - \phi \, \Omega_{0T}^e) \Psi(\boldsymbol{C}_u) + \phi \, U^e(\boldsymbol{u}_h). \tag{58}$$

where $U^e(\boldsymbol{u}_h)$ is the stabilization energy from (52) computed for the Lamé parameters λ and μ. Observe that for $\phi = 0$ only the energy related to the consistency part acts which would lead to a singular element. On the other hand for $\phi = \Omega_0^e / \Omega_{0T}^e$ the consistency part disappears and a pure finite element formulation based on the internal triangles is left over. In Wriggers et al. [34] it was shown that the latter case does not have the same good solution behavior as a virtual element with a factor ϕ computed from (55).
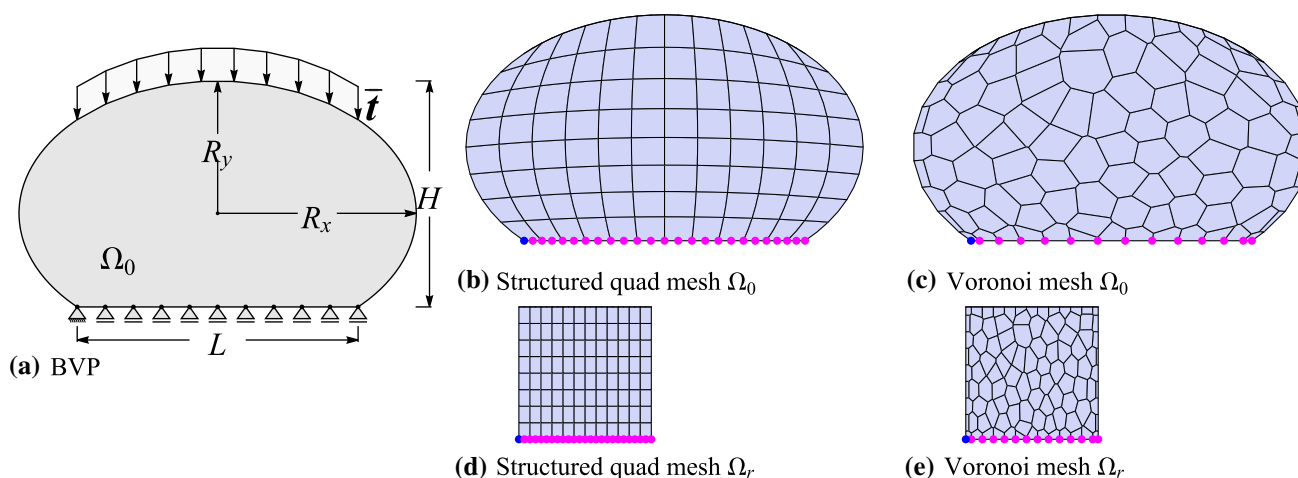
Fig. 7 Ellipsoidal plate, **a** Set-up of the problem, **b**–**e** structured and Voronoi meshes

5 Numerical examples

In this section we compare the new mapping procedure for virtual element discretizations with existing formulations. The main goal is to show that this formulation blends perfectly into the idea of virtual elements and produces the same results for higher order isoparametric or NURBS mappings. The examples are subjected to loads that lead to finite deformation strain states. For an efficient solution all numerical simulations of the nonlinear problems have to be performed by using a Newton–Raphson algorithm with load stepping when necessary. Due to the fact that all formulations are linearized in a consistent manner, using ACEGEN, see Korelc [18], quadratic convergence is achieved. All geometric and constitutive data in the examples are given in consistent units.

In all examples either NURBS (denoted by a MB) or isoparametric functions (denoted by a MN) were used for the map from the reference to the initial configuration. Simulations were conducted with different virtual elements using the software ACEFEM, see Korelc [19] and Korelc and Wriggers [20]:

- VEM-N4: virtual elements with four nodes and linear ansatz,
- VEM-N8: virtual elements with eight nodes and linear ansatz,
- VEM-VO: virtual elements within a Voronoi mesh with arbitrary number of nodes

The results of these discretizations were compared with linear Q1 and quadratic Q2 quadrilateral finite elements.

5.1 Squeezing of an ellipse

The solid structure shown in Fig. 7 is subjected to a traction load $\bar{t}_y = -300$ at the top. The ellipse is defined by its principal axes $R_x = 3$ and $R_y = 2$. The Lamé parameters, that govern the hyperelastic response of the solid, are given as $\lambda = 71.0719$ and $\mu = 36.6128$. The bottom part of the ellipse is flat, see Fig. 7a. It has the length $L = \sqrt{2}R_x$ and is fixed in y-direction. The left corner at the bottom is additionally fixed in x-direction. The total height of the ellipse is $H = (1 + \frac{1}{2}\sqrt{2})R_y$.

In Fig. 7b, c the structured and Voronoi meshes are shown, respectively, with respect to the initial configuration Ω_0. The associated meshes in the reference configuration Ω_r are displayed in Fig. 7d, e.

The von Mises stress distribution is depicted in Fig. 8 for a relatively coarse mesh of 8×12 elements. The discretizations in Fig. 8a, d show results that were computed with a standard finite element and virtual element discretization *without* mapping. The results depicted in Fig. 8b, c, e, f were obtained with the new mapping techniques from reference to initial configuration using either the isoparametric (M0) map or the NURBS (MB) approach. The results are almost equal for the standard virtual element formulation with 8 node elements (d) and the approach using M0 or MB mapping in (b) and (e) which shows that the mapping formulation works correctly. Here we do not expect much improvement of the results since there is for the linear ansatz only a small difference between the curved geometry and the straight edge approach of the classical virtual element scheme.
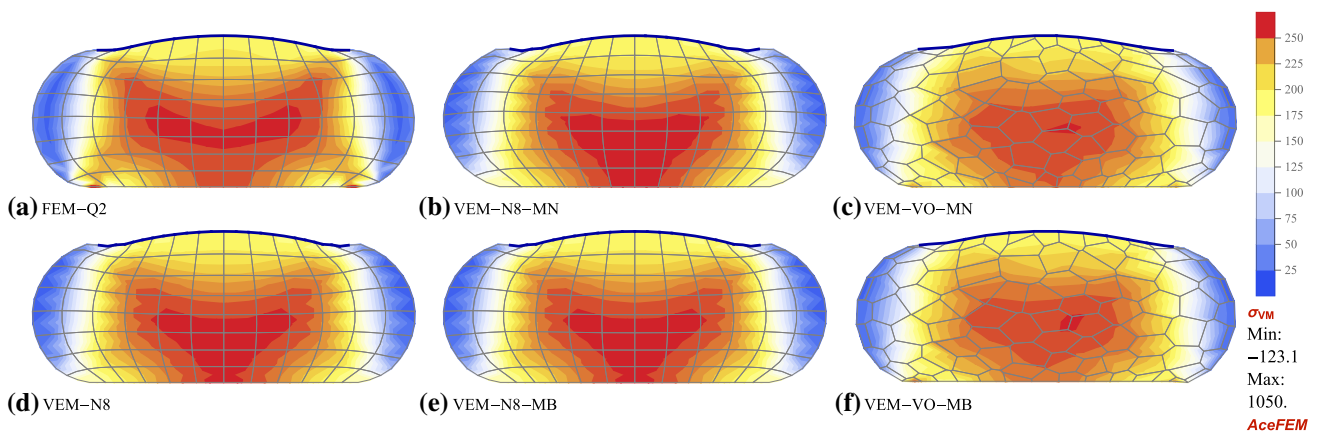
Fig. 8 Von Mises stress in the ellipsoidal structure for different formulations

5.2 Square plate with a hole

A plate with a hole, is depicted in Fig. 9, is subjected to an extension. The plate in Fig. 9a has a height of $H = 2$, a width of $L = 2$. The radius of the internal hole is $R = 0.5$. The displacement is fixed at the bottom in y-direction. In the mid of the plate, at $x = \frac{L}{2}$ the displacement in x-direction was fixed. This example was considered in De Bellis et al. [12].

A prescribed displacement $\bar{u}_y = 1.5\,H$ is applied at the top of the plate. The Lamé parameters, that govern the hyperelastic response of the solid, are given as $\lambda = 71.0719$ and $\mu = 36.6128$. In Fig. 9b, c the structured and Voronoi meshes are shown, respectively, with respect to the initial configuration. In Fig. 9d, e the associated meshes in the reference configuration are depicted. As well a quadratic isoparametric map as a NURBS map were employed to map the structure from the reference to the initial configuration. Note that here a combination of 8 mappings was used to model the problem. The different mappings were tied together in a suitable way such that the displacement constraints related to the boundary value problem were enforced correctly.

In Fig. 10 the von Mises stress is depicted for different discretizations and mappings at the deformed configuration. It is clearly visible that the new scheme leads to results that cannot be distinguished from existing virtual element formulations, see Wriggers et al. [34], and a finite element solution using quadrilateral elements with quadratic shape functions.

In order to compare the different approaches a convergence study was performed for the displacement component u_x at the point $(x = L, y = \frac{H}{2})$ and the total force R_y, needed to obtain the resulting deformation state. Both results are depicted in Fig. 11 where a range from 30 to 9600 elements was used. Since there is no analytical solution available for this finite strain example an overkill solution based on the quadratic finite element with a mesh of around 0.5 Mio. element is used in this convergence study as reference solution

for the displacement component $u_{x,ref}$ and the total reaction $R_{y,ref}$.

As expected the solution with the quadratic finite element Q2 is superior. However it seems that the virtual element based on the NURBS mapping performs slightly better than the other formulations, especially for the Voronoi discretization. All virtual element formulations yield a better response than the Q1 finite element.

5.3 Bending of a C-profile

In the last example a C-profil is considered, see Fig. 12. On the right side the web of the profile is cut. The geometrical data are $H = 3$, $L = 1$ and $R = 0.5$. The profile is subjected at its flanges to a traction load $\bar{t}_x = 2$ in opposite directions, see Fig. 12a which varies from 0 to 2 in magnitude. The Lamé constitutive parameters of the hyperelastic strain energy function are $\lambda = 71.0719$ and $\mu = 36.6128$. The profile is fixed in the middle of the left web at $y = \frac{H}{2}$. As in the previous example, the total mesh of the boundary value problem is obtained by using 10 mappings from different reference configurations, shown in Fig. 12d, e. These generate the initial configuration Ω_0 in Fig. 12b, c for the isoparametric and NURBS map.

Due to the loading the C-profil bends and at the largest load level it assumes the deformed state that is depicted in Fig. 13. Again the results are plotted for different discretizations. The von Mises stresses are displayed in Fig. 13a for a finite element solution using Q2 finite element quadrilaterals and in (b) for a regular mesh with virtual elements that have 8 nodes. Furthermore, Fig. 13 includes results for the new mapping procedure. In (c) and (e) are the results for the isoparametric mapping for regular and Voronoi meshes, respectively, while (d) and (f) relate to the solutions obtained with the NURBS mapping. All discretizations produce basically the same answers.
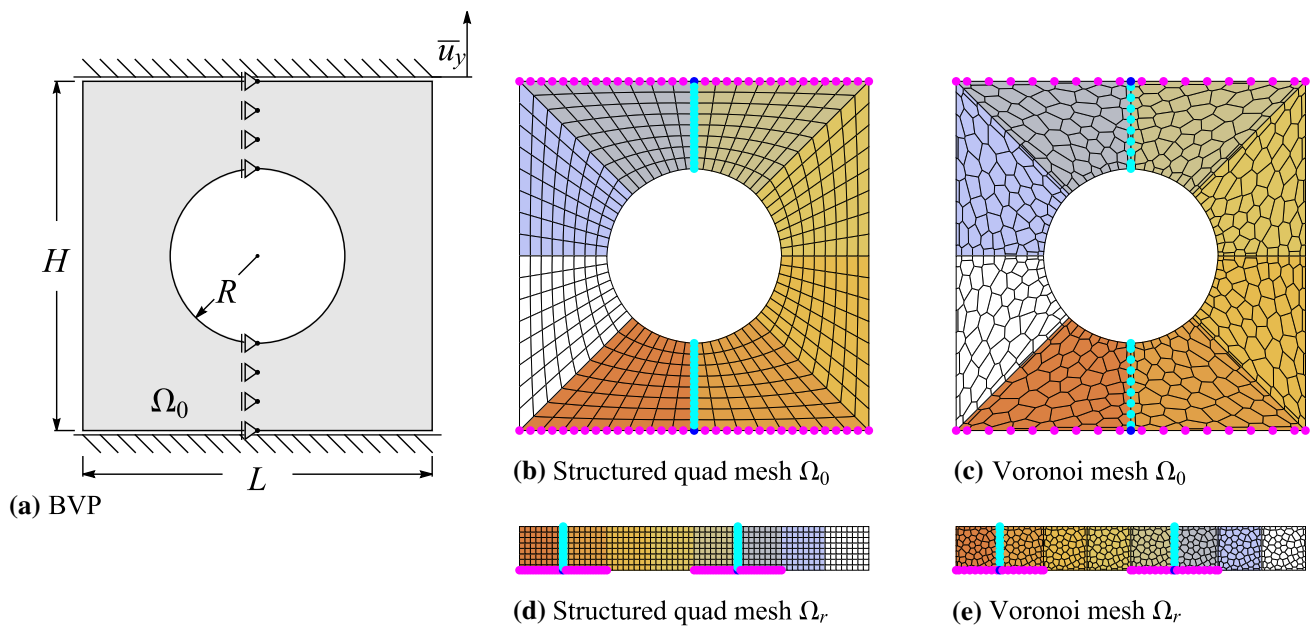
(a) BVP

(b) Structured quad mesh Ω_0

(c) Voronoi mesh Ω_0

(d) Structured quad mesh Ω_r

(e) Voronoi mesh Ω_r

Fig. 9 Plate with a hole, **a** set-up of the problem, **b–e** structured and Voronoi meshes

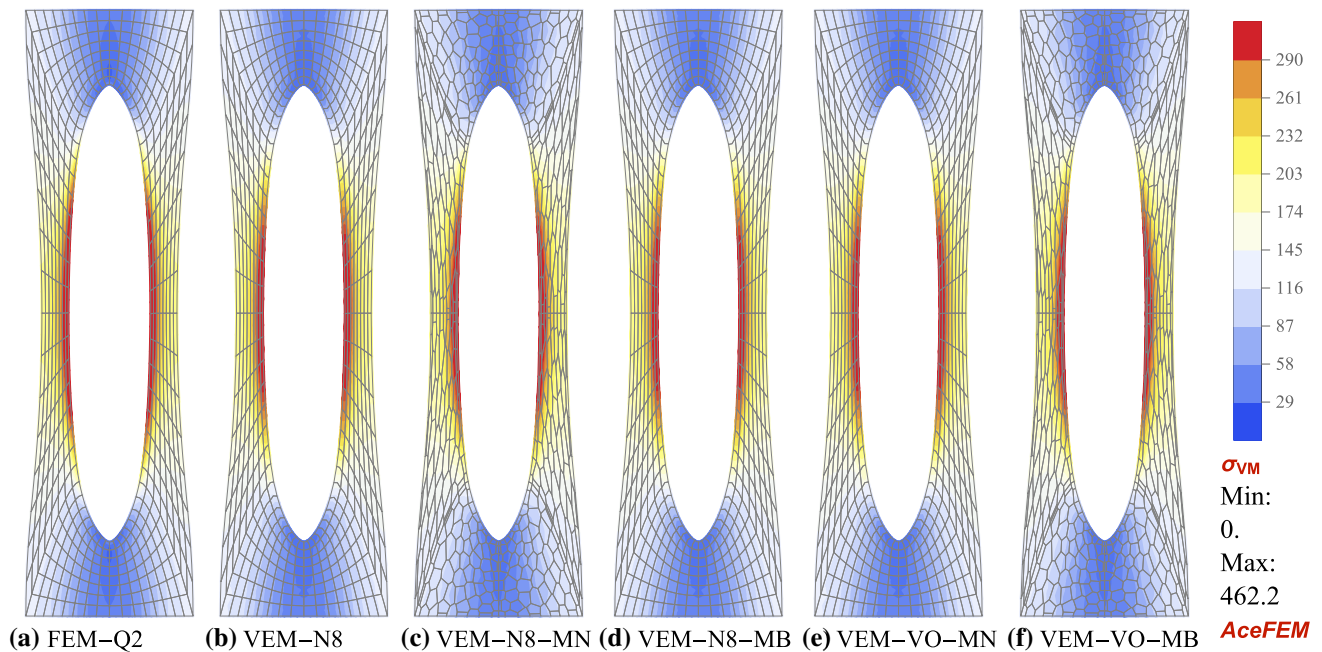(a) FEM−Q2 (b) VEM−N8 (c) VEM−N8−MN (d) VEM−N8−MB (e) VEM−VO−MN (f) VEM−VO−MB

Fig. 10 Von Mises stress in a plate with a hole for different formulations

Difference can be seen in the convergence study that is shown in Fig. 14. A solution based on quadratic finite elements with a very fine discretization of about 0.65 Mio. elements is used as a reference since analytical solutions are not available.

All discretization schemes produce the same load displacement curve, see the left part of Fig. 14. Differences occur in the convergence of the displacement u_y that is measured at the point $(x = L, y = H − L)$. All formulations converge.

Again the virtual element formulation using a Voronoi discretizations depicts good coarse mesh accuracy.

6 Summary and conclusions

In this contribution a virtual element method for finite strain elasticity was derived. The key novel feature is a general mapping which allows to deviate from the assumption of
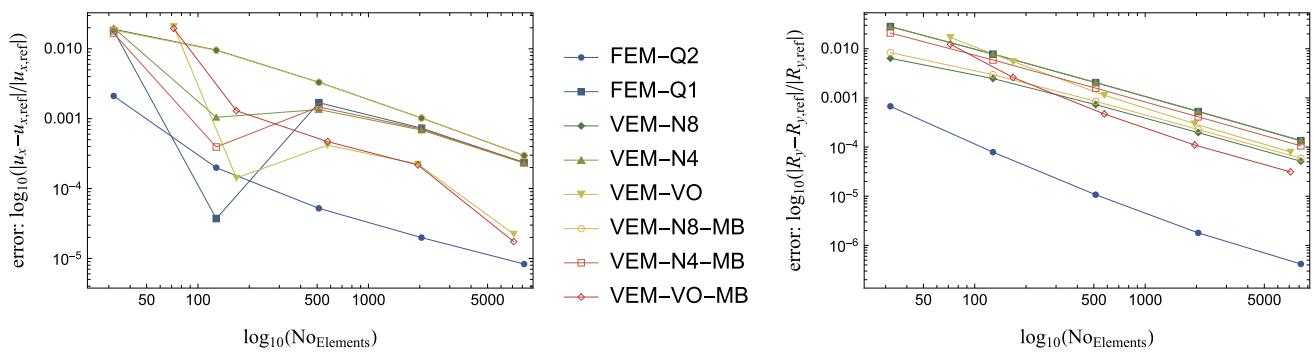
Fig. 11 Plate with a hole, convergence study
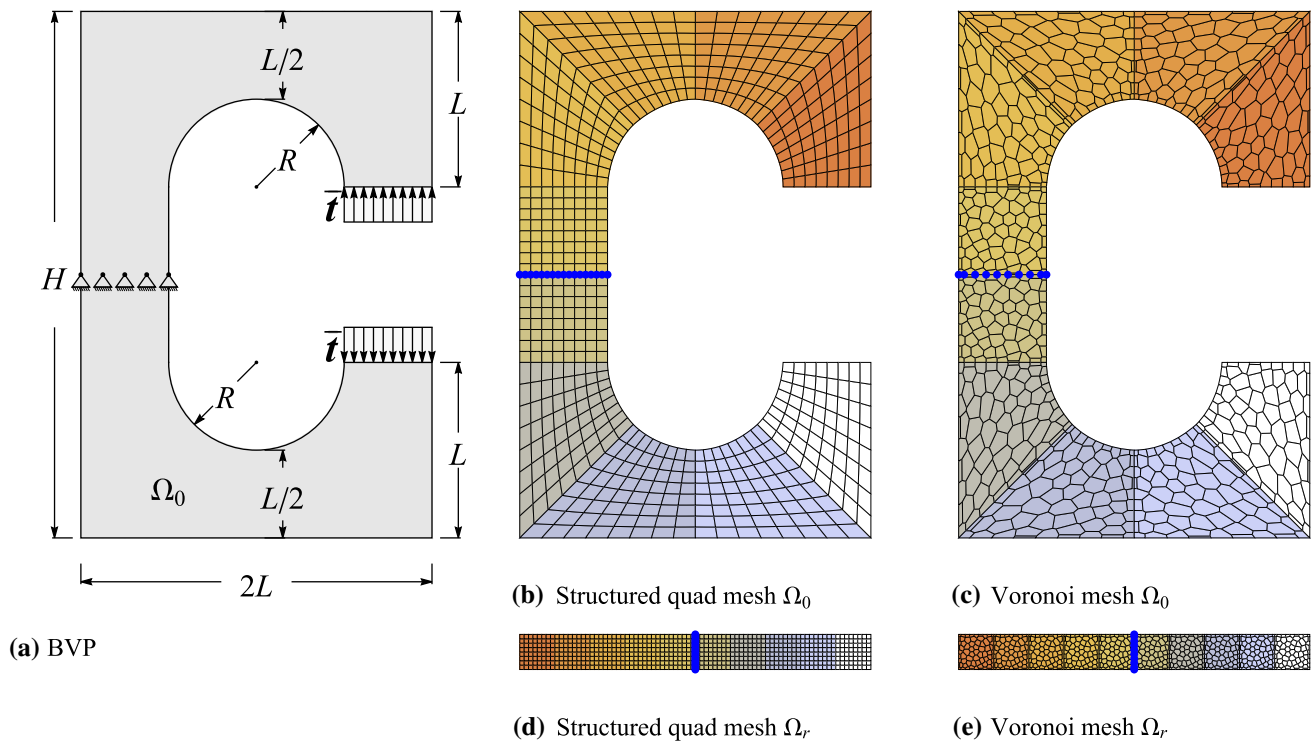
Fig. 12 C-profile, Set-up of the problem, structured and Voronoi meshes

straight edges of virtual elements. It was shown that general mappings like the isoparametric map and the NURBS approach, related to computer aided design, can be introduced to model exact geometries of complex structures with virtual elements. The formulation is simple and has been presented for both isoparametric and NURBS maps. It seems that the projected formulation is better in the case of Voronoi meshes when compared to solutions obtained with standard virtual elements using Voronoi meshes. For the special case of structured meshes the difference is very small. Here the virtual element solutions are basically reproduced in an exact manner.

The method proposed here is amenable to extensions of various kinds: for example to higher-order VEM formulations, like serendipity elements, to problems in three dimensions, and to other nonlinear problems such as those involving inelastic material behavior. First results indicate that the same mapping procedure can be employed for virtual elements with higher order ansatz functions.
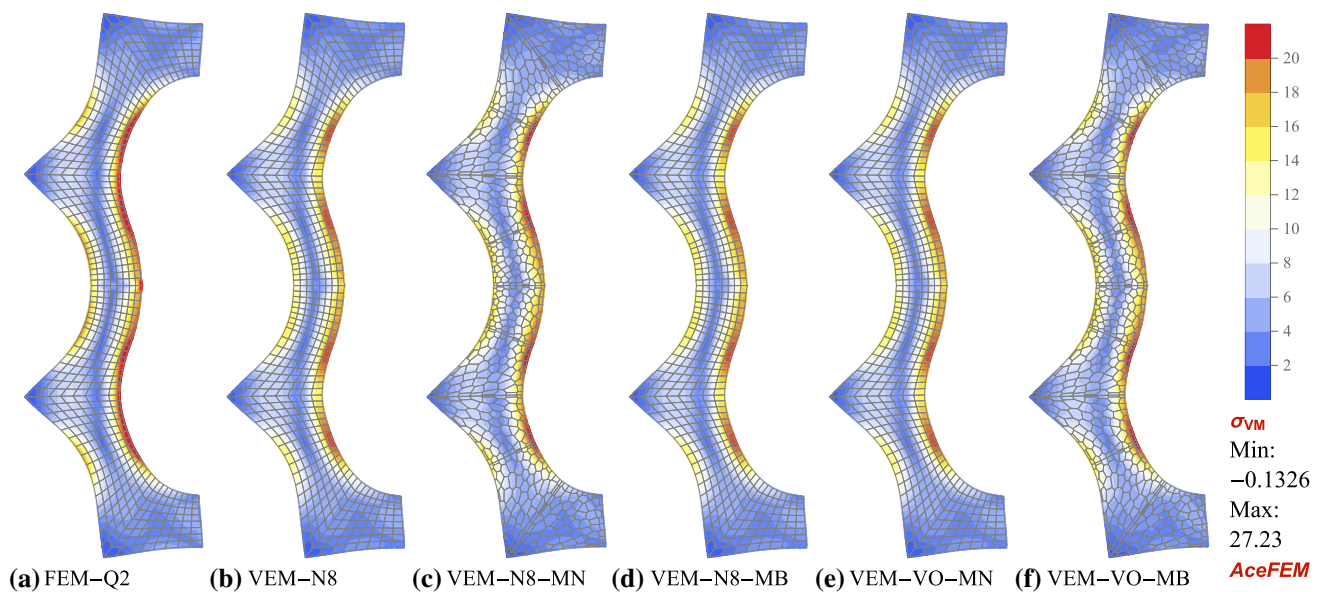
Fig. 13 Von Mises stress in the C-profile, different formulations
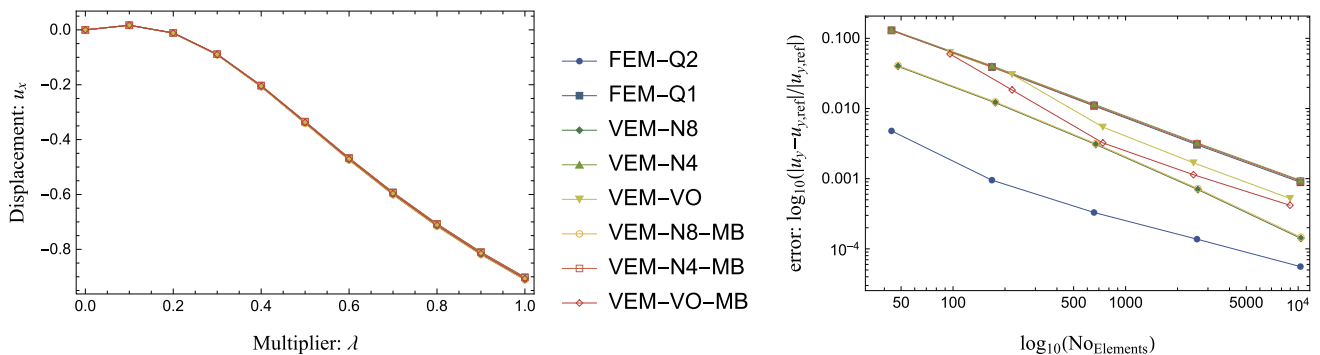
Fig. 14 C-profile, convergence study

References

1. Aldakheel F, Hudobivnik B, Wriggers P (2019) Virtual elements for finite thermo-plasticity problems. Comput Mech 64:1347–1360

2. Aldakheel F, Hudobivnik B, Artioli E, Beirão da Veiga L, Wriggers P (2020) Curvilinear virtual elements for contact mechanics. Comput Methods Appl Mech Eng **(submitted)**

3. Artioli E, Beirão da Veiga L, Lovadina C, Sacco E (2017) Arbitrary order 2D virtual elements for polygonal meshes: part II, inelastic problem. Comput Mech 60:643–657

4. Artioli E, Beirão da Veiga L, Dassi F (2020) Curvilinear virtual elements for 2D solid mechanics applications. Comput Methods Appl Mech Eng 359:112667

5. Beirão da Veiga L, Brezzi F, Marini L (2013) Virtual elements for linear elasticity problems. SIAM J Numer Anal 51:794–812

6. Beirão da Veiga L, Lovadina C, Mora D (2015) A virtual element method for elastic and inelastic problems on polytope meshes. Comput Methods Appl Mech Eng 295:327–346

7. Beirão da Veiga L, Russo A, Vacca G (2019) The virtual element method with curved edges. ESAIM Math Model Numer Anal 53(2):375–404

8. Belytschko T, Bindeman LP (1991) Assumed strain stabilization of the 4-node quadrilateral with 1-point quadrature for nonlinear problems. Comput Methods Appl Mech Eng 88(3):311–340

9. Chi H, Beirão da Veiga L, Paulino G (2017) Some basic formulations of the virtual element method (VEM) for finite deformations. Comput Methods Appl Mech Eng 318:148–192

10. Cottrell JA, Hughes TJR, Bazilevs Y (2009) Isogeometric analysis: toward integration of CAD and FEA. Wiley, New York

11. De Bellis M, Wriggers P, Hudobivnik B, Zavarise G (2018) Virtual element formulation for isotropic damage. Finite Elem Anal Des 144:38–48

12. De Bellis M, Wriggers P, Hudobivnik B (2019) Serendipity virtual element formulation for nonlinear elasticity. Comput Struct 223:106094

13. Farin G (1999) Curves and surfaces for computer aided geometric design: a practical guide, 5th edn. Academic Press, Boston

14. Flanagan D, Belytschko T (1981) A uniform strain hexahedron and quadrilateral with orthogonal hour-glass control. Int J Numer Methods Eng 17:679–706

15. Hughes TJR (1987) The finite element method. Prentice Hall, Englewood Cliffs

16. Hughes TJR, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. Comput Methods Appl Mech Eng 194(39–41):4135–4195

17. Hussein A, Aldakheel F, Hudobivnik B, Wriggers P, Guidault PA, Allix O (2019) A computational framework for brittle crack propagation based on an efficient virtual element method. Finite Elem Anal Des 159:15–32

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

19. Korelc J (2002) Multi-language and multi-environment generation of nonlinear finite element codes. Eng Comput 18:312–327

20. Korelc J, Wriggers P (2016) Automation of finite element methods. Springer, Berlin

21. Korelc J, Solinc U, Wriggers P (2010) An improved EAS brick element for finite deformation. Comput Mech 46:641–659

22. Krysl P (2015) Mean-strain eight-node hexahedron with stabilization by energy sampling. Int J Numer Methods Eng 103:437–449

23. Krysl P (2016) Mean-strain 8-node hexahedron with optimized energy-sampling stabilization. Finite Elem Anal Des 108:41–53

24. Mueller-Hoeppe DS, Loehnert S, Wriggers P (2009) A finite deformation brick element with inhomogeneous mode enhancement. Int J Numer Methods Eng 78:1164–1187

25. Onate E (2009) Structural analysis with the finite element method, vol 1: basis and solids. Springer, New York

26. Piegl L, Tiller W (1997) The NURBS book (monographs in visual communication), 2nd edn. Springer, New York

27. Pimenta PM, Campello EMB (2009) Shell curvature as an initial deformation: a geometrically exact finite element approach. Int J Numer Methods Eng 78(9):1094–1112

28. Reese S (2003) On a consistent hourglass stabilization technique to treat large inelastic deformations and thermo-mechanical coupling in plane strain problems. Int J Numer Methods Eng 57:1095–1127

29. Reese S, Wriggers P (2000) A new stabilization concept for finite elements in large deformation problems. Int J Numer Methods Eng 48:79–110

30. Reese S, Kuessner M, Reddy BD (1999) A new stabilization technique to avoid hourglassing in finite elasticity. Int J Numer Methods Eng 44:1617–1652

31. van Huyssteen D, Reddy BD (2020) A virtual element method for isotropic hyperelasticity. Comput Methods in Appl Mech Eng 367:113134

32. Wriggers P (2008) Nonlinear finite elements. Springer, Berlin

33. Wriggers P, Hudobivnik B (2017) A low order virtual element formulation for finite elasto-plastic deformations. Comput Methods Appl Mech Eng 327:459–477

34. Wriggers P, Reddy B, Rust W, Hudobivnik B (2017) Efficient virtual element formulations for compressible and incompressible finite deformations. Comput Mech 60:253–268

35. Zienkiewicz OC, Taylor RL (1989) The finite element method, vol 1, 4th edn. McGraw-Hill, London