



**POLITECNICO**  
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE  
E DELL'INFORMAZIONE

SCUOLA DI INGEGNERIA CIVILE,  
AMBIENTALE E TERRITORIALE

# Stabilized and self-stabilized virtual elements based on the Hu-Washizu variational principle for 3D linear elastostatics

DOUBLE DEGREE MASTER THESIS IN  
CIVIL ENGINEERING - MATHEMATICAL ENGINEERING

Author: **Elias Pesci**

Student ID: 962695

Advisor: Prof. Massimiliano Cremonesi

Co-advisor: Prof. Umberto Perego

Academic Year: 2022-23



# Abstract

This Master thesis presents the Virtual Element Method (VEM) for 3D linear elastostatics, developed from a mixed variational formulation based on the three-field Hu-Washizu functional.

A general mixed finite element scheme typically used to construct diverse solvers is firstly presented. Then, VEM are introduced within the mathematical formalism, and a numerical implementation following the general formulation discussed before is thoroughly explained and tested for the first two order of approximation  $k = 1, 2$ .

Subsequently, improvements of standard VEM are sought to address two major drawbacks: (1) the projection over the faces of each element of the virtual shape functions onto the space of polynomials and (2) the need of stabilization for the local stiffness matrix which exhibits a surplus of rank deficiency. Therefore, polyhedral elements with only triangular faces are introduced (*deltahedra*, hence  $\Delta$ VEM) and an enhanced formulation of  $k = 1$  VEM is proposed by carefully enhancing the strain field. It is shown how a local linear strain model is not sufficient to achieve self-stabilization while three non-complete quadratic polynomial strain fields are proposed and successfully tested for 8-nodes 24-DOFs self-stabilizing virtual elements.

**Keywords:** linear elasticity, Galerkin methods, mixed finite elements, polyhedral mesh, virtual element method, enhanced strain



# Abstract in lingua italiana

Questo lavoro di tesi di laurea Magistrale presenta il Metodo agli Elementi Virtuali (VEM) per l'elastostaticità lineare in tre dimensioni, sviluppato a partire da una formulazione variazionale mista basata sul funzionale a tre campi di Hu-Washizu.

In un primo momento viene presentato uno schema agli elementi finiti misti molto generale che può essere adottato per costruire diversi risolutori numerici. Successivamente, il VEM viene inquadrato nel contesto matematico e una sua implementazione numerica derivante dalla formulazione descritta in precedenza viene spiegata in dettaglio e sperimentata per i primi due ordini di approssimazione  $k = 1, 2$ .

Nella seconda parte del lavoro si cercano dei miglioramenti del VEM per sopperire a due principali svantaggi: (1) la proiezione sulle facce di ogni elemento delle funzioni di forma virtuali sullo spazio dei polinomi e (2) la necessità di stabilizzare la matrice di rigidità locale. Si introducono quindi poliedri le cui facce sono triangoli (*deltaedri*, da cui  $\Delta$ VEM) e una formulazione "enhanced" del VEM di ordine  $k = 1$  tramite un attento arricchimento del campo di deformazione. Viene mostrato come un modello lineare di deformazione locale non sia sufficiente a ottenere autostabilizzazione, mentre tre diversi campi di deformazione polinomiali quadratici, non completi, sono proposti e verificati con successo per un elemento virtuale autostabilizzato a 8 nodi e 24 gradi di libertà.

**Parole chiave:** elasticità lineare, metodi di Galerkin, elementi finiti misti, mesh poliedrica, metodo degli elementi virtuali, deformazione enhanced



# Contents

<b>Abstract</b>	<b>i</b>
<b>Abstract in lingua italiana</b>	<b>iii</b>
<b>Contents</b>	<b>v</b>
<b>Introduction</b>	<b>1</b>
0.1 Setting . . . . .	1
0.2 State of the art of the virtual element method . . . . .	2
0.3 Outline . . . . .	3
<b>1 Elastic problem in 3D</b>	<b>5</b>
1.1 Continuum mechanics for linear elasticity . . . . .	5
1.2 Mixed variational formulation of the continuous problem . . . . .	9
1.2.1 Stationarity of the continuous mixed functional . . . . .	10
1.3 Finite element approximation . . . . .	11
1.3.1 Discretization . . . . .	11
1.3.2 Stationarity of the discretized mixed functional . . . . .	17
1.3.3 Hourglass modes stabilization . . . . .	18
1.3.4 Assembly of the global system . . . . .	23
1.3.5 Enforcement of boundary conditions and solution . . . . .	24
1.3.6 Strains and stresses recovery . . . . .	25
1.3.7 Summary of the scheme . . . . .	25
<b>2 Tools of the virtual element method</b>	<b>27</b>
2.1 Mesh . . . . .	27
2.2 Local virtual element space in $\mathbb{R}^2$ . . . . .	32
2.3 Projection operator . . . . .	39
2.4 Enhanced local virtual element space in $\mathbb{R}^2$ . . . . .	47

2.5	Local virtual element space in $\mathbb{R}^3$ . . . . .	49
<b>3</b>	<b>Virtual elements for elastostatics in 3D</b>	<b>53</b>
3.1	Definition of the displacement and strain models . . . . .	53
3.2	Local stiffness matrix . . . . .	54
3.2.1	Consistent part of the local stiffness matrix . . . . .	54
3.2.2	Stabilizing part of the local stiffness matrix . . . . .	61
3.3	Equivalent nodal forces vector . . . . .	63
3.3.1	Approximation of body forces for first order VEM . . . . .	64
3.3.2	Approximation of body forces for higher order VEM . . . . .	64
<b>4</b>	<b>Numerical tests of the virtual element method for 3D elastostatics</b>	<b>67</b>
4.1	Data of the problem . . . . .	67
4.2	Meshes . . . . .	74
4.3	h-refinement convergence tests . . . . .	78
4.3.1	First order virtual elements . . . . .	79
4.3.2	Second order virtual elements . . . . .	85
<b>5</b>	<b>Deltahedra and enhanced strain VEM</b>	<b>87</b>
5.1	Deltahedral mesh first order VEM ( $\Delta$ VEM) . . . . .	87
5.1.1	Deltahedra . . . . .	88
5.1.2	Summary of the $\Delta$ VEM scheme . . . . .	92
5.2	Hu-Washizu-based enhanced strain VEM . . . . .	94
5.2.1	Enhanced strain field . . . . .	95
5.2.2	Stress and displacement field . . . . .	96
5.2.3	Discretized mixed functional . . . . .	96
5.2.4	Stationarity of the mixed functional . . . . .	98
5.2.5	Assembly, boundary conditions and solution . . . . .	99
5.2.6	Strains and stresses recovery . . . . .	99
5.3	Enhanced strain fields choices for the VEM . . . . .	100
5.3.1	Linear strain field . . . . .	101
5.3.2	31-strain-parameters 24-DOFs self-stabilized VE . . . . .	102
5.3.3	25-strain-parameters 24-DOFs self-stabilized VE . . . . .	105
5.3.4	18-strain-parameters 24-DOFs self-stabilized VE . . . . .	107
<b>6</b>	<b>Numerical tests of the <math>\Delta</math>VEM and SSVEM for 3D elastostatics</b>	<b>109</b>
6.1	h-convergence with brick-type meshes . . . . .	109
6.1.1	Trigonometric displacement field . . . . .	110

6.1.2	Cubic polynomial displacement field . . . . .	112
6.1.3	Quadratic polynomial displacement field . . . . .	113
6.1.4	Incompressible limit . . . . .	115
6.2	Cantilever beam . . . . .	118
6.3	Simply supported plate . . . . .	120
<b>7</b>	<b>Conclusions and future developments</b>	<b>123</b>
7.1	Summary of results . . . . .	123
7.2	Outlook and future work . . . . .	124
	<b>Bibliography</b>	<b>125</b>
	<b>A Gauss-Lobatto quadrature rule</b>	<b>131</b>
	<b>B Integration of monomials over polytopic domains</b>	<b>135</b>
	<b>List of Figures</b>	<b>137</b>
	<b>List of Tables</b>	<b>141</b>
	<b>List of Symbols</b>	<b>143</b>
	<b>Acknowledgements</b>	<b>149</b>



# Introduction

## 0.1. Setting

It is widely known that the study of partial differential equations very rarely yields to a solution in closed form, i.e. to an expression formed with constants, variables and a finite number of standard operations and functions. Even in those limited cases where analytical methods can be applied, the general integral depends on arbitrary functions and the imposition of the boundary conditions results in cumbersome computations which make the procedure impractical for engineering applications. Therefore, the analysis of partial differential equations is normally bound by establishing well-posedness of the problem, i.e. existence, uniqueness and continuous dependence from the data, and to study the regularity of the solution, while solving them is usually addressed by finding an approximation of the exact solution through numerical methods.

One of the most extensive techniques employed to solve boundary value and initial-boundary value differential problems is the *finite element method* (FEM, first appeared in [62]), whose general idea consists of partitioning the domain in a *finite* set of *elements*, called mesh, and approximating the unknown fields with piecewise polynomial functions. As a special case of the Galerkin method, the procedure translates a continuous operator problem to a discrete one, more precisely to a set of algebraic equations for steady-state differential problems and a set of ordinary differential equations for time-dependent problems. There is a wide and varied range of literature for the finite element method and detailed insights can be found, e.g., in the celebrated [22, 70].

The drawbacks linked to standard finite element schemes usually reside in the regularity requirements of the element shapes, which are the major issue for convergence properties of the approximate solution and occasionally represent a constraint for modelling smooth geometries in engineering applications.

Many variants of the finite element method undergoing the Galerkin schemes have been proposed by the numerical research community and successfully implemented in production codes. Among those, it is worth mentioning the *mixed finite element method*, where extra unknown fields are included and constrained by using Lagrangian multipliers in the

posing of the formulation. The resulting scheme is sometimes more robust with respect to irreducible, also called *primal* finite element methods, where only one unknown field is modeled, as it happens for nearly incompressible elasticity. The starting point of the present work will actually be a three-field mixed formulation. The *mimetic finite difference* (MFD, [26, 44]) method tries to preserve or *mimic* in the discrete problem the intrinsic mathematical properties of the continuous counterpart, as conservation laws, symmetries, maximum principle and asymptotic limits. The scheme lies between finite differences (FD) and finite elements since the discrete solution is represented by a set of degrees of freedom-like nodal values as in FD and not as a function, but the numerical scheme is built through a variational approach where grid functions are employed both for the trial and test functions as in FEM. The method allows to exploit very general polygonal/polyhedral meshes as in finite volume methods without the need of complex integration of shape functions. However, MFD schemes hardly cope with nonlinear partial differential equations since iterative methods, usually needed to deal with nonlinearities, require the evaluation of the numerical solution, which in turn is not available unless some proper reconstruction from the degrees of freedom is performed. The *weak Galerkin methods* relax the continuity requirement of approximating functions across the element boundaries, hence allowing the use of more general shapes constituting the mesh, the insertion of hanging nodes (i.e. nodes on the shared boundary between at least two neighboring elements but only belonging to one of the two elements) and the adoption of *hp*-adaptivity (the size  $h$  of the elements and the approximating polynomial order  $p$  vary in the mesh).

The present work focuses on a recently born technique named *virtual element method* (VEM, [9]), which allows to use general, possibly non-convex polygons/polyhedra and even elements with curved boundaries, support the embedding of hanging nodes and can easily be integrated in a standard FEM environment. In the following, a virtual element implementation for the elastic problem in three dimensions is proposed starting from a mixed variational formulation based on the three-field Hu-Washizu functional and a self-stabilizing scheme is presented from an enhancement of the strain field.

## 0.2. State of the art of the virtual element method

The *virtual element method* (VEM) was first presented in [9] by a group of Italian mathematicians as the ultimate evolution of the mimetic finite differences approach. The key idea, already implemented in generalized/extended finite element methods, relies on the addition of suitable *non-polynomial* functions to the usual finite element spaces. The novelty of the method, however, consists in performing a particular choice of the spaces and

the degrees of freedom so that computing the stiffness matrix does not require the computation of the non-polynomial shape functions, whose explicit expressions are actually never needed throughout the scheme. It is then possible to deal with complicated element geometries and higher-order continuity requirements, making the method applicable to a wide range of problems.

The first works on the topic focused on the Poisson equation, in particular on its numerical implementation ([10]) and on the possibility to achieve higher-order regularity ([24]). A linear elasticity VEM was developed in 2D in [25] and a corresponding low-order scheme in 3D in [36]. It was then extended to 2D inelastic problems, as in [27], where the possibility of dealing with hanging nodes is shown for a hard inclusion, modelled as a single element. Subsequently, contact problems were tackled, as in [66] and in [64] applied to Kirchhoff plates. A mixed formulation derived from the Hellinger-Reissner functional for the elastic problem can be found in [6]. Nonconforming VEM are developed in [34] and applied to the Stokes problem ([15]), Navier-Stokes equations ([45]), and bending of plates ([68]). Finite deformations are implemented in [17] and the incompressible version in [67]. A suitable application of the virtual element method, given its capability of dealing with very general mesh elements, is in fracture mechanics, as presented in [51]. A wide range of other problems in the VEM literature includes: a 3D mixed formulation for elastoplastic dynamics ([19]), the divergence-free Stokes problem ([29]), the Helmholtz problem ([55]), Morley-type plate bending ([69]), vibration for Kirchhoff plates ([48]), acoustic vibrations ([12]), geomechanics ([3]), Maxwell equations ([32]), magnetostatics ([28, 30]) and magnetohydrodynamics ([2]). Other interesting applications worth mentioning are the implementation of elements with curved edges ([31, 33]), the tackling of the obstacle problem ([63]) and topology optimization ([18]).

Given the intrinsic need of a *stabilization* procedure, many works are specifically oriented towards the study of its properties and solutions, as [11, 14]. Few results have been achieved in the seek of avoiding the need of stabilization. One notable solution for 2D linear elastic problems can be found in [43] and [16], where some self-stabilizing elements have been proposed.

Finally, a particularly instructive guide on the implementation in MATLAB [40] of the VEM for the Poisson equation is presented in [59].

### 0.3. Outline

The present Master thesis is structured in 7 chapters.

- **Chapter 1** recalls the concepts of continuum mechanics for three-dimensional linear elasticity and presents a mixed variational formulation, starting from the Hu-Washizu three field functional, which can be exploited to construct very general finite element schemes, amongst which the *virtual element method*.
- **Chapter 2** is entirely devoted to the virtual element method and explains some of its relevant mathematical formalism, such as the allowed meshes, the functional spaces it relies on and the projection operator.
- **Chapter 3** develops the virtual element method in the context of three-dimensional linear elastostatics, following the procedure presented in Chapter 1, and exploiting the tools explained in Chapter 2. The procedure for the construction of all the local matrices and vectors is given and motivated in detail.
- **Chapter 4** presents numerical tests of the virtual element method implemented in a MATLAB [40] program. These are conducted in the form of *h-refinement* for the first two order of accuracy of VEM, with various meshes (even non-convex) and compared with a trigonometric analytical solution. The deformed numerical solution of the displacement field is shown and stress contours are also rendered for a qualitative visual interpretation of the results.
- **Chapter 5** introduces some novelties in virtual element methods, trying to address two drawbacks: the projection operation required on all the faces of every polyhedral element and the need of stabilization. Encouraging solutions are found by adopting polyhedra with triangular faces (*deltahedra*) and properly enhancing the strain field to obtain *self-stabilizing* virtual elements. A numerical study on the spectrum of the local stiffness matrix generated by the such elements is executed as a first step towards validation.
- **Chapter 6** contains various numerical tests executed in MATLAB [40] for the new virtual elements in the form of *h*-convergence. Every analysis is additionally performed and compared with standard VEM, providing a wide range of results such as tests of capturing trigonometric and polynomial exact displacement fields and behaviour in the material nearly-incompressible limit.
- **Chapter 7** summarizes the main results, gives an outlook and proposes some future developments based on the present achievements.

# 1 | Elastic problem in 3D

## 1.1. Continuum mechanics for linear elasticity

This section briefly describes the continuum boundary value problem for solid mechanics and recalls its governing equations (see, e.g. [37]).

Infinitesimal strain theory applies, thus displacements are assumed much smaller than the relevant dimension of the deformable body and strains are much smaller than unity. The material is modelled through a linear elastic constitutive law.

Henceforth, the *Voigt notation* will be adopted ([39]), in view of the symmetric nature of the 2<sup>nd</sup> order stress tensor  $\boldsymbol{\sigma}$  and strain tensor  $\boldsymbol{\varepsilon}$ . Namely, the convention reduces the order of symmetric tensors and therefore allows to treat in 3D the aforementioned two quantities as  $[6 \times 1]$  vectors and the 4<sup>th</sup> order stiffness tensor coupling them as a  $[6 \times 6]$  matrix.

The solid body represented in Figure 1.1 by the domain  $\Omega \subset \mathbb{R}^3$  is set in a Cartesian reference system  $Oxyz$  with  $\boldsymbol{x} = \{x \ y \ z\}^T$  being a generic position vector, and its sufficiently regular boundary  $\partial\Omega$  is partitioned in a constrained part  $\partial_u\Omega$  and a free part  $\partial_p\Omega$ , such that  $\partial_u\Omega \cap \partial_p\Omega = \emptyset$ . On each point of the boundary  $\boldsymbol{x} \in \partial\Omega$  a local reference system  $\boldsymbol{x}e_u e_v n$  is set, where  $\boldsymbol{n}$  is the outward normal unit vector orthogonal to the surface and  $\boldsymbol{e}_u$  and  $\boldsymbol{e}_v$  are two mutually orthogonal arbitrarily chosen unit vectors tangent to the surface of  $\Omega$  so that the triple represents a right-hand oriented coordinate system. On the constrained subset of the boundary imposed displacements  $\bar{\boldsymbol{u}} = \bar{\boldsymbol{u}}(\boldsymbol{x})$  are assigned and on the free subset surface tractions  $\boldsymbol{p} = \boldsymbol{p}(\boldsymbol{x})$  are applied. The volume described by  $\Omega$  is subjected to body forces  $\boldsymbol{b} = \boldsymbol{b}(\boldsymbol{x})$ . The data of the problem is summarized below.

- body forces vector  $\boldsymbol{b}$  in  $\Omega$

$$\boldsymbol{b} = \begin{Bmatrix} b_x(x, y, z) \\ b_y(x, y, z) \\ b_z(x, y, z) \end{Bmatrix}$$

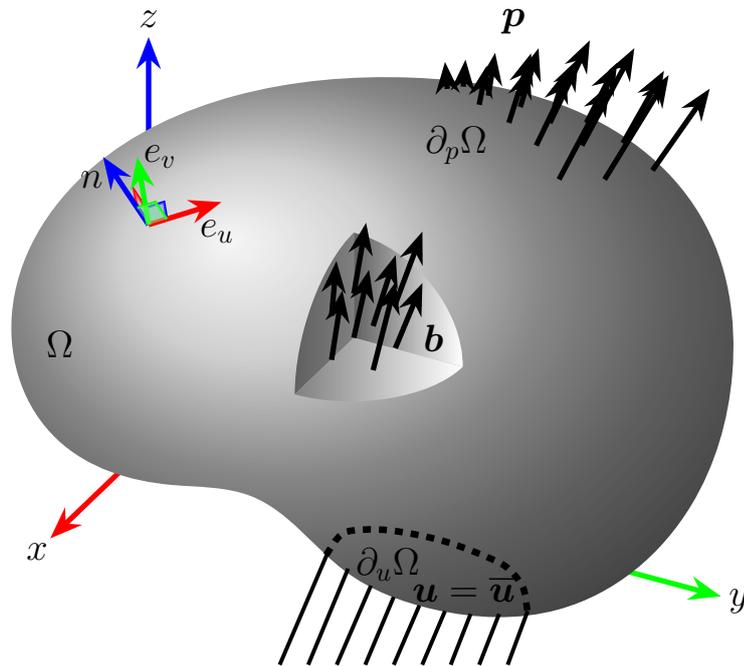


Figure 1.1: Elastic boundary value problem.

- surface tractions  $\mathbf{p}$  on  $\partial_p\Omega$

$$\mathbf{p} = \begin{Bmatrix} p_x(x, y, z) \\ p_y(x, y, z) \\ p_z(x, y, z) \end{Bmatrix}$$

- imposed displacements  $\bar{\mathbf{u}}$  on  $\partial_u\Omega$

$$\bar{\mathbf{u}} = \begin{Bmatrix} \bar{u}_x(x, y, z) \\ \bar{u}_y(x, y, z) \\ \bar{u}_z(x, y, z) \end{Bmatrix}$$

The unknowns for the problem are:

- displacement vector  $\mathbf{u}$  in  $\Omega$

$$\mathbf{u} = \begin{Bmatrix} u_x(x, y, z) \\ u_y(x, y, z) \\ u_z(x, y, z) \end{Bmatrix}$$

- strains vector  $\boldsymbol{\varepsilon}$  in  $\Omega$

$$\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_x(x, y, z) \\ \varepsilon_y(x, y, z) \\ \varepsilon_z(x, y, z) \\ \gamma_{xy}(x, y, z) \\ \gamma_{yz}(x, y, z) \\ \gamma_{xz}(x, y, z) \end{pmatrix}$$

- stresses vector  $\boldsymbol{\sigma}$  in  $\Omega$

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_x(x, y, z) \\ \sigma_y(x, y, z) \\ \sigma_z(x, y, z) \\ \tau_{xy}(x, y, z) \\ \tau_{yz}(x, y, z) \\ \tau_{xz}(x, y, z) \end{pmatrix}$$

The governing equations for the problem in matrix form are:

- indefinite equilibrium in  $\Omega$

$$\mathbf{S}^T \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0} \quad (1.1)$$

with boundary conditions on  $\partial_p \Omega$

$$\mathbb{N} \boldsymbol{\sigma} = \mathbf{p} \quad (1.2)$$

- kinematic compatibility in  $\Omega$

$$\boldsymbol{\varepsilon} = \mathbf{S} \mathbf{u} \quad (1.3)$$

with boundary conditions on  $\partial_u \Omega$

$$\mathbf{u} = \bar{\mathbf{u}} \quad (1.4)$$

- constitutive law for linear elasticity in  $\Omega$

$$\boldsymbol{\sigma} = \mathbf{D} \boldsymbol{\varepsilon} \quad (1.5)$$

In (1.3) the matrix  $\mathbf{S}$  is the compatibility differential operator<sup>1</sup>

$$\mathbf{S} = \begin{bmatrix} \partial_x & 0 & 0 \\ 0 & \partial_y & 0 \\ 0 & 0 & \partial_z \\ \partial_y & \partial_x & 0 \\ 0 & \partial_z & \partial_y \\ \partial_z & 0 & \partial_x \end{bmatrix} \quad (1.6)$$

and its transpose  $\mathbf{S}^T$  in (1.1) is the equilibrium differential operator<sup>2</sup>

$$\mathbf{S}^T = \begin{bmatrix} \partial_x & 0 & 0 & \partial_y & 0 & \partial_z \\ 0 & \partial_y & 0 & \partial_x & \partial_z & 0 \\ 0 & 0 & \partial_z & 0 & \partial_y & \partial_x \end{bmatrix} \quad (1.7)$$

where  $\partial_{(\cdot)}$  represents the partial derivative with respect to  $(\cdot)$ .

The matrix  $\mathbb{N}$  contains the direction cosines of the outward normal unit vector  $\mathbf{n}$  so that the matrix product in (1.2) correctly represents the tensor counterpart  $\sigma_{ij}n_j$ .

$$\mathbb{N} = \begin{bmatrix} n_x & 0 & 0 & n_y & 0 & n_z \\ 0 & n_y & 0 & n_x & n_z & 0 \\ 0 & 0 & n_z & 0 & n_y & n_x \end{bmatrix} \quad (1.8)$$

where  $n_x$ ,  $n_y$  and  $n_z$  are the three components of the outward normal unit vector  $\mathbf{n} = \{n_x \ n_y \ n_z\}^T$ . In (1.5), for homogeneous isotropic media, the  $[6 \times 6]$  material elastic stiffness matrix  $\mathbf{D}$  represents in Voigt notation the tensor identity  $\sigma_{ij} = 2\mu\varepsilon_{ij} + \lambda\delta_{ij}\varepsilon_{kk}$ , where  $\delta_{ij}$  is the Kronecker delta, and reduces to

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \quad (1.9)$$

where  $\lambda$  is the *first Lamé parameter* and  $\mu$  the *second Lamé parameter*, equal to the

<sup>1</sup>The compatibility differential operator  $\mathbf{S} : \mathbb{R}^3 \rightarrow \mathbb{R}^6$  is the equivalent in Voigt notation of the symmetric gradient operator  $\nabla_{\text{sym}}^{3 \times 3} : \mathbb{R}^3 \rightarrow \mathbb{R}_{\text{sym}}^{3 \times 3}$  which produces a tensor.

<sup>2</sup>The equilibrium differential operator  $\mathbf{S}^T : \mathbb{R}^6 \rightarrow \mathbb{R}^3$  is the equivalent in Voigt notation of the tensor divergence operator  $\mathbf{div} : \mathbb{R}_{\text{sym}}^{3 \times 3} \rightarrow \mathbb{R}^3$ .

shear modulus  $G$ . Equivalently, the above material stiffness matrix can be expressed as a function of the Young's modulus  $E$  and the Poisson's ratio  $\nu$ , broadly used in engineering.

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \quad (1.10)$$

where the following identities have been applied

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu} \quad \nu = \frac{\lambda}{2(\lambda + \mu)}$$

## 1.2. Mixed variational formulation of the continuous problem

In this section a mixed three-field variational formulation is presented based on the *Hu-Washizu functional* ([65]). The formulation is the most general one since it assumes the three unknown fields  $\mathbf{u}$ ,  $\boldsymbol{\varepsilon}$  and  $\boldsymbol{\sigma}$  of the elastic problem to be independent. Under suitable assumptions, the formulation is equivalent to other reduced formulations, such as the *Hellinger-Reissner* two-field formulation, which takes only  $\mathbf{u}$  and  $\boldsymbol{\sigma}$  as independent fields, as shown in [35]. Eventually, under proper requirements based on energy conservation at the discrete level, the Hu-Washizu formulation can be brought back to the irreducible primal one-field displacement-based formulation, corresponding to the widely known *total potential energy* functional.

**Definition 1.1** (Hu-Washizu functional). *The Hu-Washizu functional  $\Pi$  is defined as*

$$\Pi : U \times \mathcal{E} \times \Sigma \rightarrow \mathbb{R}$$

$$\Pi(\mathbf{u}, \boldsymbol{\varepsilon}, \boldsymbol{\sigma}) = \frac{1}{2} \int_{\Omega} \boldsymbol{\varepsilon}^T \mathbf{D} \boldsymbol{\varepsilon} \, d\Omega - \int_{\Omega} \boldsymbol{\sigma}^T (\boldsymbol{\varepsilon} - \mathbf{S} \mathbf{u}) \, d\Omega - \int_{\Omega} \mathbf{u}^T \mathbf{b} \, d\Omega - \int_{\partial_p \Omega} \mathbf{u}^T \mathbf{p} \, d\Sigma \quad (1.11)$$

where  $U$ ,  $\mathcal{E}$  and  $\Sigma$  are suitable functional spaces defined in the domain  $\Omega$ .

The last integral in (1.11) is computed only on the free part  $\partial_p \Omega$  of the boundary surface since on the constrained part  $\partial_u \Omega$  it is assumed a priori that  $\mathbf{u} = \bar{\mathbf{u}}$ .

### 1.2.1. Stationarity of the continuous mixed functional

The following result allows to find the relationships between the three unknown fields and will be used to build a general finite element scheme, exploitable for a virtual element setting.

**Theorem 1.1** (Hu-Washizu principle). *The true solution of the elastic problem amongst all those admissible is the one that makes the Hu-Washizu functional stationary.*

*Proof.* Performing a variation of the functional and setting it to 0 yields to

$$\begin{aligned} \delta\Pi = & \int_{\Omega} \delta\boldsymbol{\varepsilon}^T \mathbf{D}\boldsymbol{\varepsilon} \, d\Omega - \int_{\Omega} \boldsymbol{\sigma}^T (\delta\boldsymbol{\varepsilon} - \mathbf{S}\delta\mathbf{u}) \, d\Omega - \int_{\Omega} \delta\boldsymbol{\sigma}^T (\boldsymbol{\varepsilon} - \mathbf{S}\mathbf{u}) \, d\Omega + \\ & - \int_{\Omega} \delta\mathbf{u}^T \mathbf{b} \, d\Omega - \int_{\partial_p\Omega} \delta\mathbf{u}^T \mathbf{p} \, d\Sigma = 0 \quad \forall \delta\mathbf{u}, \delta\boldsymbol{\varepsilon}, \delta\boldsymbol{\sigma} \end{aligned} \quad (1.12)$$

Integrating by parts the integral involving  $\boldsymbol{\sigma}^T \mathbf{S}\delta\mathbf{u}$  and recalling that the variation of the displacement field  $\delta\mathbf{u}$  on  $\partial_u\Omega$  is null one has

$$\begin{aligned} \int_{\Omega} \boldsymbol{\sigma}^T \mathbf{S}\delta\mathbf{u} \, d\Omega &= \int_{\partial\Omega=\partial_p\Omega\cup\partial_u\Omega} \delta\mathbf{u}^T \mathbb{N}\boldsymbol{\sigma} \, d\Sigma - \int_{\Omega} \delta\mathbf{u}^T \mathbf{S}^T \boldsymbol{\sigma} \, d\Omega = \\ &= \int_{\partial_p\Omega} \delta\mathbf{u}^T \mathbb{N}\boldsymbol{\sigma} \, d\Sigma - \int_{\Omega} \delta\mathbf{u}^T \mathbf{S}^T \boldsymbol{\sigma} \, d\Omega \end{aligned} \quad (1.13)$$

so that

$$\begin{aligned} \delta\Pi = & \int_{\Omega} \delta\boldsymbol{\varepsilon}^T \mathbf{D}\boldsymbol{\varepsilon} \, d\Omega - \int_{\Omega} \boldsymbol{\sigma}^T \delta\boldsymbol{\varepsilon} \, d\Omega + \int_{\partial_p\Omega} \delta\mathbf{u}^T \mathbb{N}\boldsymbol{\sigma} \, d\Sigma - \int_{\Omega} \delta\mathbf{u}^T \mathbf{S}^T \boldsymbol{\sigma} \, d\Omega + \\ & - \int_{\Omega} \delta\boldsymbol{\sigma}^T (\boldsymbol{\varepsilon} - \mathbf{S}\mathbf{u}) \, d\Omega - \int_{\Omega} \delta\mathbf{u}^T \mathbf{b} \, d\Omega - \int_{\partial_p\Omega} \delta\mathbf{u}^T \mathbf{p} \, d\Sigma = 0 \quad \forall \delta\mathbf{u}, \delta\boldsymbol{\varepsilon}, \delta\boldsymbol{\sigma} \end{aligned} \quad (1.14)$$

Finally, gathering the terms involving common variations leads to

$$\begin{aligned} \delta\Pi = & - \int_{\Omega} \delta\mathbf{u}^T (\mathbf{S}^T \boldsymbol{\sigma} + \mathbf{b}) \, d\Omega + \int_{\Omega} \delta\boldsymbol{\varepsilon}^T (\mathbf{D}\boldsymbol{\varepsilon} - \boldsymbol{\sigma}) \, d\Omega + \\ & - \int_{\Omega} \delta\boldsymbol{\sigma}^T (\boldsymbol{\varepsilon} - \mathbf{S}\mathbf{u}) \, d\Omega + \int_{\partial_p\Omega} \delta\mathbf{u}^T (\mathbb{N}\boldsymbol{\sigma} - \mathbf{p}) \, d\Sigma = 0 \quad \forall \delta\mathbf{u}, \delta\boldsymbol{\varepsilon}, \delta\boldsymbol{\sigma} \end{aligned} \quad (1.15)$$

Equation (1.15) is the weak form of the elastic problem described in 1.1. By the *fundamental lemma of the calculus of variations*, the variation of the functional  $\delta\Pi$  being null for *any* admissible variation of the three fields implies that the four quantities in round brackets above are null in a suitable sense in the domains described by their respective integrals, giving rise to four *Euler's equations* corresponding to (1.1), (1.5), (1.3) and

(1.2). Hence, the set  $(\mathbf{u}, \boldsymbol{\varepsilon}, \boldsymbol{\sigma})$  satisfying the governing equations for the linear elastic problem makes the Hu-Washizu functional stationary.  $\square$

## 1.3. Finite element approximation

### 1.3.1. Discretization

The preprocessing phase, or *meshing*, in finite element methods consists of generating a partition of the domain  $\mathcal{P}_h$  where the governing equations of the problem are being studied. The volume of the body then becomes

$$|\Omega| \approx |\mathcal{P}_h| = \sum_{e=1}^{n_e} |\Omega_e|$$

where  $n_e$  is the number of elements composing the mesh and  $e$  is the general index representing the element  $e$ .

By linearity of the integral operator, the definite integrals in the domain  $\Omega$  become

$$\int_{\Omega} (\cdot) d\Omega \approx \sum_{e=1}^{n_e} \int_{\Omega_e} (\cdot) d\Omega \quad (1.16)$$

The same discretization procedure can be applied to the Hu-Washizu functional, so that

$$\Pi \approx \sum_{e=1}^{n_e} \Pi_e \quad (1.17)$$

where

$$\begin{aligned} \Pi_e : U_e \times \mathcal{E}_e \times \Sigma_e &\rightarrow \mathbb{R} \\ \Pi_e(\mathbf{u}, \boldsymbol{\varepsilon}, \boldsymbol{\sigma}) &= \frac{1}{2} \int_{\Omega_e} \boldsymbol{\varepsilon}^T(\boldsymbol{\xi}) \mathbf{D} \boldsymbol{\varepsilon}(\boldsymbol{\xi}) d\Omega - \int_{\Omega_e} \boldsymbol{\sigma}^T(\boldsymbol{\xi}) (\boldsymbol{\varepsilon}(\boldsymbol{\xi}) - \mathbf{S} \mathbf{u}(\boldsymbol{\xi})) d\Omega + \\ &\quad - \int_{\Omega_e} \mathbf{u}^T(\boldsymbol{\xi}) \mathbf{b}(\boldsymbol{\xi}) d\Omega - \int_{\partial_p \Omega_e} \mathbf{u}^T(\boldsymbol{\xi}) \mathbf{p}(\boldsymbol{\xi}) d\Sigma \end{aligned} \quad (1.18)$$

in which  $U_e$ ,  $\mathcal{E}_e$  and  $\Sigma_e$  are suitable functional spaces defined in the domain  $\Omega_e$  and  $\boldsymbol{\xi}$  is the vector of non-dimensional barycentric local coordinates of the element (Figure 1.2b), given by

$$\boldsymbol{\xi} = \begin{Bmatrix} \xi \\ \eta \\ \zeta \end{Bmatrix} = \frac{\mathbf{x} - \mathbf{x}_G}{h_e} = \begin{Bmatrix} \frac{x-x_G}{h_e} \\ \frac{y-y_G}{h_e} \\ \frac{z-z_G}{h_e} \end{Bmatrix} \quad (1.19)$$

with  $\mathbf{x}_G = \{x_G \ y_G \ z_G\}^T$  being the centroid of the element and  $h_e$  the *diameter* of the element, or *element size*, namely the maximum distance between two points belonging to the element boundary (Figure 1.2a)

$$h_e = \max_{\mathbf{x}_1, \mathbf{x}_2 \in \partial\Omega_e} \|\mathbf{x}_1 - \mathbf{x}_2\| \quad (1.20)$$

The element size plays a fundamental role in convergence analysis of finite element methods.

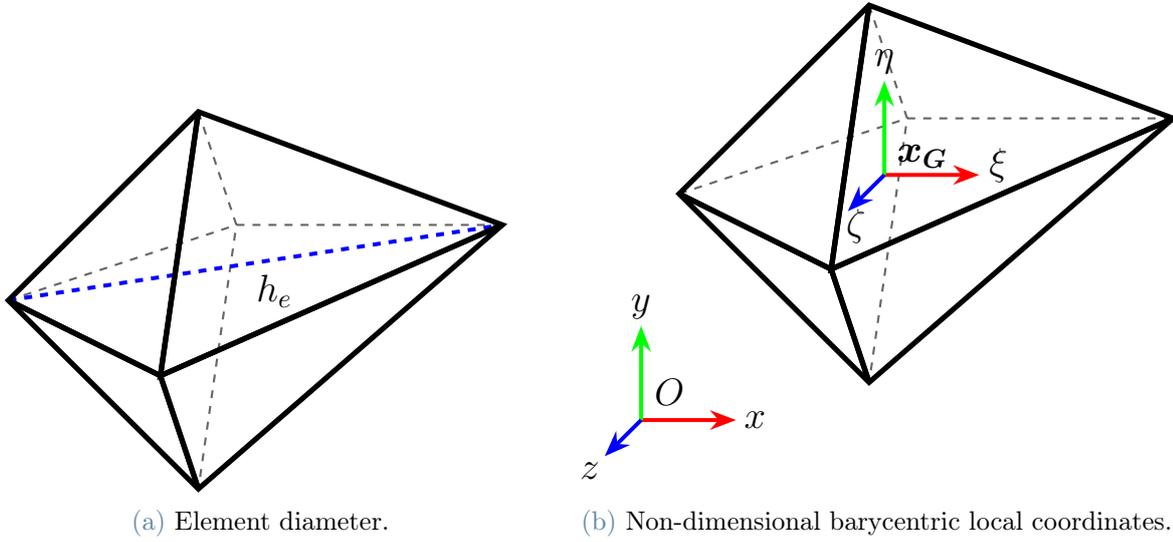


Figure 1.2: Example of element diameter and barycentric local coordinates for a polyhedron.

The discretization (1.17) of the functional holds in view of (1.16) for the integrals of (1.18) over  $\Omega$  and by global continuity of the displacement field  $\mathbf{u}$  and equal modulus and opposite sign of the elements surface tractions  $\mathbf{p}$  across the elements boundaries. More precisely,

$$\sum_{e=1}^{n_e} \int_{\partial_p \Omega_e \setminus \partial_p \Omega} \mathbf{u}^T(\boldsymbol{\xi}) \mathbf{p}(\boldsymbol{\xi}) \, d\Sigma = 0$$

An approximation of the three independent fields at elemental level  $\mathbf{u}$ ,  $\boldsymbol{\varepsilon}$  and  $\boldsymbol{\sigma}$  is introduced ([7, 13, 70])

$$\mathbf{u}(\boldsymbol{\xi}) \approx \mathbf{u}^h(\boldsymbol{\xi}) = \mathbf{N}_u(\boldsymbol{\xi}) \hat{\mathbf{u}} \quad (1.21)$$

$$\boldsymbol{\varepsilon}(\boldsymbol{\xi}) \approx \boldsymbol{\varepsilon}^h(\boldsymbol{\xi}) = \mathbf{N}_\varepsilon(\boldsymbol{\xi}) \hat{\boldsymbol{\varepsilon}} \quad (1.22)$$

$$\boldsymbol{\sigma}(\boldsymbol{\xi}) \approx \boldsymbol{\sigma}^h(\boldsymbol{\xi}) = \mathbf{N}_\sigma(\boldsymbol{\xi}) \hat{\boldsymbol{\sigma}} \quad (1.23)$$

where the superscript  $h$  indicates that the field is approximated<sup>3</sup> and matrices  $\mathbf{N}_{(\cdot)}$  map a discrete quantity  $(\hat{\cdot})$  into a continuous function and are known in the finite element literature as *shape functions*. More specifically, in 3D the three ansätze  $\mathbf{N}_u$ ,  $\mathbf{N}_\varepsilon$ ,  $\mathbf{N}_\sigma$  have respectively dimensions  $[3 \times n_u]$ ,  $[6 \times n_\varepsilon]$  and  $[6 \times n_\sigma]$ , where  $n_{(\cdot)}$  is the number of parameters required to describe the discrete field  $(\hat{\cdot})$ . In standard primal FE,  $\hat{\mathbf{u}}$  are nodal displacement values, while in a mixed formulation where also strains or stresses are unknowns,  $\hat{\boldsymbol{\varepsilon}}$  and  $\hat{\boldsymbol{\sigma}}$  are not necessarily nodal values and might lose physical meaning. All the maps  $\mathbf{N}_u$ ,  $\mathbf{N}_\varepsilon$  and  $\mathbf{N}_\sigma$  are *locally* continuous, i.e. continuous in the element interior  $\Omega_e$ , while only the first one is also *globally* continuous so that the approximate displacement field  $\mathbf{u}^h \in [C^0(\Omega)]^3$ . The last requirement is relaxed for discontinuous Galerkin methods ([20]).

The assumed strain field must satisfy a particular case of *integrability conditions* undergoing the name of *internal compatibility*, namely the body must exhibit strains for which a continuous, single-valued displacement field is guaranteed ([57]). More specifically, solving the differential strain-displacement relations in the vector unknown  $\mathbf{u}$ , given a strain field  $\boldsymbol{\varepsilon}$ , is an overdetermined problem as it involves six independent equations and only three unknowns. Hence, additional equations for the strain field must be provided when trying to reconstruct the displacement field, which is the goal of finite element schemes. The mechanical reason behind this requirement lies in the fact that no overlapping nor tears of the constitutive material are allowed. The following theorem provides the above-mentioned differential equations that a generic strain field must satisfy in order to guarantee the existence of the corresponding continuous, single-valued displacement field. We shall briefly abandon Voigt notation when dealing with this theorem, where tensor calculus comes into help, so that  $\boldsymbol{\varepsilon}$  is a 2<sup>nd</sup> order tensor. Moreover, to indicate partial derivatives  $\frac{\partial}{\partial(\cdot)}(\diamond)$  the following notation will be adopted  $(\diamond)_{,(\cdot)}$

**Theorem 1.2 (Compatibility conditions).** *If the displacement field  $\mathbf{u}$  is continuous and single-valued, then the following identity holds for the strain field  $\boldsymbol{\varepsilon}$*

$$\nabla \times (\nabla \times \boldsymbol{\varepsilon}) = \mathbf{0} \quad (1.24)$$

or, using Einstein summation convention,

$$e_{ikr} e_{jls} \varepsilon_{kl,ij} = 0 \quad (1.25)$$

where  $e_{ijk}$  is the Levi-Civita symbol.

---

<sup>3</sup>Henceforth, the superscript  $h$  will be omitted for the sake of conciseness when referring to the local approximated fields.

If the body is simply connected, the above conditions are also sufficient for the existence of a continuous single-valued displacement field.

In (1.25) the *Levi-Civita*, or *permutation* symbol<sup>4</sup> is defined as

$$e_{ijk} = \begin{cases} 1 & \text{if } (i, j, k) \text{ is an even permutation of } (1, 2, 3) \\ -1 & \text{if } (i, j, k) \text{ is an odd permutation of } (1, 2, 3) \\ 0 & \text{if any index is repeated} \end{cases} \quad (1.26)$$

*Proof.* The proof for the necessary condition is reported here, while for the sufficient condition a detailed explanation can be found in [57], together with an exhaustive chapter on tensor calculus.

In the hypothesis of infinitesimal strains, the gradient of the displacements  $\nabla \mathbf{u}$  can be decomposed in a symmetric part corresponding to the strains  $\boldsymbol{\varepsilon}$  and a skew-symmetric part  $\boldsymbol{\omega}$

$$\begin{aligned} \nabla \mathbf{u} &= \boldsymbol{\varepsilon} + \boldsymbol{\omega} \\ \boldsymbol{\varepsilon} &= \frac{1}{2} (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) & \boldsymbol{\omega} &= \frac{1}{2} (\nabla \mathbf{u} - (\nabla \mathbf{u})^T) \\ \varepsilon_{ij} &= \frac{1}{2} (u_{i,j} + u_{j,i}) & \omega_{ij} &= \frac{1}{2} (u_{i,j} - u_{j,i}) \end{aligned} \quad (1.27)$$

Taking the gradient of  $\boldsymbol{\omega}$

$$\begin{aligned} \omega_{ij,k} &= \frac{1}{2} (u_{i,j} - u_{j,i})_k = \frac{1}{2} (u_{i,jk} + u_{k,ij} - u_{k,ij} - u_{j,ik}) = \\ &= (u_{i,kj} + u_{k,ij} - u_{k,ji} - u_{j,ki}) = \varepsilon_{ik,j} - \varepsilon_{jk,i} \end{aligned} \quad (1.28)$$

If  $\boldsymbol{\omega}$  is continuously differentiable, differentiating another time and exploiting Schwarz theorem yields to

$$\omega_{ij,kl} = \omega_{ij,lk} \quad (1.29)$$

hence

$$\varepsilon_{ik,jl} - \varepsilon_{jk,il} - \varepsilon_{il,jk} + \varepsilon_{jl,ik} = 0 \quad (1.30)$$

which correspond to  $3^4 = 81$  equations by letting the indices  $i, j, k, l$  vary within the values 1, 2, 3. These, however are not independent since the tensor  $\boldsymbol{\varepsilon}$  is symmetric and reduce to the 9 equations in the free indices  $r$  and  $s$  of (1.25). The latter are, in turn, again symmetric in  $r$  and  $s$  and eventually give rise to 6 independent internal compatibility

---

<sup>4</sup>The Levi-Civita symbol is usually indicated by  $\varepsilon_{ijk}$  and here is represented by  $e_{ijk}$ , not to be confused with the strain symbol.

equations, now written in Voigt notation,

$$\left\{ \begin{array}{l} \varepsilon_{x,yy} + \varepsilon_{y,xx} - \gamma_{xy} = 0 \\ \varepsilon_{y,zz} + \varepsilon_{z,yy} - \gamma_{yz} = 0 \\ \varepsilon_{z,xx} + \varepsilon_{x,zz} - \gamma_{xz} = 0 \\ (\gamma_{xy,z} - \gamma_{yz,x} + \gamma_{xz,y})_{,x} - 2\varepsilon_{x,yz} = 0 \\ (\gamma_{yz,x} - \gamma_{xz,y} + \gamma_{xy,z})_{,y} - 2\varepsilon_{y,xz} = 0 \\ (\gamma_{xz,y} - \gamma_{xy,z} + \gamma_{yz,x})_{,z} - 2\varepsilon_{z,xy} = 0 \end{array} \right. \quad (1.31)$$

Finally, the equivalence of (1.24) with the identity (1.25) is obtained applying twice the definition of curl of a 2<sup>nd</sup> order tensor

$$(\nabla \times \boldsymbol{\varepsilon})_{sk} = e_{jls} \varepsilon_{kl,j}$$

so that

$$\left( \nabla \times (\nabla \times \boldsymbol{\varepsilon}) \right)_{sk} = e_{ikr} (\nabla \times \boldsymbol{\varepsilon})_{sk,i} = e_{ikr} e_{jls} \varepsilon_{kl,ji} = e_{ikr} e_{jls} \varepsilon_{kl,ij}$$

□

It is important to remark that the governing equations of the elastic problem 1.1, 1.3 and 1.5 already close the problem setting in a displacement-based procedure without forcing internal compatibility conditions, which are actually automatically satisfied. However, following a mixed variational scheme, this is not necessarily the case and internal compatibility (1.24) is required to correctly choose an admissible strain field.

A possibility to preserve the elements energy is to choose the approximated strain and stress fields such that the energy given by their scalar product is conserved. To perform the scalar product we must have  $n_\varepsilon = n_\sigma$  and hence

$$\hat{\boldsymbol{\sigma}}^T \hat{\boldsymbol{\varepsilon}} = \int_{\Omega_\varepsilon} \boldsymbol{\sigma}^T \boldsymbol{\varepsilon} \, d\Omega = \hat{\boldsymbol{\sigma}}^T \left( \int_{\Omega_\varepsilon} \mathbf{N}_\sigma^T \mathbf{N}_\varepsilon \, d\Omega \right) \hat{\boldsymbol{\varepsilon}} \quad (1.32)$$

so that

$$\int_{\Omega_\varepsilon} \mathbf{N}_\sigma^T \mathbf{N}_\varepsilon \, d\Omega = \mathbf{I} \quad (1.33)$$

where  $\mathbf{I}$  is the  $[n_\varepsilon \times n_\varepsilon]$  identity matrix. If (1.33) holds, the variables  $\hat{\boldsymbol{\sigma}}$  and  $\hat{\boldsymbol{\varepsilon}}$  are said to be *generalized variables*. Possible choices for the stress ansatz  $\mathbf{N}_\sigma$  so that (1.33) is satisfied are

- first possible choice

$$\mathbf{N}_\sigma = \mathbf{D}\mathbf{N}_\varepsilon \left( \int_{\Omega_e} \mathbf{N}_\varepsilon^\top \mathbf{D}\mathbf{N}_\varepsilon d\Omega \right)^{-1} = \mathbf{D}\mathbf{N}_\varepsilon \mathbf{E}^{-1} \quad (1.34)$$

where the square  $[n_\varepsilon \times n_\varepsilon]$  invertible *elastic matrix*  $\mathbf{E}$  is given by

$$\mathbf{E} = \int_{\Omega_e} \mathbf{N}_\varepsilon^\top \mathbf{D}\mathbf{N}_\varepsilon d\Omega \quad (1.35)$$

- second possible choice, proposed by Corradi in [23] for elasto-plasticity

$$\mathbf{N}_\sigma = \mathbf{N}_\varepsilon \left( \int_{\Omega_e} \mathbf{N}_\varepsilon^\top \mathbf{N}_\varepsilon d\Omega \right)^{-1} = \mathbf{N}_\varepsilon \mathbf{G}^{-1} \quad (1.36)$$

where the square  $[n_\varepsilon \times n_\varepsilon]$  invertible matrix  $\mathbf{G}$  is given by

$$\mathbf{G} = \int_{\Omega_e} \mathbf{N}_\varepsilon^\top \mathbf{N}_\varepsilon d\Omega \quad (1.37)$$

Exploiting the three models for the unknown fields given by (1.21), (1.22) and (1.23), the discretized functional  $\Pi_e$  given in (1.18) can be approximated as the function

$$\begin{aligned} \Pi_e^h : \mathbb{R}^{n_u} \times \mathbb{R}^{n_\varepsilon} \times \mathbb{R}^{n_\sigma} &\rightarrow \mathbb{R} \\ \Pi_e^h(\hat{\mathbf{u}}, \hat{\boldsymbol{\varepsilon}}, \hat{\boldsymbol{\sigma}}) &= \frac{1}{2} \hat{\boldsymbol{\varepsilon}}^\top \left( \int_{\Omega_e} \mathbf{N}_\varepsilon^\top \mathbf{D}\mathbf{N}_\varepsilon d\Omega \right) \hat{\boldsymbol{\varepsilon}} - \hat{\boldsymbol{\sigma}}^\top \left( \int_{\Omega_e} \mathbf{N}_\sigma^\top (\mathbf{N}_\varepsilon \hat{\boldsymbol{\varepsilon}} - \mathbf{S}\mathbf{N}_u \hat{\mathbf{u}}) d\Omega \right) + \\ &\quad - \hat{\mathbf{u}}^\top \left( \int_{\Omega_e} \mathbf{N}_u^\top \mathbf{b} d\Omega + \int_{\partial_p \Omega_e} \mathbf{N}_u^\top \mathbf{p} d\Sigma \right) \end{aligned} \quad (1.38)$$

which, by (1.33), becomes

$$\Pi_e^h(\hat{\mathbf{u}}, \hat{\boldsymbol{\varepsilon}}, \hat{\boldsymbol{\sigma}}) = \frac{1}{2} \hat{\boldsymbol{\varepsilon}}^\top \mathbf{E} \hat{\boldsymbol{\varepsilon}} - \hat{\boldsymbol{\sigma}}^\top (\hat{\boldsymbol{\varepsilon}} - \mathbf{C} \hat{\mathbf{u}}) - \hat{\mathbf{u}}^\top \mathbf{F}_e \quad (1.39)$$

where  $\mathbf{E}$  is the  $[n_\varepsilon \times n_\varepsilon]$  elastic matrix already defined in (1.35) and where the expressions below have been introduced

- $[n_\varepsilon \times n_u]$  compatibility matrix, obtained by the choice made in (1.36)

$$\mathbf{C} = \int_{\Omega_e} \mathbf{N}_\sigma^\top \mathbf{S}\mathbf{N}_u d\Omega = \mathbf{G}^{-1} \int_{\Omega_e} \mathbf{N}_\varepsilon^\top \mathbf{S}\mathbf{N}_u d\Omega = \mathbf{G}^{-1} \mathbf{A} \quad (1.40)$$

where the  $[n_\varepsilon \times n_u]$  matrix  $\mathbf{A}$  is

$$\mathbf{A} = \int_{\Omega_e} \mathbf{N}_\varepsilon^T \mathbf{S} \mathbf{N}_u d\Omega \quad (1.41)$$

- $[n_u \times 1]$  local equivalent nodal forces vector

$$\mathbf{F}_e = \int_{\Omega_e} \mathbf{N}_u^T \mathbf{b} d\Omega + \int_{\partial_p \Omega_e} \mathbf{N}_u^T \mathbf{p} d\Sigma \quad (1.42)$$

In (1.42) the term *nodal* is placed here to maintain consistency with standard FE nomenclature. However, as will be clearer in the following chapters, the vector of discrete unknowns  $\hat{\mathbf{u}}$  (and hence  $\mathbf{F}_e$ ) may not contain only nodal values of the corresponding field, but different quantities, not necessarily leading to a physical interpretation or simple visualization.

### 1.3.2. Stationarity of the discretized mixed functional

By enforcing the stationarity of the mixed discrete functional (1.39) with respect to the variables  $\hat{\mathbf{u}}$ ,  $\hat{\boldsymbol{\varepsilon}}$  and  $\hat{\boldsymbol{\sigma}}$  one obtains the algebraic governing equations

- equilibrium

$$\partial_{\hat{\mathbf{u}}} \Pi_e^h = \mathbf{0} \implies \mathbf{C}^T \hat{\boldsymbol{\sigma}} = \mathbf{F}_e \quad (1.43)$$

- constitutive law

$$\partial_{\hat{\boldsymbol{\varepsilon}}} \Pi_e^h = \mathbf{0} \implies \hat{\boldsymbol{\sigma}} = \mathbf{E} \hat{\boldsymbol{\varepsilon}} \quad (1.44)$$

- kinematic compatibility

$$\partial_{\hat{\boldsymbol{\sigma}}} \Pi_e^h = \mathbf{0} \implies \hat{\boldsymbol{\varepsilon}} = \mathbf{C} \hat{\mathbf{u}} \quad (1.45)$$

Replacing (1.45) in (1.44) and (1.44) in (1.43) the following local algebraic system is obtained

$$(\mathbf{C}^T \mathbf{E} \mathbf{C}) \hat{\mathbf{u}} = \mathbf{F}_e \quad (1.46)$$

which in more compact form becomes

$$\mathbf{K}_e^c \hat{\mathbf{u}} = \mathbf{F}_e \quad (1.47)$$

where  $\mathbf{K}_e^c$  is the  $[n_u \times n_u]$  symmetric, positive-semi definite *local stiffness matrix consistent with the strain and displacement models*. Positive definiteness is not yet achieved since at

this stage the matrix  $\mathbf{K}_e^c$  is *at least* 6 times singular, corresponding to the 6 rigid motions a body can undergo in 3D.

An analysis of the degree of singularity of the matrix  $\mathbf{K}_e^c$  is needed to ensure the *stability* of the element. If  $n_u - n_\varepsilon \leq 6$  and  $\mathbf{C}$  has rank equal to  $n_u - 6$ , then  $\mathbf{K}_e^c$  has the correct degree of singularity (6), corresponding to the six orthogonal rigid body modes in 3D. Conversely, if  $n_u - n_\varepsilon > 6$ ,  $\mathbf{K}_e^c$  has a surplus of rank deficiency  $n_u - n_\varepsilon - 6$  and *zero-energy modes* can appear, known in the finite element solid mechanics community as *hourglass modes* (see Figure 1.3), from the most common characteristic shape exhibited by 2D quadrilateral elements, and often appear in mixed finite element formulations. These modes are spurious deformations as they represent a configuration of indefinite displacements that the element can exhibit under null external forces. It is a trivial consequence of the eigenvalue problem

$$\mathbf{K}_e^c \mathbf{V}_e = \mathbf{\Lambda}_e \mathbf{F}_e$$

where the  $[n_u \times n_u]$  matrix  $\mathbf{V}_e$  collects the  $[n_u \times 1]$  eigenvectors  $\mathbf{v}_e^{(i)}$  corresponding to the eigenvalues  $\lambda_e^{(i)}$  gathered on the diagonal matrix  $\mathbf{\Lambda}_e$ , with  $i = 1, \dots, n_u$ . It is clear that an eigenvalue  $\lambda_e^{(i)}$  being null implies that the load multiplier of the local equivalent nodal forces  $\mathbf{F}_e$  is zero, producing a possibly non-zero indefinite displaced configuration described by the corresponding eigenvector  $\mathbf{v}_e^{(i)}$  in the kernel of  $\mathbf{K}_e^c$ . If the null eigenvalues of the consistent matrix  $\mathbf{K}_e^c$  are more than 6, the chosen strain field is *not rich enough* to correctly represent all the configurations of the chosen displacement field, and hence require some stabilizing technique to allow the element to be used.

### 1.3.3. Hourglass modes stabilization

The key idea to control hourglass modes and suppress the possible rise in the approximated solution of configurations such as those displayed in Figure 1.3 is to add a fictitious stiffness to the element. The mixed continuous functional (1.18) becomes

$$\begin{aligned} \Pi_e(\mathbf{u}, \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}_H, \boldsymbol{\sigma}) &= \frac{1}{2} \int_{\Omega_e} \boldsymbol{\varepsilon}^T(\boldsymbol{\xi}) \mathbf{D} \boldsymbol{\varepsilon}(\boldsymbol{\xi}) d\Omega + \frac{1}{2} \int_{\Omega_e} \boldsymbol{\varepsilon}_H^T(\boldsymbol{\xi}) \mathbf{D}_H \boldsymbol{\varepsilon}_H(\boldsymbol{\xi}) d\Omega + \\ &\quad - \int_{\Omega_e} \boldsymbol{\sigma}^T(\boldsymbol{\xi}) (\boldsymbol{\varepsilon}(\boldsymbol{\xi}) - \mathbf{S} \mathbf{u}(\boldsymbol{\xi})) d\Omega - \int_{\Omega_e} \mathbf{u}^T(\boldsymbol{\xi}) \mathbf{b}(\boldsymbol{\xi}) d\Omega + \\ &\quad - \int_{\partial_p \Omega_e} \mathbf{u}^T(\boldsymbol{\xi}) \mathbf{p}(\boldsymbol{\xi}) d\Sigma \end{aligned} \quad (1.48)$$

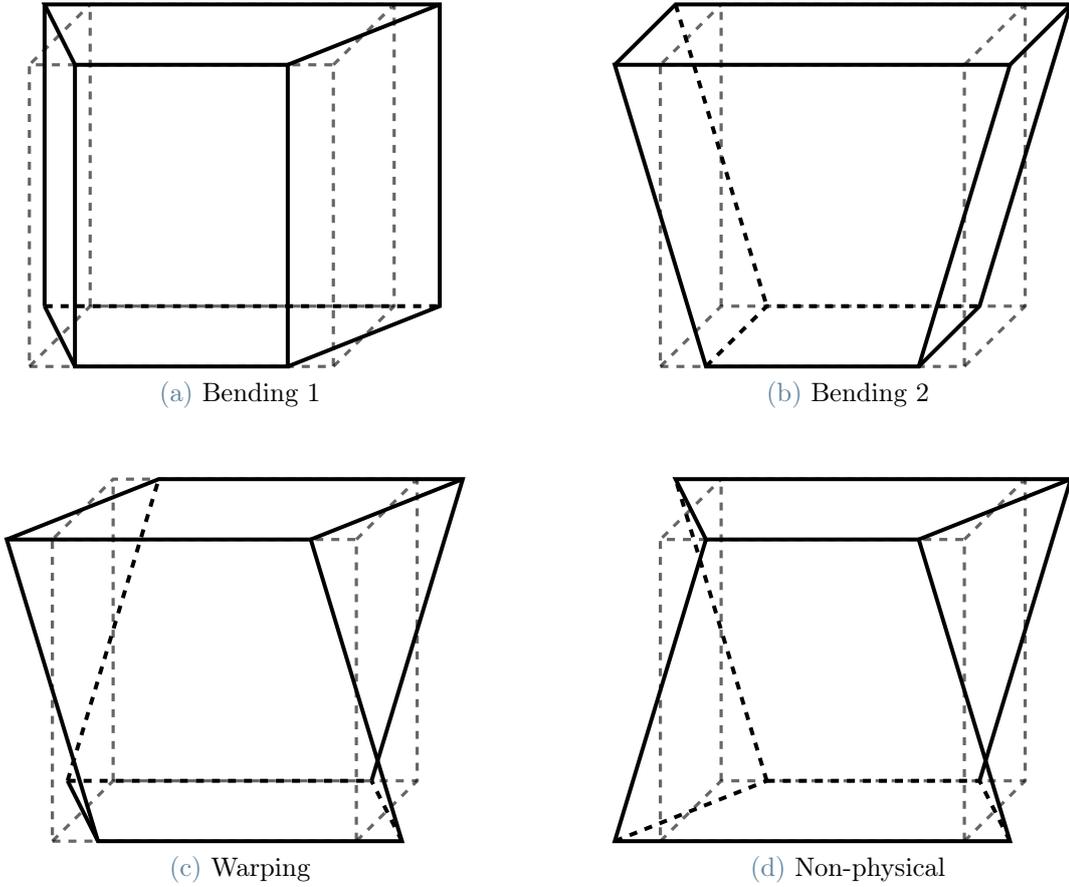


Figure 1.3: Some hourglass modes for a cubic element.

where here  $\boldsymbol{\varepsilon}$  is the deformation field obtained by removing the deformation induced by hourglass modes  $\boldsymbol{\varepsilon}_H$  and  $\mathbf{D}_H$  is a *hourglass fictitious material stiffness matrix*. A possible choice, among many others, for  $\mathbf{D}_H$  is

$$\mathbf{D}_H = \frac{1}{6} \text{tr}(\mathbf{D}) \quad (1.49)$$

where  $\text{tr}(\cdot)$  stands for the trace operator of matrix  $(\cdot)$ . The discretized form of (1.48) becomes

$$\Pi_e^h(\hat{\mathbf{u}}, \hat{\mathbf{u}}_H, \hat{\boldsymbol{\varepsilon}}, \hat{\boldsymbol{\sigma}}) = \frac{1}{2} \hat{\boldsymbol{\varepsilon}}^T \mathbf{E} \hat{\boldsymbol{\varepsilon}} + \frac{1}{2} \hat{\mathbf{u}}_H^T \left( \int_{\Omega_e} \mathbf{B}^T \mathbf{D}_H \mathbf{B} d\Omega \right) \hat{\mathbf{u}}_H - \hat{\boldsymbol{\sigma}}^T (\boldsymbol{\varepsilon} - \mathbf{C} \hat{\mathbf{u}}) - \hat{\mathbf{u}}^T \mathbf{F}_e \quad (1.50)$$

where the hourglass strains  $\boldsymbol{\varepsilon}_H$  have been approximated with the  $[n_H \times 1]$  vector of *unknown* hourglass discrete displacements  $\hat{\mathbf{u}}_H$  through the standard compatibility operator in finite elements  $\mathbf{B}$

$$\boldsymbol{\varepsilon}_H(\boldsymbol{\xi}) = \mathbf{S} \mathbf{N}_u(\boldsymbol{\xi}) \hat{\mathbf{u}}_H = \mathbf{B}(\boldsymbol{\xi}) \hat{\mathbf{u}}_H \quad (1.51)$$

At this point the goal is to extract the vector  $\hat{\mathbf{u}}_H$  from the vector  $\hat{\mathbf{u}}$ . By splitting the displacement parameters  $\hat{\mathbf{u}}$  in a  $[n_u \times 1]$  vector describing the deformations and rigid body motions  $\hat{\mathbf{u}}_{D+R}$  and a  $[n_u \times 1]$  vector describing the hourglass modes  $\hat{\mathbf{u}}_H$ , the displacement field can be expressed as

$$\mathbf{u}(\boldsymbol{\xi}) = \mathbf{N}_u(\boldsymbol{\xi})\hat{\mathbf{u}} = \mathbf{N}_u(\boldsymbol{\xi})(\hat{\mathbf{u}}_{D+R} + \hat{\mathbf{u}}_H) = \mathbf{N}_u(\boldsymbol{\xi})(\mathbf{T}_{D+R}\hat{\boldsymbol{\rho}}_{D+R} + \mathbf{T}_H\hat{\boldsymbol{\rho}}_H) \quad (1.52)$$

where the vectors  $\hat{\mathbf{u}}_{D+R}$  and  $\hat{\mathbf{u}}_H$  have been described by the *natural parameters*  $[n_{D+R} \times 1]$  vector  $\hat{\boldsymbol{\rho}}_{D+R}$  and  $[n_H \times 1]$  vector  $\hat{\boldsymbol{\rho}}_H$  through the  $[n_u \times n_{D+R}]$  matrix  $\mathbf{T}_{D+R}$  and  $[n_u \times n_H]$  matrix  $\mathbf{T}_H$  respectively. By matrix augmentation it is possible to obtain the square and invertible  $[n_u \times n_u]$  matrix

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_{D+R} & \mathbf{T}_H \end{bmatrix} \quad (1.53)$$

and gathering the natural parameters in the  $[n_u \times 1]$  vector  $\hat{\boldsymbol{\rho}}$

$$\hat{\boldsymbol{\rho}} = \begin{Bmatrix} \hat{\boldsymbol{\rho}}_{D+R} \\ \hat{\boldsymbol{\rho}}_H \end{Bmatrix} \quad (1.54)$$

so that

$$\hat{\mathbf{u}} = \mathbf{T}\hat{\boldsymbol{\rho}} \quad (1.55)$$

An important property of the decomposition described in (1.52) is the *orthogonality* between deformative or rigid body modes and hourglass modes

$$(\hat{\mathbf{u}}_{D+R})^T \hat{\mathbf{u}}_H = \mathbf{0} \quad (1.56)$$

which translates into

$$(\mathbf{T}_{D+R})^T \mathbf{T}_H = \mathbf{0} \quad (1.57)$$

Exploiting this last condition applied to (1.52) yields to

$$(\mathbf{T}_{D+R})^T \hat{\mathbf{u}} = (\mathbf{T}_{D+R})^T \mathbf{T}_{D+R} \hat{\boldsymbol{\rho}}_{D+R} + (\mathbf{T}_{D+R})^T \mathbf{T}_H \hat{\boldsymbol{\rho}}_H = (\mathbf{T}_{D+R})^T \mathbf{T}_{D+R} \hat{\boldsymbol{\rho}}_{D+R} \quad (1.58)$$

so that the natural parameters corresponding to deformative or rigid body modes  $\hat{\boldsymbol{\rho}}_{D+R}$  can be extracted

$$\hat{\boldsymbol{\rho}}_{D+R} = [(\mathbf{T}_{D+R})^T \mathbf{T}_{D+R}]^{-1} (\mathbf{T}_{D+R})^T \hat{\mathbf{u}} \quad (1.59)$$

and plugged into (1.52) to obtain

$$\hat{\mathbf{u}} = \mathbf{T}_{D+R} \left\{ [(\mathbf{T}_{D+R})^T \mathbf{T}_{D+R}]^{-1} (\mathbf{T}_{D+R})^T \hat{\mathbf{u}} \right\} + \mathbf{T}_H \hat{\mathbf{p}}_H \quad (1.60)$$

and hence, rearranging,  $\hat{\mathbf{u}}_H$  is finally obtained

$$\hat{\mathbf{u}}_H = \mathbf{T}_H \hat{\mathbf{p}}_H = \left\{ \mathbf{I} - \mathbf{T}_{D+R} [(\mathbf{T}_{D+R})^T \mathbf{T}_{D+R}]^{-1} (\mathbf{T}_{D+R})^T \right\} \hat{\mathbf{u}} = \mathbf{H} \hat{\mathbf{u}} \quad (1.61)$$

In the above equation the  $[n_u \times n_u]$  *hourglass matrix*  $\mathbf{H}$  has been introduced

$$\mathbf{H} = \mathbf{I} - \mathbf{T}_{D+R} [(\mathbf{T}_{D+R})^T \mathbf{T}_{D+R}]^{-1} (\mathbf{T}_{D+R})^T \quad (1.62)$$

and it allows the computation of the extra term responsible for the stabilization contained in the functional (1.50) through the construction of matrix  $\mathbf{T}_{D+R}$ :

$$\Pi_e^h(\hat{\mathbf{u}}, \hat{\boldsymbol{\varepsilon}}, \hat{\boldsymbol{\sigma}}) = \frac{1}{2} \hat{\boldsymbol{\varepsilon}}^T \mathbf{E} \hat{\boldsymbol{\varepsilon}} + \frac{1}{2} \hat{\mathbf{u}}^T \mathbf{K}_e^s \hat{\mathbf{u}} - \hat{\boldsymbol{\sigma}}^T (\boldsymbol{\varepsilon} - \mathbf{C} \hat{\mathbf{u}}) - \hat{\mathbf{u}}^T \mathbf{F}_e \quad (1.63)$$

where  $\mathbf{K}_e^s$  is the  $[n_u \times n_u]$  *local stabilizing stiffness matrix*

$$\mathbf{K}_e^s = \mathbf{H}^T \left( \int_{\Omega_e} \mathbf{B}^T \mathbf{D}_H \mathbf{B} d\Omega \right) \mathbf{H} \quad (1.64)$$

The integral contained in round brackets has to be approximated since it contains the fictitious matrix  $\mathbf{D}_H$  and the differential operator  $\mathbf{B}$  which in turn includes the shape functions for the displacement field  $\mathbf{N}_u$ , not always explicitly known, as will be in the case of the virtual element method. The stabilizing local stiffness matrix is required to scale with respect to the elements and meshes partitioning the domain the same way the local consistent matrix does ([10, 56]). Under a suitable choice of the degrees of freedom (so that they scale as 1, as will be clear in the following chapters), the above integral is hence required to scale according only to the problem at hands and the dimension it is embedded in. For second-order differential problems (as stationary elasticity) it has to scale as 1 in two dimensions, and as  $\frac{1}{h}$  in three dimensions.

A possible choice to approximate such integral is the *scalar-based stabilization*, used e.g. in [5], and consisting to set it to  $\frac{1}{2} \text{tr}(\mathbf{K}_e^c) \mathbf{I}$ , leading to

$$\mathbf{K}_e^s = \frac{1}{2} \text{tr}(\mathbf{K}_e^c) \mathbf{H}^T \mathbf{H} \quad (1.65)$$

Another choice (presented, e.g., in [53]) is to perform *diagonal matrix-based stabilization*,

approximating the integral of (1.64) with a diagonal matrix  $\Lambda$ , whose elements in row  $i$  and column  $j$  are obtained as follows

$$[\Lambda]_{ij} = \delta_{ij} \max \left\{ [\mathbf{K}_e^c]_{ij}, \frac{\alpha_0 h_e^{d-2}}{n_D} \text{tr}(\mathbf{D}) \right\} \quad (1.66)$$

where  $\alpha_0$  is a coefficient between 0 and 1 providing a lower bound for the diagonal matrix  $\Lambda$  and could be set as  $\frac{1}{3}$  in 2D and  $\frac{1}{9}$  in 3D,  $d$  is the dimension of the embedded space and  $n_D$  is the dimension of the matrix  $\mathbf{D}$  (3 in 2D and 6 in 3D). The expression (1.64) then becomes

$$\mathbf{K}_e^s = \mathbf{H}^T \Lambda \mathbf{H} \quad (1.67)$$

**Proposition 1.1.** *The hourglass matrix  $\mathbf{H}$  fulfills the property*

$$\mathbf{H} = \mathbf{H}^T \mathbf{H} \quad (1.68)$$

*Proof.* By direct substitution

$$\begin{aligned} \mathbf{H}^T \mathbf{H} &= \left\{ \mathbf{I} - \mathbf{T}_{D+R} [(\mathbf{T}_{D+R})^T \mathbf{T}_{D+R}]^{-1} (\mathbf{T}_{D+R})^T \right\}^T \left\{ \mathbf{I} - \mathbf{T}_{D+R} [(\mathbf{T}_{D+R})^T \mathbf{T}_{D+R}]^{-1} \right. \\ &\quad \left. (\mathbf{T}_{D+R})^T \right\} = \mathbf{I} - \mathbf{T}_{D+R} [(\mathbf{T}_{D+R})^T \mathbf{T}_{D+R}]^{-1} (\mathbf{T}_{D+R})^T + \\ &\quad - \left\{ \mathbf{T}_{D+R} [(\mathbf{T}_{D+R})^T \mathbf{T}_{D+R}]^{-1} (\mathbf{T}_{D+R})^T \right\}^T + \left\{ \mathbf{T}_{D+R} [(\mathbf{T}_{D+R})^T \mathbf{T}_{D+R}]^{-1} \right. \\ &\quad \left. (\mathbf{T}_{D+R})^T \right\}^T \mathbf{T}_{D+R} [(\mathbf{T}_{D+R})^T \mathbf{T}_{D+R}]^{-1} (\mathbf{T}_{D+R})^T = \\ &= \mathbf{I} - \mathbf{T}_{D+R} [(\mathbf{T}_{D+R})^T \mathbf{T}_{D+R}]^{-1} (\mathbf{T}_{D+R})^T - \mathbf{T}_{D+R} [(\mathbf{T}_{D+R})^T \mathbf{T}_{D+R}]^{-T} \\ &\quad (\mathbf{T}_{D+R})^T + \mathbf{T}_{D+R} [(\mathbf{T}_{D+R})^T \mathbf{T}_{D+R}]^{-T} (\mathbf{T}_{D+R})^T \mathbf{T}_{D+R} [(\mathbf{T}_{D+R})^T \mathbf{T}_{D+R}]^{-1} \\ &\quad (\mathbf{T}_{D+R})^T = \mathbf{I} - \mathbf{T}_{D+R} [(\mathbf{T}_{D+R})^T \mathbf{T}_{D+R}]^{-1} (\mathbf{T}_{D+R})^T = \mathbf{H} \end{aligned}$$

□

In view of Proposition 1.1, the choice for  $\mathbf{K}_e^s$  described in (1.65) results particularly appealing as can be further simplified in

$$\mathbf{K}_e^s = \frac{1}{2} \text{tr}(\mathbf{K}_e^c) \mathbf{H} \quad (1.69)$$

Enforcing the stationarity of the mixed discrete stabilized functional (1.63) with respect to  $\hat{\varepsilon}$  and  $\hat{\sigma}$  one obtains the same equations (1.44) and (1.3), while setting the derivative

with respect to  $\hat{\mathbf{u}}$  to 0 one obtains

$$\partial_{\hat{\mathbf{u}}}\Pi_e^h = \mathbf{0} \implies \mathbf{C}^T \hat{\boldsymbol{\sigma}} + \mathbf{K}_e^s \hat{\mathbf{u}} = \mathbf{F}_e \quad (1.70)$$

which implies, recombining the three algebraic systems

$$(\mathbf{K}_e^c + \mathbf{K}_e^s) \hat{\mathbf{u}} = \mathbf{F}_e \quad (1.71)$$

where  $\mathbf{K}_e = \mathbf{K}_e^c + \mathbf{K}_e^s$  is the *local stiffness matrix*, having the correct degree of singularity corresponding to the 6 rigid body motions in the three dimensional space.

### 1.3.4. Assembly of the global system

To obtain the solution of the discrete unknown field describing the approximated problem, an *assembly* procedure is required, gathering the contributions of the single finite elements, here referred to as *local* quantities, and building the *global* algebraic system to be passed to the solver. More precisely, the assembly operator is applied to the local stiffness matrix  $\mathbf{K}_e$  to produce the global stiffness matrix  $\mathbf{K}$

$$\mathbf{K} = \underset{e=1}{\overset{n_e}{\mathbf{A}}} \mathbf{K}_e \quad (1.72)$$

and to the local equivalent nodal forces vector  $\mathbf{F}_e$  to produce the global equivalent nodal forces vector

$$\mathbf{F} = \underset{e=1}{\overset{n_e}{\mathbf{A}}} \mathbf{F}_e \quad (1.73)$$

$\mathbf{A}$  is the standard FE assembly operator, formally composed of a set of  $n_e$   $[n_u^e \times n_u]$  logical matrices  $\mathbf{L}_e$  pre- and post-multiplying the local stiffness matrices as

$$\underset{e}{\mathbf{A}}(\cdot) = \mathbf{L}_e^T(\cdot) \mathbf{L}_e$$

and mapping the vector of equivalent nodal forces as

$$\underset{e}{\mathbf{A}}(\cdot) = \mathbf{L}_e^T(\cdot)$$

where the  $\mathbf{L}_e$  non-zero entries correspond to the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column if the local degree of freedom  $i$  is represented in the global degree of freedom vector at position  $j$ . The assembly operator can also take into account suitable transformations where rotations or other mappings between local and global kinematic descriptions are needed. In common

practice the assembly operation is realized through incidence matrices, given the natural waste of memory and computational resources implied by the sparse matrices  $\mathbf{L}_e$ .

### 1.3.5. Enforcement of boundary conditions and solution

The assembled algebraic system in the  $n_u$  unknowns  $\mathbf{U}$  reads

$$\mathbf{K}\mathbf{U} = \mathbf{F} \quad (1.74)$$

which is 6 times singular, as boundary conditions have not been imposed yet, and cannot be solved. To account for the assigned displacements  $\bar{\mathbf{u}}$  on the constrained part of the boundary  $\partial\Omega_u$  the most common approach is the *Guyan reduction* ([38]), also known as *static condensation*. The method reduces the number of degrees of freedom of the system by performing a partition of the vector  $\mathbf{U}$  in a *free* part  $\mathbf{U}_f$  and a *constrained* part  $\mathbf{U}_c$  where displacements are prescribed by the problem. Formally, this coincides with the finite element counterpart of the usual operation of lifting the boundary datum performed for the continuum problem with non-homogeneous Dirichlet conditions. Indeed, lifting is achieved through functions whose support is limited to the only layer of elements of the partition that face the boundary. If the partition of the unknowns  $\mathbf{U}$  is applied, (1.74) is equivalent to

$$\begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fc} \\ \mathbf{K}_{cf} & \mathbf{K}_{cc} \end{bmatrix} \begin{Bmatrix} \mathbf{U}_f \\ \mathbf{U}_c \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_f \\ \mathbf{F}_c \end{Bmatrix} \quad (1.75)$$

where also the matrix  $\mathbf{K}$  and the vector  $\mathbf{F}$  are being partitioned accordingly. The sub-matrix  $\mathbf{K}_{ff}$  is non-singular as rigid body motions are prevented. Hence, it can be inverted and the free part of the unknowns  $\mathbf{U}_f$  can be directly found by *condensing* the known displacements  $\mathbf{U}_c$

$$\mathbf{U}_f = \mathbf{K}_{ff}^{-1}(\mathbf{F}_f - \mathbf{K}_{fc}\mathbf{U}_c) \quad (1.76)$$

Assembling back the vector  $\mathbf{U}$  is straightforward by recomposing the partition and the reaction forces for the degrees of freedom belonging to the constrained boundary can be found by plugging  $\mathbf{U}_f$  in the system given in the second row of (1.75)

$$\mathbf{F}_c = \mathbf{K}_{fc}^T \mathbf{U}_f + \mathbf{K}_{cc} \mathbf{U}_c \quad (1.77)$$

where the identity  $\mathbf{K}_{fc}^T = \mathbf{K}_{cf}$  implied by the symmetry of the original matrix has been exploited.

### 1.3.6. Strains and stresses recovery

Once the unknown discrete field  $\mathbf{U}$  is found, it is possible to recover the local degrees of freedom  $\hat{\mathbf{u}}$  and reconstruct the local continuous field  $\mathbf{u}(\boldsymbol{\xi})$  through the selected model for the displacements  $\mathbf{N}_u(\boldsymbol{\xi})$  (if available) by (1.21). The global field is then piecewise defined by the local fields after switching back to global coordinates  $\mathbf{x}$ . Recovering the local strain field  $\boldsymbol{\varepsilon}(\boldsymbol{\xi})$  is similarly achieved exploiting (1.22) and (1.45), so that

$$\boldsymbol{\varepsilon}(\boldsymbol{\xi}) = \mathbf{N}_\varepsilon(\boldsymbol{\xi})\hat{\boldsymbol{\varepsilon}} = \mathbf{N}_\varepsilon(\boldsymbol{\xi})\mathbf{C}\hat{\mathbf{u}} \quad (1.78)$$

Analogously, the local stress field  $\boldsymbol{\sigma}(\boldsymbol{\xi})$  is obtained by the condition prescribed by the generalized variables in (1.36), and the model (1.23) and identity in (1.44), so that

$$\boldsymbol{\sigma}(\boldsymbol{\xi}) = \mathbf{N}_\sigma(\boldsymbol{\xi})\hat{\boldsymbol{\sigma}} = \mathbf{N}_\varepsilon(\boldsymbol{\xi})\mathbf{G}^{-1}\hat{\boldsymbol{\sigma}} = \mathbf{N}_\varepsilon(\boldsymbol{\xi})\mathbf{G}^{-1}\mathbf{E}\mathbf{C}\hat{\mathbf{u}} \quad (1.79)$$

or, if one adopts the stress model from (1.34),

$$\boldsymbol{\sigma}(\boldsymbol{\xi}) = \mathbf{N}_\sigma(\boldsymbol{\xi})\hat{\boldsymbol{\sigma}} = \mathbf{D}\mathbf{N}_\varepsilon(\boldsymbol{\xi})\mathbf{E}^{-1}\hat{\boldsymbol{\sigma}} = \mathbf{D}\mathbf{N}_\varepsilon(\boldsymbol{\xi})\mathbf{C}\hat{\mathbf{u}} \quad (1.80)$$

It is a common practice in FE programs to output the global displacement field as generated by the shape functions (if available) mapping the discrete nodal values. For strains and stresses which are often discontinuous across the elements boundaries, their values at a generic point of the computational domain  $\mathbf{x}$  is obtained as an interpolation of the neighbouring sampling points. These latter are approximated by various averaging techniques as algebraic, volume, or strain energy mean.

### 1.3.7. Summary of the scheme

A concise list of the steps followed by the above-presented approach for the solution of the elastostatic problem through the Hu-Washizu mixed finite element approximation is reported below.

- discretization of the domain  $\Omega$  in  $\mathcal{P}_h$
- definition of the strain  $\mathbf{N}_\varepsilon$  and displacement  $\mathbf{N}_u$  models
- computation of matrices  $\mathbf{A}$ ,  $\mathbf{G}$  and the compatibility matrix  $\mathbf{C}$
- computation of the local stiffness matrix consistent with the strain and displacement models  $\mathbf{K}_e^c$

- computation of the transformation matrix  $\mathbf{T}_{D+R}$ , hourglass matrix  $\mathbf{H}$  and local stabilizing stiffness matrix  $\mathbf{K}_e^s$
- computation of the local equivalent nodal forces vector  $\mathbf{F}_e$
- assembly of the global system from the local quantities
- enforcement of the Dirichlet boundary conditions
- solution of the global algebraic system

$$\mathbf{KU} = \mathbf{F}$$

- recover of the strains and stresses parameters and reconstruction of the three continuous fields of the problem

## 2 | Tools of the virtual element method

The mixed formulation described in Chapter 1 allows to implement very general schemes, amongst which the *virtual element method* (VEM). Before moving to applying it to the problem of linear elastostatics, a mathematical description of the tools employed by the method is necessary. The goal of this chapter is to present the key points of this technique: the polytopic mesh, the virtual element spaces and the projection operators.

### 2.1. Mesh

The *virtual element mesh* is one of the striking features this technique is mostly known for, as it can be of very general nature, including non-convex elements, aligned edges and faces, geometrically hanging nodes (Figure 2.1), and extremely complicated and diverse elements ([54]). In principle, the method is even suitable to support elements with curved edges ([31, 33]), allowing an incredibly accurate description of the geometries of the domain. However, in the subsequent sections only meshes containing straight edges and plane faces will be considered.

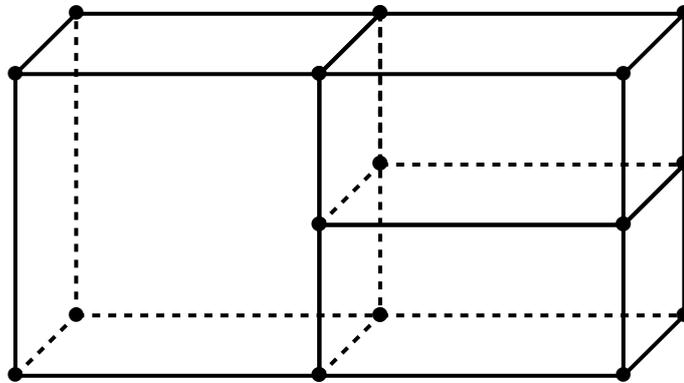


Figure 2.1: Hanging nodes situation for the element on the left.

The body represented in Figure 1.1 by the open bounded set  $\Omega \subset \mathbb{R}^3$  can be partitioned

in a finite collection  $\mathcal{P}_h$  of  $n_p$  non-overlapping polyhedra  $P$ , such that

$$\Omega \approx \bigcup_{P \in \mathcal{P}_h} P$$

Each polyhedron  $P$  has a centroid  $\mathbf{x}_P$ , a diameter  $h_P$ , and a volume  $|P|$ . The polyhedron is described by a set of  $n_{p,f}$  oriented planar polygons representing its facets  $F$ , i.e. its *2-dimensional faces* ([46]), from here on simply referred to as *faces*. The boundary of the polyhedron  $\partial P$  then becomes

$$\partial P = \bigcup_{F \in \partial P} F$$

Each face  $F$  has a centroid  $\mathbf{x}_F$ , a diameter  $h_F$ , and an area  $|F|$ . The face is defined by a sequence of  $n_{p,f,e}$  edges representing its facets  $E$ , i.e. its *1-dimensional faces*, from now on simply called edges. The boundary of the face  $\partial F$  is

$$\partial F = \bigcup_{E \in \partial F} E$$

Each edge  $E$  is then defined by the set of the two vertices  $V_1$  and  $V_2$  it is composed of, i.e. its *0-dimensional faces*. From the above construction it is evident how the polytopic virtual element mesh can handle very general situations.

Some remarks have to be specified concerning the description of the spatial entities. Local non-dimensional coordinates as described in (1.19) will be adopted through a simple linear mapping between the local and global reference systems. In VEM, there is no *parent element* as for *isoparametric FE*, where nonlinear maps are needed, limiting the set of allowable shapes. Scaled monomials will be extensively used throughout the subsequent discussion. In 2D they are defined for polygons as

$$m_{\boldsymbol{\alpha}}(\mathbf{x}) := \left( \frac{\mathbf{x} - \mathbf{x}_F}{h_F} \right)^{\boldsymbol{\alpha}} \quad (2.1)$$

where  $\boldsymbol{\alpha} = (\alpha_{\xi}, \alpha_{\eta})$  is a multiindex, and  $|\boldsymbol{\alpha}| = \alpha_{\xi} + \alpha_{\eta}$  its order. A one-to-one correspondence can be established between the scaled monomials of (2.1) and the indices  $\alpha \in \mathbb{N}$ , starting from index  $\alpha = 1$  corresponding to the constant monomial 1, and following each row, from left to right, of *Pascal's triangle* (Figure 2.2). Given an integer  $k$  the number of parameters  $n_k$  necessary to describe a polynomial in  $\mathcal{P}_k(F)$  is given by the number of total elements up to row  $k$  of Pascal's triangle, that is

$$n_k = \dim \mathcal{P}_k(F) = \frac{(k+1)(k+2)}{2} \quad (2.2)$$

$$\begin{array}{cccccccc}
k = 0 & & & & & & & 1 \\
k = 1 & & & & \xi & & \eta & \\
k = 2 & & & \xi^2 & \xi\eta & & \eta^2 & \\
k = 3 & & \xi^3 & \xi^2\eta & \xi\eta^2 & & \eta^3 & \\
k = 4 & \xi^4 & \xi^3\eta & \xi^2\eta^2 & \xi\eta^3 & & \eta^4 & \\
k = 5 & \xi^5 & \xi^4\eta & \xi^3\eta^2 & \xi^2\eta^3 & \xi\eta^4 & \eta^5 & \\
k = 6 & \xi^6 & \xi^5\eta & \xi^4\eta^2 & \xi^3\eta^3 & \xi^2\eta^4 & \xi\eta^5 & \eta^6
\end{array}$$

Figure 2.2: Pascal's triangle truncated at  $k = 6$ .

Therefore, the scaled monomials of degree less or equal to  $k$  can be gathered in the  $[n_k \times 1]$  vector  $\mathbf{m}_k$

$$\mathbf{m}_k := \{1 \quad \xi \quad \eta \quad \xi^2 \quad \dots \quad \eta^k\}^T \quad (2.3)$$

whose elements form a basis for the polynomials  $\mathcal{P}_k(F)$  of degree less or equal to  $k$ .

In 3D scaled monomials are defined for polyhedra as

$$\mu_{\alpha}(\mathbf{x}) := \left( \frac{\mathbf{x} - \mathbf{x}_P}{h_P} \right)^{\alpha} \quad (2.4)$$

where  $\alpha = (\alpha_{\xi}, \alpha_{\eta}, \alpha_{\zeta})$  is a multiindex, and  $|\alpha| = \alpha_{\xi} + \alpha_{\eta} + \alpha_{\zeta}$  its order. Analogously, a one-to-one correspondence can be established between the scaled monomials of (2.4) and the indices  $\alpha \in \mathbb{N}$ , starting from index 1 corresponding to the constant monomial 1, and following each layer, read counterclockwise starting from the top, of *Pascal's pyramid*, whose first layers are depicted in Figure 2.3 and Figure 2.4. Given an integer  $k$ , the number of parameters  $\nu_k$  necessary to describe a polynomial in  $\mathcal{P}_k(P)$  is given by the number of total elements up to layer  $k$  of Pascal's pyramid, that is

$$\nu_k = \dim \mathcal{P}_k(P) = \frac{(k+1)(k+2)(k+3)}{6} \quad (2.5)$$

Therefore, once again, the scaled monomials of degree less or equal to  $k$  can be gathered in the  $[\nu_k \times 1]$  vector  $\boldsymbol{\mu}_k$

$$\boldsymbol{\mu}_k := \{1 \quad \xi \quad \eta \quad \zeta \quad \xi^2 \quad \xi\eta \quad \eta^2 \quad \eta\zeta \quad \zeta^2 \quad \xi\zeta \quad \xi^3 \quad \dots \quad \mu_k\}^T \quad (2.6)$$

whose elements form a basis for the polynomials  $\mathcal{P}_k(P)$  of degree less or equal to  $k$ .

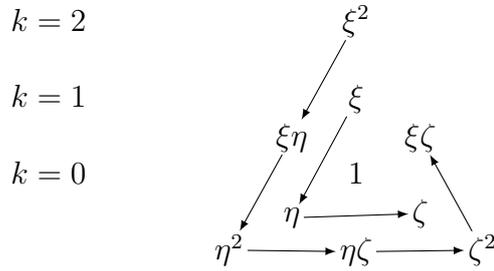


Figure 2.3: First three layers of Pascal's pyramid.

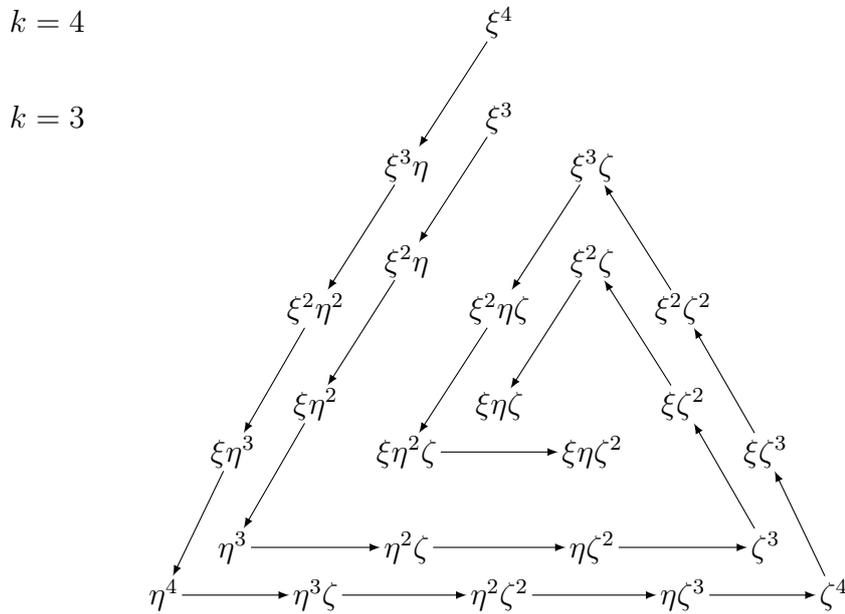


Figure 2.4: Layers corresponding to  $k = 3$  and  $k = 4$  of Pascal's pyramid.

Given a polynomial of degree  $k$  embedded in  $\mathbb{R}^3$ , it is possible to express its restriction on a plane through a polynomial of degree  $k$  embedded in  $\mathbb{R}^2$ . Specifically, given a scaled monomial  $\mu_k(\boldsymbol{\xi})$ , one can map its value through the scaled barycentric coordinates of the face  $\boldsymbol{\xi}_f$  according to the subsequent procedure, following Figure 2.5.

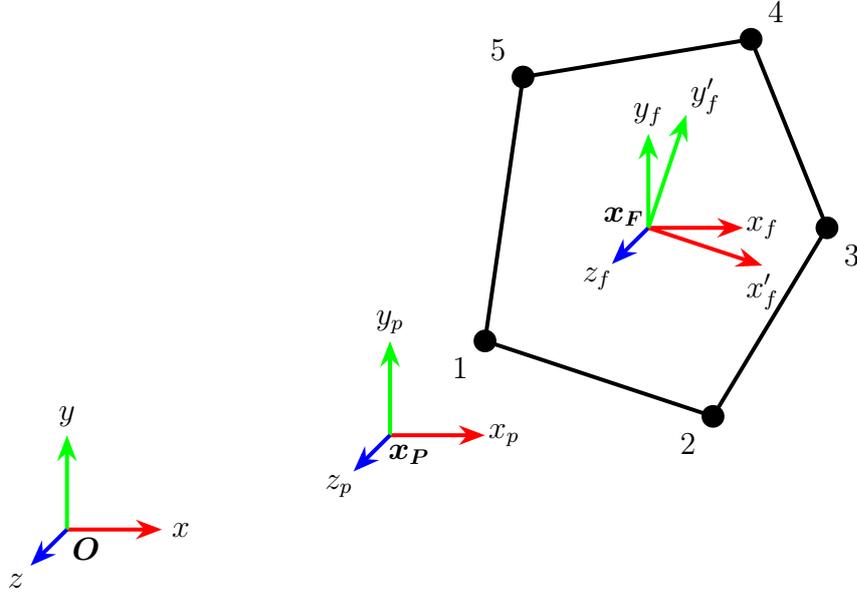


Figure 2.5: Coordinates transformation from 2D face barycentric scaled reference system to 3D scaled polyhedral reference system.

A polyhedron whose centroid lies in  $\mathbf{x}_P$  has a pentagonal face with centroid  $\mathbf{x}_F$ , depicted in Figure 2.5. The reference frames  $Oxyz$ ,  $x_P x_p y_p z_p$  and  $x_F x_f y_f z_f$  are obtained with a translation only, since no rotation nor scaling is applied. A rotation is applied from reference system  $x_F x_f y_f z_f$  to  $x_F x'_f y'_f z'_f$  so that  $z'_f$  axis is orthogonal to the polygon,  $x'_f$  axis is parallel to the first edge, following the nodes numbering, and the  $y'_f$  axis is obtained with the right-hand rule. A scaling by  $h_P$  is further performed in  $x_P x_p y_p z_p$  system, so to retrieve the scaled coordinates system  $x_P \xi \eta \zeta$  (simply indicated with  $x_P \xi \eta \zeta$ ) and by  $h_F$  in  $x_F x'_f y'_f z'_f$  so to obtain the scaled coordinates  $x_F \xi_f \eta_f \zeta_f$ . Hence, the following relations hold

$$\mathbf{x}_p = \mathbf{x} - \mathbf{x}_P \qquad \mathbf{x}_f = \mathbf{x} - \mathbf{x}_F \qquad (2.7)$$

$$\boldsymbol{\xi} = \frac{\mathbf{x}_p}{h_P} \qquad \boldsymbol{\xi}_f = \frac{\mathbf{x}'_f}{h_F} \qquad (2.8)$$

$$\mathbf{x}'_f = \mathbf{R} \mathbf{x}_f = \begin{bmatrix} (\mathbf{e}_{x'_f} \cdot \mathbf{e}_{x_f}) & (\mathbf{e}_{x'_f} \cdot \mathbf{e}_{y_f}) & (\mathbf{e}_{x'_f} \cdot \mathbf{e}_{z_f}) \\ (\mathbf{e}_{y'_f} \cdot \mathbf{e}_{x_f}) & (\mathbf{e}_{y'_f} \cdot \mathbf{e}_{y_f}) & (\mathbf{e}_{y'_f} \cdot \mathbf{e}_{z_f}) \\ (\mathbf{e}_{z'_f} \cdot \mathbf{e}_{x_f}) & (\mathbf{e}_{z'_f} \cdot \mathbf{e}_{y_f}) & (\mathbf{e}_{z'_f} \cdot \mathbf{e}_{z_f}) \end{bmatrix} \mathbf{x}_f \qquad (2.9)$$

where the orthogonal rotation matrix  $\mathbf{R}$  in (2.9) contains the cosines directors of the coordinates axes. Exploiting the above relations and the orthogonality of  $\mathbf{R}$ , one has

$$\boldsymbol{\xi} = \frac{\mathbf{x} - \mathbf{x}_P}{h_P} = \frac{\mathbf{x}_f + \mathbf{x}_F - \mathbf{x}_P}{h_P} = \frac{\mathbf{R}^{-1} \mathbf{x}'_f + \mathbf{x}_F - \mathbf{x}_P}{h_P} = \frac{h_F \mathbf{R}^T \boldsymbol{\xi}_f + \mathbf{x}_F - \mathbf{x}_P}{h_P} \qquad (2.10)$$

## 2.2. Local virtual element space in $\mathbb{R}^2$

In the following, focus is put on a single element  $P$  of the polytopic mesh  $\mathcal{P}_h$  defined in 2.1, and will be referred to as the *virtual element*. As in FE, the approximating functions are sought in a suitable space, namely the *virtual element space*. One of the major complexities of dealing with VE embedded in  $\mathbb{R}^3$  resides in the natural structure of the polyhedron, whose boundary is composed of polygons where a proper virtual element space embedded in  $\mathbb{R}^2$  should be given.

Therefore, in order to properly tackle the 3D problem, it is necessary to first understand the 2D setting and:

- define the local virtual element space in  $\mathbb{R}^2$
- find a uniquely-defying way of representing its elements (which will be the *degrees of freedom*).

**Definition 2.1.** *Local virtual element space embedded in  $\mathbb{R}^2$ . The local virtual element space  $V_k(F)$  of order  $k$ ,  $k \in \mathbb{N}, k \geq 1$ , for a polygon  $F$  is defined by functions  $v$  such that*

$$\left\{ \begin{array}{l} \bullet v \text{ is a polynomial of degree } k \text{ on each edge } E \text{ of the polygon } F, \text{ i.e. } v|_E \in \mathcal{P}_k(E) \\ \bullet v \text{ is globally continuous on } \partial F, \text{ i.e. } v|_{\partial F} \in C^0(\partial F) \\ \bullet \Delta v \text{ is a polynomial of degree } k - 2 \text{ in } F, \text{ i.e. } \Delta v \in \mathcal{P}_{k-2}(F) \end{array} \right.$$

In simple terms, the above definition includes in  $V_k(F)$  all polynomials of order  $k$  (as usually required for standard FE) plus some additional functions whose restriction on an edge is still a polynomial of order  $k$ . These *additional functions* are one of the key points of VEM, as their explicit expression is unknown and never required to be computed, remaining *virtual* throughout the whole process, so that they lent the name to the method itself. The third condition is enforced to fix the dimension of the space, as will be clarified later. Since a polygon of order  $k$  satisfies all the three requirements of Definition 2.1, the following inclusion holds

$$\mathcal{P}_k(F) \subset V_k(F) \tag{2.11}$$

which is essential for convergence properties. As already anticipated, the next step is to uniquely identify an element of  $V_k(F)$  through cleverly chosen parameters, the *local degrees of freedom* (DOFs) for the 2D virtual element space ([9]).

**Proposition 2.1.** *Local degrees of freedom for the virtual element space embedded in  $\mathbb{R}^2$ .*

*An element  $v$  of the space  $V_k(F)$  defined in Definition 2.1 is uniquely identified by the*

DOFs

$$\Xi : V_k(F) \rightarrow \mathbb{R}$$

grouped in the three following sets

$$\left\{ \begin{array}{l} \bullet \text{ the value of } v \text{ at the vertices of } F \\ \bullet \text{ for each edge } E \text{ of } F, \text{ the value of } v \text{ at the } k-1 \text{ internal points of the } (k+1)\text{-point} \\ \quad \text{Gauss-Lobatto quadrature rule on } E \\ \bullet \text{ the } n_{k-2} \text{ scaled moments up to order } k-2 \text{ of } v \text{ in } F: \\ \\ \frac{1}{|F|} \int_F v m_\alpha d\Sigma, \quad \alpha = 1, \dots, n_{k-2} \\ \\ \text{where } m_\alpha \text{ are the scaled monomials defined in (2.1) and } n_{k-2} \text{ in (2.2)} \end{array} \right. \quad (2.12)$$

*Proof.* To prove the unisolvence of the chosen degrees of freedom for the space  $V_k(F)$ , we split the argument in two steps. First we prove that if the value of the function  $v$  on the boundary  $\partial F$  and the polynomial of degree  $k-2$   $\Delta v$  are known, then the element  $v$  is uniquely identified. Let us consider the following problem

$$\left\{ \begin{array}{ll} \Delta v = \sum_{\alpha=1}^{n_{k-2}} m_\alpha = p_{k-2} & \text{in } F \\ v = g & \text{on } \partial F \end{array} \right. \quad (2.13)$$

where the polynomial of degree  $k-2$   $p_{k-2}$  is composed of the scaled monomials  $m_\alpha$  and  $g$  is the boundary datum, i.e., the value of  $v$  on the edges of the polygon. We claim that given suitable  $g$  and  $p_{k-2}$ , there exists a unique  $v$ . The corresponding weak form then reads

$$\begin{aligned} & \text{Given } p_{k-2} \in L^2(F), g \in H^{\frac{1}{2}}(\partial F), \text{ and } v_0 \in H^1(F) \text{ such that } v_0 = g \text{ on } \partial F, \\ & \text{find } v \in H^1(F) \text{ such that } v - v_0 \in H_0^1 \text{ and} \end{aligned} \quad (2.14)$$

$$\int_F \nabla v \nabla \phi d\Sigma = - \int_F p_{k-2} \phi d\Sigma \quad \forall \phi \in H_0^1(F)$$

Setting  $w = v - v_0$  so that  $w \in H_0^1(F)$ , problem 2.14 becomes

Given  $p_{k-2} \in L^2(F)$ ,  $g \in H^{\frac{1}{2}}(\partial F)$ , and  $v_0 \in H^1(F)$  such that  $v_0 = g$  on  $\partial F$ , find  $w \in H_0^1(F)$  such that

$$\int_F \nabla w \nabla \phi \, d\Sigma = - \int_F p_{k-2} \phi \, d\Sigma - \int_F \nabla v_0 \nabla \phi \, d\Sigma \quad \forall \phi \in H_0^1(F) \quad (2.15)$$

Now, the following hypotheses for problem 2.15 hold:

- Continuity of the bilinear form  $\int_F \nabla w \nabla \phi \, d\Sigma$

$$\int_F \nabla w \nabla \phi \, d\Sigma \leq \|\nabla w\|_{L^2(F)} \|\nabla \phi\|_{L^2(F)} \leq \|w\|_{H^1(F)} \|\phi\|_{H^1(F)} \quad (2.16)$$

where Cauchy-Schwartz inequality and the definition of the  $H^1$ -norm have been applied;

- Coercivity of the bilinear form  $\int_F \nabla w \nabla \phi \, d\Sigma$

$$\int_F \nabla w \nabla \phi \, d\Sigma = \|\nabla w\|_{L^2(F)}^2 \geq \frac{1}{1 + C_F^2} \|w\|_{H^1(F)}^2 = \alpha \|w\|_{H^1(F)}^2 \quad (2.17)$$

where  $H^1$ -norm definition and the below Poincaré inequality have been exploited, with  $C_F$  being a constant depending on the domain  $F$

$$\|w\|_{L^2(F)} \leq C_F \|\nabla w\|_{L^2(F)} \quad (2.18)$$

- Continuity of the linear functional  $-\int_F p_{k-2} \phi \, d\Sigma - \int_F \nabla v_0 \nabla \phi \, d\Sigma$

$$\begin{aligned} - \int_F p_{k-2} \phi \, d\Sigma - \int_F \nabla v_0 \nabla \phi \, d\Sigma &\leq \|p_{k-2}\|_{L^2(F)} \|\phi\|_{L^2(F)} + \|\nabla v_0\|_{L^2(F)} \|\nabla \phi\|_{L^2(F)} \leq \\ &\leq (\|p_{k-2}\|_{L^2(F)} + \|\nabla v_0\|_{L^2(F)}) \|\phi\|_{H^1(F)} \leq \\ &\leq C \|\phi\|_{H^1(F)} \end{aligned}$$

where Cauchy-Schwartz,  $H^1$ -norm definition and the embedding  $\mathcal{P}_{k-2}(F) \subset L^2(F)$  have been exploited;

In view of the above properties, Lax-Milgram lemma guarantees existence and uniqueness of a function  $w \in H_0^1(F)$ . However, the latter still depends on the choice of  $v_0$ , hence only existence for  $v \in H^1(F)$  is guaranteed. To prove uniqueness, we argue by contradiction assuming there exist two solutions  $v_1$  and  $v_2$  to Problem 2.14, both coinciding with  $g$  on

the boundary. Taking the difference of the resulting weak forms, the following is obtained

$$\int_F \nabla (v_1 - v_2) \nabla \phi \, d\Sigma = 0 \quad \forall \phi \in H_0^1 \quad (2.19)$$

and testing with the particular choice of  $(v_1 - v_2)$  for  $\phi$  leads to

$$0 = \int_F \nabla (v_1 - v_2) \nabla (v_1 - v_2) \, d\Sigma = \|\nabla (v_1 - v_2)\|_{L^2(F)}^2 \geq \|v_1 - v_2\|_{L^2(F)}^2 \geq 0 \quad (2.20)$$

where Poincaré inequality has been applied. Equation (2.20) implies that  $v_1 = v_2$ , and uniqueness for Problem 2.14 is also proved.

The second step of the proof consists in checking that the selected degrees of freedom of Proposition 2.1 uniquely identify the data of Problem 2.13. To this aim, we note that the first two sets of DOFs, corresponding to the value of  $v$  at the vertices  $V$  of the polygon  $F$  and at the  $k - 1$  internal points of each edge  $E$ , uniquely define the boundary datum  $g$ . In fact, for each edge, the polynomial of degree  $k$  prescribed by Definition 2.1 is described exactly by the  $k + 1$  points, vertices included, belonging to the edge. It only remains to check that if the internal DOFs are 0, together with the boundary datum, then the function  $v$  is identically null. Substituting  $p_{k-2}$  with the sum of the scaled monomials  $m_\alpha$  in Problem 2.14, one obtains

$$\begin{aligned} \int_F \nabla v \nabla \phi \, d\Sigma &= - \int_F \left( \sum_{\alpha=1}^{n_{k-2}} m_\alpha \right) \phi \, d\Sigma = \\ &= - \sum_{\alpha=1}^{n_{k-2}} \left( \int_F m_\alpha \phi \, d\Sigma \right) \quad \forall \phi \in H_0^1(F) \end{aligned} \quad (2.21)$$

Being the data  $g = 0$ , we can take  $v \in H_0^1(F)$ , and testing with the particular  $v$  in place of  $\phi$ , the following holds

$$0 \leq \|v\|_{L^2(F)}^2 \leq C_F^2 \|\nabla v\|_{L^2(F)}^2 = - \sum_{\alpha=1}^{n_{k-2}} \left( \int_F m_\alpha \phi \, d\Sigma \right) \quad (2.22)$$

where Poincaré inequality has once again been exploited, justified by  $v \in H_0^1(F)$ . The quantities in brackets are exactly the internal moments: if these are zero, then the  $L^2$ -norm of  $v$  is bounded from below and above by 0, and therefore  $v$  is identically null, completing the proof.  $\square$

As a consequence of the above proposition, the dimension of the space  $V_k(F)$  coincides

with the number of corresponding degrees of freedom  $N_{DOF}$

$$\dim V_k(F) = N_{DOF} = N_V + N_V(k-1) + n_{k-2} = kN_V + \frac{(k-1)k}{2} \quad (2.23)$$

where  $N_V$  is the number of vertices (equal to the number of edges) belonging to the polygon  $F$ . The space  $V_k(F)$  can be decomposed into its  $N_{DOF}$  *basis functions* (or *shape functions*), here denoted with  $\varphi$

$$\Xi_i(\varphi_j) = \delta_{ij} \quad \forall i, j = 1, \dots, N_{DOF} \quad (2.24)$$

so that any element of  $v$  can be expressed through the Lagrangian interpolation

$$v = \sum_{i=1}^{N_{DOF}} \Xi_i(v) \varphi_i \quad \forall v \in V_k(F) \quad (2.25)$$

A visual representation of the DOFs described in Proposition 2.1 is depicted in Figure 2.6 for the first three virtual element spaces corresponding to  $k = 1$ ,  $k = 2$  and  $k = 3$ . To better understand the structure of the virtual shape functions  $\varphi$ , one can go against the philosophy of the VEM and find them explicitly by solving the corresponding PDE (e.g., through finite elements) with boundary conditions given by the polynomial restriction described by the vertex-type and edge-type DOFs. For higher order VEM ( $k \geq 2$ ), this task additionally requires an optimization process since the shape functions are no longer harmonic<sup>1</sup> and the polynomial of the Laplacian is not known a priori, but can be found so that the moment-type DOFs are either 0 or 1. In Figures 2.7, 2.8 and 2.9 some virtual shape functions are computed and rendered.

It is important to remark that the choice for the DOFs made in Proposition 2.1 is not the only possible one in order to achieve unique identification of an element  $v$  of the space  $V_k(F)$ . One could take, for instance,  $k-1$  non-coinciding randomly-picked internal points on each edge  $E$  and still achieve the same results. However, as will be clarified later, the sets of local DOFs have been specifically selected to ease computations and perform the steps with the minimum amount of information required to still guarantee convergence.

---

<sup>1</sup>The Laplacian of a virtual shape function  $\varphi$  of degree  $k$  embedded in  $\mathbb{R}^2$  is a polynomial of degree  $k-2$ , hence it is 0 if  $k = 1$ .

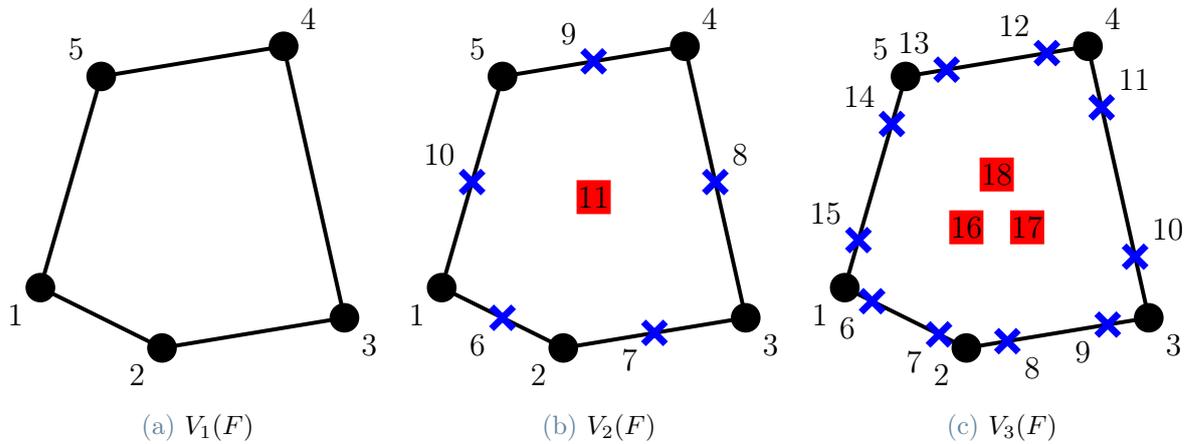


Figure 2.6: Local degrees of freedom for the three virtual element spaces  $V_1(F)$ ,  $V_2(F)$  and  $V_3(F)$  in a pentagon  $F$ . The black dots correspond to vertices DOFs, blue crosses to edge DOFs, and red squares to internal DOFs. Note that while the first two sets match with the function evaluation at the precise location shown in the figure, the latter do not have a geometric punctual representation and are displayed inside the polygon for the sake of simplicity.

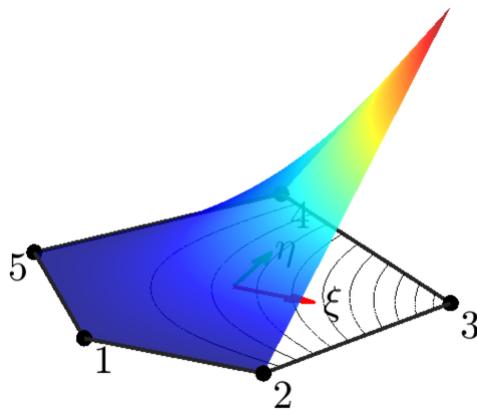


Figure 2.7: Virtual shape function  $\varphi_3$  corresponding to the vertex  $V_3$  for a first order pentagonal virtual element. The restriction on the boundary is linear and nonzero on the edges  $\overline{V_2V_3}$  and  $\overline{V_3V_4}$ . Note that the shape function is harmonic ( $\Delta\varphi_3 = 0$ ).

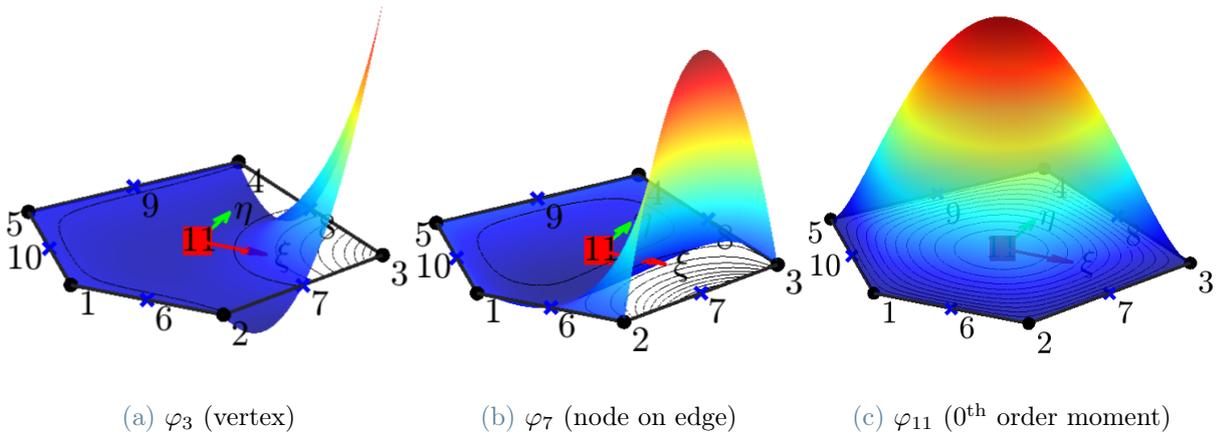


Figure 2.8: Virtual shape functions  $\varphi_3$ ,  $\varphi_7$  and  $\varphi_{11}$  corresponding to the vertex  $V_3$ , the edge node 7 and the internal  $0^{\text{th}}$  order moment respectively for a second order pentagonal virtual element. The restriction on the boundary is a piecewise quadratic polynomial. Note that for each shape function, its Laplacian is a constant and in 2.8a and 2.8b the integral of the shape function over the pentagon is null.

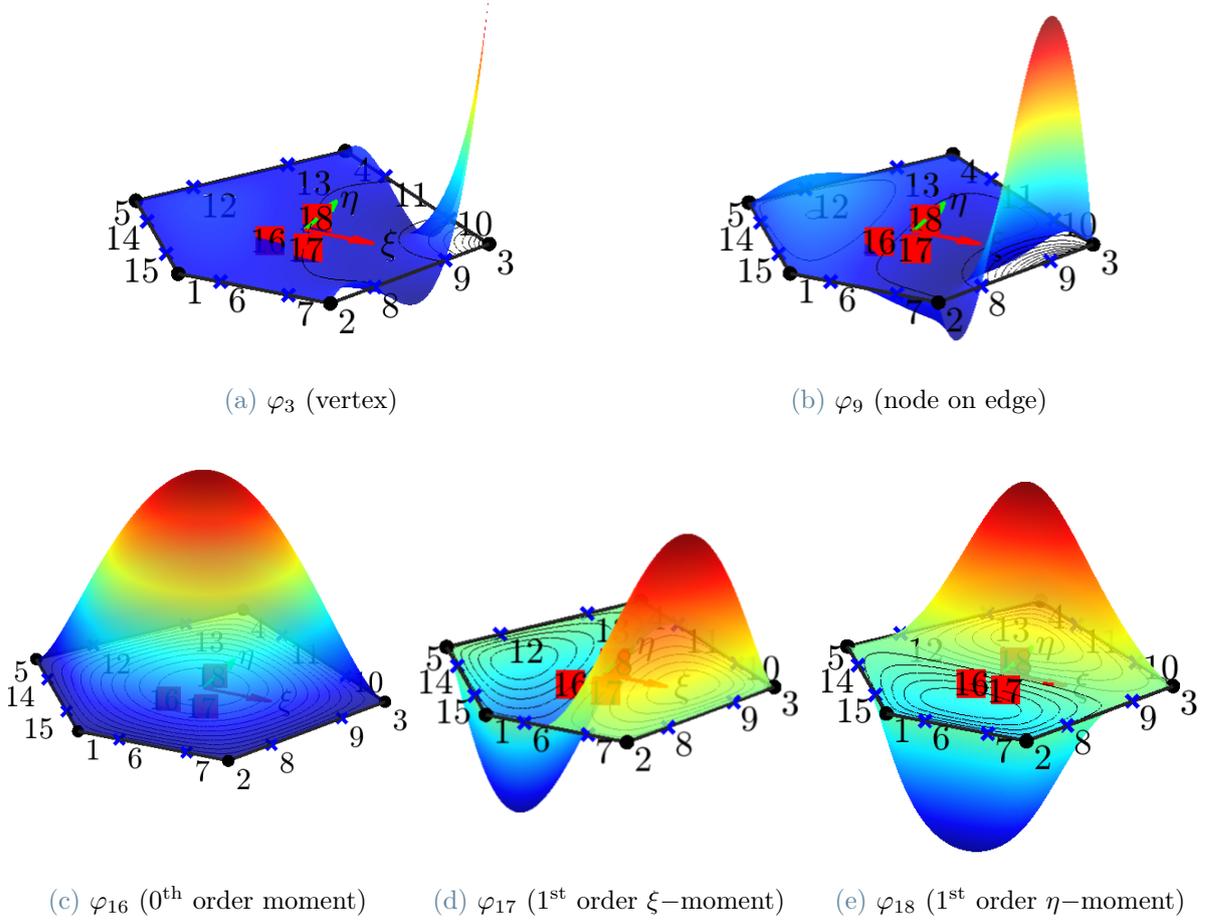


Figure 2.9: Virtual shape functions  $\varphi_3$ ,  $\varphi_9$ ,  $\varphi_{16}$ ,  $\varphi_{17}$  and  $\varphi_{18}$  corresponding to the vertex  $V_3$ , the edge node 9, the internal 0<sup>th</sup> order and the two first order moments respectively, for a 3<sup>rd</sup> order pentagonal virtual element. The restriction on the boundary is a piecewise cubic polynomial. Note that in (a) and (b) the moments up to first order over the pentagon are null, in (c) the first order moments are null, in (d) the 0<sup>th</sup> order and first order moment w.r.t.  $\eta$  are null while in (e) the 0<sup>th</sup> order and first order moment w.r.t.  $\xi$  are null.

### 2.3. Projection operator

Given the unknown structure of the functions  $v$  of the space  $V_k(F)$  described in 2.2, whose explicit form is in general known only after solving a boundary value problem, a projection operator has been introduced in the VEM literature. In this specific work, the projector will be explicitly used for the faces only, since for projecting operation applied for functions defined in polyhedra will already be incorporated in the definition of the strain field.

**Definition 2.2.** *Projection operator  $\Pi_{F,k}^\nabla$ . The projection operator  $\Pi_{F,k}^\nabla$  maps an element of  $v \in V_k(F)$  into a polynomial of order  $k$*

$$\Pi_{F,k}^\nabla : V_k(F) \rightarrow \mathcal{P}_k(F)$$

and it is defined by the orthogonality condition

$$\int_F \nabla p_k \cdot \nabla (\Pi_{F,k}^\nabla v - v) \, d\Sigma = 0 \quad \forall p_k \in \mathcal{P}_k(F) \quad (2.26)$$

and the conditions to fix the constants

$$\begin{cases} \frac{1}{N_V} \sum_{j=1}^{N_V} \{ [\Pi_{F,k}^\nabla v] (\boldsymbol{\xi}_{V_j}) - v(\boldsymbol{\xi}_{V_j}) \} = 0 & \text{if } k = 1 \\ \frac{1}{|F|} \int_F (\Pi_{F,k}^\nabla v - v) \, d\Sigma = 0 & \text{if } k \geq 2 \end{cases} \quad (2.27)$$

where  $\boldsymbol{\xi}_{V_j}$  are the coordinates of the vertex  $V_j$ .

To extract the polynomial projection of a virtual function, the following procedure is applied. We focus the attention on a virtual basis function  $\varphi_i$  instead of a virtual function  $v$ , in view of (2.24) and (2.25). From Definition 2.2, since  $\varphi_i \in V_k(F)$  and  $m_\alpha \in \mathcal{P}_k(F)$  we have

$$\int_F \nabla m_\alpha \cdot \nabla (\Pi_{F,k}^\nabla \varphi_i - \varphi_i) \, d\Sigma = 0 \quad \forall \alpha \leq n_k, \quad \forall i = 1, \dots, \dim V_k(F) \quad (2.28)$$

Since  $\Pi_{F,k}^\nabla \varphi_i \in \mathcal{P}_k(F)$ , we have

$$\Pi_{F,k}^\nabla \varphi_i = \sum_{\beta=1}^{n_k} s_i^\beta m_\beta \quad \forall i = 1, \dots, \dim V_k(F)$$

which plugged into (2.28), after splitting the integral, leads to

$$\sum_{\beta=1}^{n_k} s_i^\beta \int_F \nabla m_\alpha \cdot \nabla m_\beta \, d\Sigma = \int_F \nabla m_\alpha \cdot \nabla \varphi_i \, d\Sigma \quad \forall \alpha \leq n_k, \quad \forall i = 1, \dots, \dim V_k(F) \quad (2.29)$$

The right hand side of (2.29) can be integrated by parts, yielding to

$$\begin{aligned} \int_F \nabla m_\alpha \cdot \nabla \varphi_i d\Sigma &= \int_{\partial F} \nabla m_\alpha \cdot \mathbf{n} \varphi_i d\Gamma - \int_F \Delta m_\alpha \varphi_i d\Sigma = \\ &= \sum_{E \in \partial F} \left[ \int_E \nabla m_\alpha \cdot \mathbf{n} \varphi_i d\Gamma \right] - \int_F \Delta m_\alpha \varphi_i d\Sigma \quad \forall \alpha \leq n_k, \\ &\quad \forall i = 1, \dots, \dim V_k(F) \end{aligned}$$

where  $\mathbf{n}$  is the outward unit vector of the boundary  $\partial F$  of the polygon  $F$ . The  $n_k$  linear equations in the  $n_k$  unknowns of (2.29) can be gathered in an algebraic system. However, the first equation coming from  $\alpha = 1$  is the trivial identity  $0 \equiv 0$  and is therefore replaced with the equation corresponding to the correct order in (2.27), which read

$$\begin{cases} \sum_{\beta=1}^{n_k} s_i^\beta \left[ \frac{1}{N_V} \sum_{j=1}^{N_V} m_\beta(\boldsymbol{\xi}_{V_j}) \right] = \frac{1}{N_V} \sum_{j=1}^{N_V} \varphi_i(\boldsymbol{\xi}_{V_j}) & \text{if } k = 1 \\ \sum_{\beta=1}^{n_k} s_i^\beta \left[ \frac{1}{|F|} \int_F m_\beta d\Sigma \right] = \frac{1}{|F|} \int_F \varphi_i d\Sigma & \text{if } k \geq 2 \end{cases} \quad (2.30)$$

The linear algebraic system in the  $n_k$  unknowns  $s_i^\beta$  then reads

$$\sum_{\beta=1}^{n_k} [G_F]_{\alpha\beta} s_i^\beta = b_{F,i}^\alpha \quad \forall \alpha = 1, \dots, n_k \quad (2.31)$$

where the  $\alpha^{\text{th}}$  row and  $\beta^{\text{th}}$  column entries of  $[n_k \times n_k]$  matrix  $\mathbf{G}_F$  are

$$\begin{cases} \frac{1}{N_V} \sum_{j=1}^{N_V} m_\beta(\boldsymbol{\xi}_{V_j}) & \text{if } \alpha = 1 \text{ and } k = 1 \\ \frac{1}{|F|} \int_F m_\beta d\Sigma & \text{if } \alpha = 1 \text{ and } k \geq 2 \\ \int_F \nabla m_\alpha \cdot \nabla m_\beta d\Sigma & \text{if } \alpha > 1 \end{cases} \quad (2.32)$$

and the  $\alpha^{\text{th}}$  component of  $[n_k \times 1]$  vector  $\mathbf{b}_{F,i}$  is

$$\begin{cases} \frac{1}{N_V} \sum_{j=1}^{N_V} \varphi_i(\boldsymbol{\xi}_{V_j}) & \text{if } \alpha = 1 \text{ and } k = 1 \\ \frac{1}{|F|} \int_F \varphi_i d\Sigma & \text{if } \alpha = 1 \text{ and } k \geq 2 \\ \sum_{E \in \partial F} \left[ \int_E \nabla m_\alpha \cdot \mathbf{n} \varphi_i d\Gamma \right] - \int_F \Delta m_\alpha \varphi_i d\Sigma & \text{if } \alpha > 1 \end{cases} \quad (2.33)$$

The quantities of matrix  $\mathbf{G}_F$  in (2.32) are all computable as they involve integrations of polynomials over  $F$  or simple function evaluations. The quantities of the vector  $\mathbf{b}_{F,i}$  are also computable as they involve, from top to bottom, evaluations of a basis function in vertices DOFs, evaluation of a basis function in a face DOF, integral of a polynomial times the basis function on the edge and evaluation of a basis function in a face DOF. To fix the ideas, the procedure applied to produce the algebraic systems to be solved to compute the coefficients of the projections  $\Pi_{F,1}^\nabla \varphi_i$  and  $\Pi_{F,2}^\nabla \varphi_i$  are explicitly reported below.

- $k = 1$

The  $n_1 = 3$  scaled monomials, their gradients and laplacians are

$$\begin{array}{ccc} m_1 = 1 & m_2 = \xi & m_3 = \eta \\ \nabla m_1 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} & \nabla m_2 = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} & \nabla m_3 = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \\ \Delta m_1 = 0 & \Delta m_2 = 0 & \Delta m_3 = 0 \end{array}$$

Lagragian-type interpolation implies

$$\varphi_i(\boldsymbol{\xi}_{V_j}) = \delta_{ij} \implies \sum_{j=1}^{N_V} \varphi_i(\boldsymbol{\xi}_{V_j}) = 1 \quad \forall i = 1, \dots, \dim V_1(F) \equiv N_V$$

Since the shape function is a polynomial of degree 1 on the edges, we can apply 2-point Gauss-Lobatto quadrature with the two vertices of the edge

$$\begin{cases} \varphi_i(\boldsymbol{\xi}_{V_j}) = \delta_{ij} \\ \varphi_i|_E \in \mathcal{P}_1(E) \end{cases} \implies \begin{cases} \int_E \nabla m_2 \cdot \mathbf{n} \varphi_i d\Gamma = \begin{cases} \frac{1}{2}|E|n_{E,\xi} & \text{if } V_i \in E \\ 0 & \text{if } V_i \notin E \end{cases} \\ \int_E \nabla m_3 \cdot \mathbf{n} \varphi_i d\Gamma = \begin{cases} \frac{1}{2}|E|n_{E,\xi} & \text{if } V_i \in E \\ 0 & \text{if } V_i \notin E \end{cases} \end{cases}$$

where  $\mathbf{n}_E = \{n_{E,\xi} \ n_{E,\eta}\}^T$  is the outward normal vector of  $\partial F$  in correspondence of edge  $E$ . Indicating the two edges shared by vertex  $i$  with  $E_+$  and  $E_-$ , the final system solving for the polynomial coefficients  $\mathbf{s}_i$  of  $\Pi_{F,1}^\nabla \varphi_i$  is obtained

$$\begin{bmatrix} 1 & \frac{1}{N_V} \sum_{j=1}^{N_V} \xi_{V_j} & \frac{1}{N_V} \sum_{j=1}^{N_V} \eta_{V_j} \\ 0 & \int_F 1 & 0 \\ 0 & 0 & \int_F 1 \end{bmatrix} \begin{Bmatrix} s_i^1 \\ s_i^2 \\ s_i^3 \end{Bmatrix} = \begin{Bmatrix} \frac{1}{N_V} \\ \frac{1}{2}(|E_+|n_{E_+,\xi} + |E_-|n_{E_-,\xi}) \\ \frac{1}{2}(|E_+|n_{E_+,\eta} + |E_-|n_{E_-,\eta}) \end{Bmatrix} \quad (2.34)$$

To help visualize what has been done above, Figure 2.10 renders the projection  $\Pi_{F,1}^\nabla$  applied to a shape function in a quadrilateral domain.

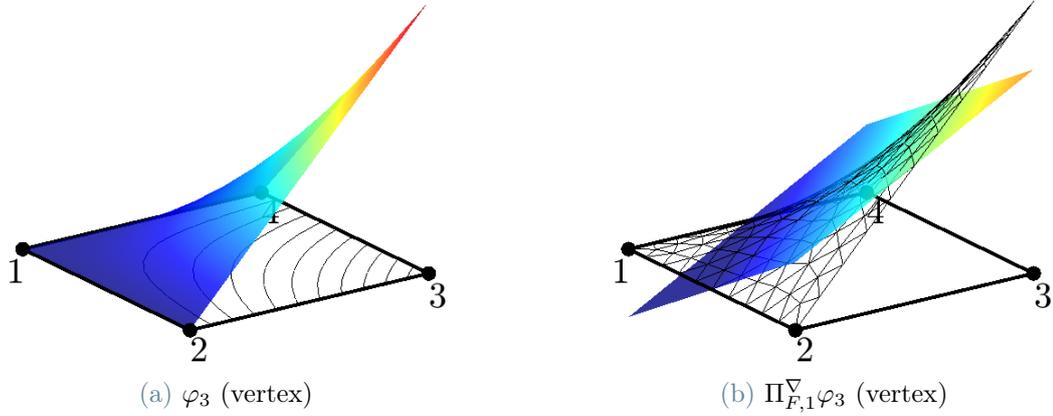


Figure 2.10: Projection  $\Pi_{F,1}^\nabla \varphi_3$  (b) of the shape function  $\varphi_3$  (a) for a linear virtual element. Note how Lagrangian-type interpolation holds for the shape function in (a) and not for the projection in (b) (the value of  $\Pi_{F,1}^\nabla \varphi_3$  at  $V_3$  is no longer 1 and it is not 0 in all the other nodes).

- $k=2$

The  $n_2 = 6$  scaled monomials, their gradients and laplacians are

$$\begin{aligned} m_1 &= 1 & m_2 &= \xi & m_3 &= \eta & m_4 &= \xi^2 & m_5 &= \xi\eta & m_6 &= \eta^2 \\ \nabla m_1 &= \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} & \nabla m_2 &= \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} & \nabla m_3 &= \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} & \nabla m_4 &= \begin{Bmatrix} 2\xi \\ 0 \end{Bmatrix} & \nabla m_5 &= \begin{Bmatrix} \eta \\ \xi \end{Bmatrix} & \nabla m_6 &= \begin{Bmatrix} 0 \\ 2\eta \end{Bmatrix} \\ \Delta m_1 &= 0 & \Delta m_2 &= 0 & \Delta m_3 &= 0 & \Delta m_4 &= 2 & \Delta m_5 &= 0 & \Delta m_6 &= 2 \end{aligned}$$

Lagrangian-type interpolation implies

$$\frac{1}{|F|} \int_F \varphi_i d\Sigma = \delta_{i, \dim V_2(F)} = \delta_{i, 2N_V+1}$$

as there is only one face moment DOF. Since the shape function is a polynomial of degree 2 on the edges and the gradients of the scaled monomials are at most of degree 1, we can apply 3-point Gauss-Lobatto quadrature with the two vertices of the edge and its midpoint

$$\left\{ \begin{array}{l} \varphi_i(\boldsymbol{\xi}_{V_j}) = \delta_{ij} \\ \varphi_i|_E \in \mathcal{P}_2(E) \\ m_\alpha|_E \in \mathcal{P}_1(E) \end{array} \right. \implies \left\{ \begin{array}{l} \int_E \nabla m_2 \cdot \mathbf{n} \varphi_i d\Gamma = \begin{cases} \frac{1}{6}|E|n_{E,\xi} & \text{if } V_i \in E \\ \frac{2}{3}|E|n_{E,\xi} & \text{if } E_i \in E \\ 0 & \text{otherwise} \end{cases} \\ \int_E \nabla m_3 \cdot \mathbf{n} \varphi_i d\Gamma = \begin{cases} \frac{1}{6}|E|n_{E,\eta} & \text{if } V_i \in E \\ \frac{2}{3}|E|n_{E,\eta} & \text{if } E_i \in E \\ 0 & \text{otherwise} \end{cases} \\ \int_E \nabla m_4 \cdot \mathbf{n} \varphi_i d\Gamma = \begin{cases} \frac{1}{6}|E|2\xi_{V_i}n_{E,\xi} & \text{if } V_i \in E \\ \frac{2}{3}|E|2\xi_{E_i}n_{E,\xi} & \text{if } E_i \in E \\ 0 & \text{otherwise} \end{cases} \\ \int_E \nabla m_5 \cdot \mathbf{n} \varphi_i d\Gamma = \begin{cases} \frac{1}{6}|E|(\eta_{V_i}n_{E,\xi} + \xi_{V_i}n_{E,\eta}) & \text{if } V_i \in E \\ \frac{2}{3}|E|(\eta_{E_i}n_{E,\xi} + \xi_{E_i}n_{E,\eta}) & \text{if } E_i \in E \\ 0 & \text{otherwise} \end{cases} \\ \int_E \nabla m_6 \cdot \mathbf{n} \varphi_i d\Gamma = \begin{cases} \frac{1}{6}|E|2\eta_{V_i}n_{E,\eta} & \text{if } V_i \in E \\ \frac{2}{3}|E|2\xi_{V_i}n_{E,\eta} & \text{if } E_i \in E \\ 0 & \text{otherwise} \end{cases} \end{array} \right.$$

where  $\mathbf{n}_E = \{n_{E,\xi} \quad n_{E,\eta}\}^\top$  is the outward normal vector of  $\partial F$  in correspondence of edge  $E$ ,  $V_i$  is the vertex where the vertex shape function  $\varphi_i$  assumes value 1 and  $E_i$  is the midpoint where the edge shape function  $\varphi_i$  assumes value 1. Indicating as before the two edges shared by vertex  $i$  with  $E_+$  and  $E_-$ , the final system solving for the polynomial coefficients  $\mathbf{s}_i$  of  $\Pi_{F,2}^\nabla \varphi_i$  is obtained (the differential symbol  $d\Sigma$

is omitted)

$$\begin{aligned}
& \begin{bmatrix} 1 & \frac{1}{|F|} \int_F \xi & \frac{1}{|F|} \int_F \eta & \frac{1}{|F|} \int_F \xi^2 & \frac{1}{|F|} \int_F \xi \eta & \frac{1}{|F|} \int_F \eta^2 \\ 0 & \int_F 1 & 0 & \int_F 2\xi & \int_F \eta & 0 \\ 0 & 0 & \int_F 1 & 0 & \int_F \xi & \int_F 2\eta \\ 0 & \int_F 2\xi & 0 & \int_F 4\xi^2 & \int_F 2\xi\eta & 0 \\ 0 & \int_F \eta & \int_F \xi & \int_F 2\xi\eta & \int_F \xi^2 + \eta^2 & \int_F 2\xi\eta \\ 0 & 0 & \int_F 2\eta & 0 & \int_F 2\xi\eta & \int_F 4\eta^2 \end{bmatrix} \begin{Bmatrix} s_i^1 \\ s_i^2 \\ s_i^3 \\ s_i^4 \\ s_i^5 \\ s_i^6 \end{Bmatrix} = \\
& = \begin{cases} \begin{Bmatrix} 0 \\ \frac{1}{6}(|E_+|n_{E_+,\xi} + |E_-|n_{E_-,\xi}) \\ \frac{1}{6}(|E_+|n_{E_+,\eta} + |E_-|n_{E_-,\eta}) \\ \frac{1}{6}2\xi_{V_i}(|E_+|n_{E_+,\xi} + |E_-|n_{E_-,\xi}) \\ \frac{1}{6} \left[ \begin{array}{l} \eta_{V_i} (|E_+|n_{E_+,\xi} + |E_-|n_{E_-,\xi}) + \\ + \xi_{V_i} (|E_+|n_{E_+,\eta} + |E_-|n_{E_-,\eta}) \end{array} \right] \\ \frac{1}{6}2\eta_{V_i}(|E_+|n_{E_+,\eta} + |E_-|n_{E_-,\eta}) \end{Bmatrix} & \text{if } i \leq N_V \\ \\ \begin{Bmatrix} 0 \\ \frac{2}{3}|E|n_{E,\xi} \\ \frac{2}{3}|E|n_{E,\eta} \\ \frac{2}{3}2\xi_{E_i}|E|n_{E,\xi} \\ \frac{2}{3}(\eta_{E_i}|E|n_{E,\xi} + \xi_{E_i}|E|n_{E,\eta}) \\ \frac{2}{3}2\eta_{E_i}|E|n_{E,\eta} \end{Bmatrix} & \text{if } N_V < i \leq 2N_V \\ \\ \begin{Bmatrix} 1 \\ 0 \\ 0 \\ -2|F| \\ 0 \\ -2|F| \end{Bmatrix} & \text{if } i = 2N_V + 1 \end{cases} \quad (2.35)
\end{aligned}$$

The above right hand side entries depend on whether the shape function  $\varphi_i$  corresponds to a vertex DOF, edge DOF, or internal moment face DOF.

Figure 2.11 shows the projection  $\Pi_{F,2}^\nabla$  applied to three shape functions in a quadrilateral domain.

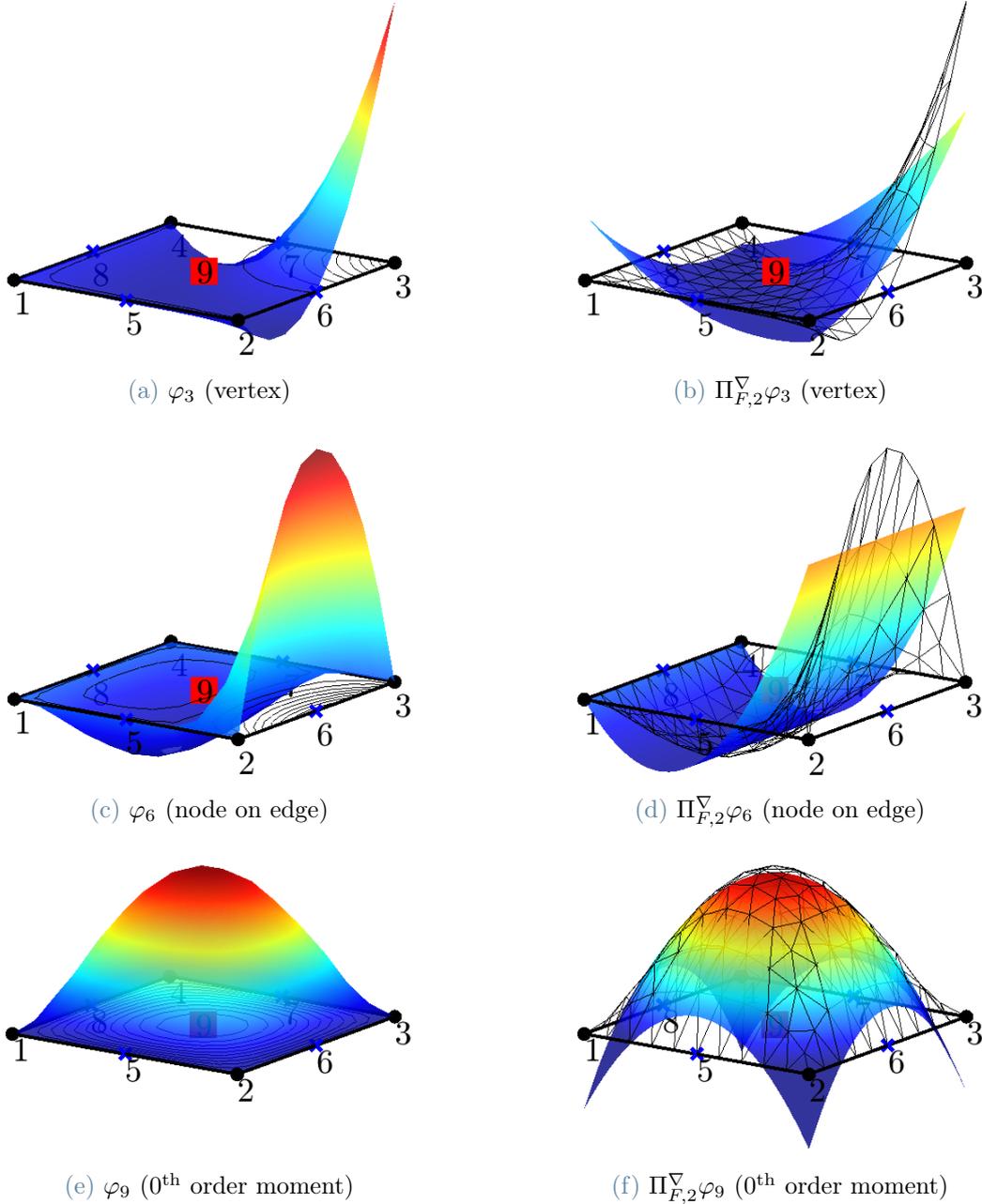


Figure 2.11: Projections  $\Pi_{F,2}^{\nabla} \varphi_3$  (b),  $\Pi_{F,2}^{\nabla} \varphi_6$  (d) and  $\Pi_{F,2}^{\nabla} \varphi_9$  (f) of the respective shape functions  $\varphi_3$  (a),  $\varphi_6$  (c) and  $\varphi_9$  (e) for a quadratic virtual quadrilateral element. Note how Lagrangian-type interpolation holds for the shape functions in (a), (c) and (e) and not for their projections in (b), (d) and (f) (e.g. the value of  $\Pi_{F,2}^{\nabla} \varphi_3$  at  $V_3$  is no longer 1 and it is not 0 in all the other nodes). However, the projection of every shape function for  $k \geq 2$  preserves the Lagrangian-type interpolation property with respect to the internal  $0^{\text{th}}$  order moment DOF, following the second condition in (2.27), so that the integral under the surfaces in (b) and (d) is 0 and in (f) coincides with the area of the domain  $|F|$ .

## 2.4. Enhanced local virtual element space in $\mathbb{R}^2$

In this section the enhanced local virtual element space  $W_k(F)$  embedded in  $\mathbb{R}^2$  is presented. The arguments on the existence of this space and the proofs of its properties are shown in [1]. The enhanced local virtual element space  $W_k(F)$  is specifically built from the local space  $V_k(F)$  defined in Definition 2.1, so that

- $w \in W_k(F)$  is still a polynomial of degree  $k$  on each edge  $E$  of the polygon  $F$
- $\mathcal{P}_k(F) \subset W_k(F)$
- the DOFs described in Proposition 2.1 for  $V_k(F)$  can still be used for the space  $W_k(F)$

and suitably modified to enjoy the property of the following

**Theorem 2.1.** *Enhanced virtual element space property. Given  $w \in W_k(F)$ , where  $W_k(F)$  is the enhanced local virtual element space embedded in  $\mathbb{R}^2$ , the following holds*

$$\int_F w m_\alpha d\Sigma = \int_F \Pi_{F,k}^\nabla w m_\alpha d\Sigma \quad |\alpha| = k - 1, k \quad (2.36)$$

As will be clear in the following chapter, the property given in 2.1 is essential to implement the virtual element method in 3D and to tackle dynamic problems where mass matrices appear. A precise definition of the space  $W_k(F)$  is reported here for completeness.

**Definition 2.3.** *Local enhanced virtual element space embedded in  $\mathbb{R}^2$ . The local virtual element space  $W_k(F)$  of order  $k$ ,  $k \in \mathbb{N}, k \geq 1$ , for a polygon  $F$  is defined by functions  $w$  such that*

$$\left\{ \begin{array}{l} \bullet w \text{ is a polynomial of degree } k \text{ on each edge } E \text{ of the polygon } F, \text{ i.e. } w|_E \in \mathcal{P}_k(E) \\ \bullet w \text{ is globally continuous on } \partial F, \text{ i.e. } w|_{\partial F} \in C^0(\partial F) \\ \bullet \Delta w \text{ is a polynomial of degree } k \text{ in } F, \text{ i.e. } \Delta w \in \mathcal{P}_k(F) \\ \bullet \text{ the enhanced property holds} \end{array} \right. \quad \int_F w m_\alpha d\Sigma = \int_F [\Pi_{F,k}^\nabla w] m_\alpha d\Sigma \quad |\alpha| = k - 1, k$$

It should be noted that the enhanced space  $W_k(F)$  has the same dimension of  $V_k(F)$ , as the  $(n_k - n_{k-2})$  additional degrees of freedom introduced by the more general Laplacian  $\Delta w$  - being a polynomial of degree  $k$  and not  $(k - 2)$  - are constrained by the supplementary  $(n_k - n_{k-2})$  conditions prescribed by the enhanced property. Moreover, since the two

spaces share the same degrees of freedom and these are uniquely identifying the projection defined in 2.2, two elements  $v \in V_k(F)$  and  $w \in W_k(F)$  having the same values for the DOFs have the exact same projection  $\Pi_{F,k}^\nabla$ , even though  $v$  and  $w$  do not coincide in general (Figure 2.12).

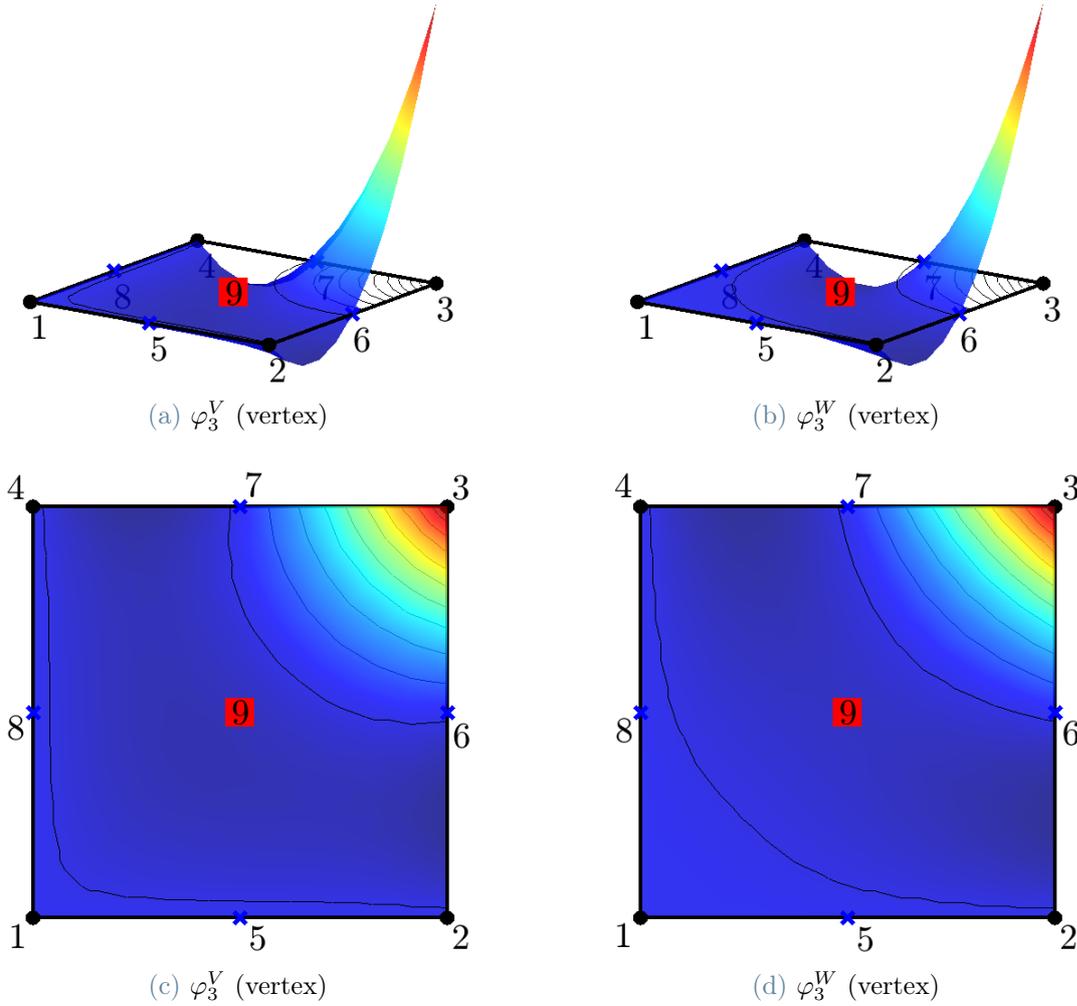


Figure 2.12: Virtual shape function  $\varphi_3^V \in V_2(F)$  (a, c) and corresponding virtual enhanced shape function  $\varphi_3^W \in W_2(F)$  (b, d) for the vertex  $V_3$  in a quadrilateral domain  $F$ . The restriction of both functions on the boundary is a piecewise quadratic polynomial. The Laplacian of  $\varphi_3^V$  is a constant, so that the 0<sup>th</sup> order moment of  $\varphi_3^V$  is null, while the Laplacian of  $\varphi_3^W$  is a 2<sup>nd</sup> order polynomial, so that the 0<sup>th</sup> order moment of  $\varphi_3^W$  is still null and additionally the 1<sup>st</sup> and 2<sup>nd</sup> order moments coincide with those of the polynomial projection  $\Pi_{F,2}^\nabla \varphi_3^W$ . Moreover, the polynomial projections of the two shape functions coincide ( $\Pi_{F,2}^\nabla \varphi_3^V = \Pi_{F,2}^\nabla \varphi_3^W$ , see Figure 2.11b).

## 2.5. Local virtual element space in $\mathbb{R}^3$

Having in mind the local virtual element space in 2D, the projector operator, and the enhanced space, it is now possible to extend the setting to three dimensions.

**Definition 2.4.** *Local virtual element space embedded in  $\mathbb{R}^3$ . The local virtual element space  $V_k(P)$  of order  $k$ ,  $k \in \mathbb{N}, k \geq 1$ , for a polyhedron  $P$  is defined by functions  $v$  such that*

- $$\left\{ \begin{array}{l} \bullet v \text{ is a polynomial of degree } k \text{ on each edge } E \text{ of the polyhedron } P, \text{ i.e. } v|_E \in \mathcal{P}_k(E) \\ \bullet v \text{ is globally continuous on } \partial P, \text{ i.e. } v|_{\partial P} \in C^0(\partial P) \\ \bullet \Delta v \text{ is a polynomial of degree } k - 2 \text{ in } P, \text{ i.e. } \Delta v \in \mathcal{P}_{k-2}(P) \\ \bullet \text{ for every face } F \text{ in } \partial P, v|_F \in W_k(F) \end{array} \right.$$

The first three conditions are the three-dimensional equivalent of Definition 2.1. Conversely, the 4<sup>th</sup> condition is added as the boundary of a polyhedron is made of the set of its faces, where information is available for moments up to order  $k - 2$  only. This statement will be much clearer in the following chapter, where VEM will be applied to elastostatics. Once again, a polyhedron of order  $k$  satisfies all of the above requirements in Definition 2.4, so that the following inclusion holds

$$\mathcal{P}_k(P) \subset V_k(P) \tag{2.37}$$

which is essential for convergence properties. The corresponding *local degrees of freedom for the 3D virtual element space* are given by the following.

**Proposition 2.2.** *Local degrees of freedom for the virtual element space embedded in  $\mathbb{R}^3$ .*

*An element  $v$  of the space  $V_k(P)$  defined in Definition 2.4 is uniquely identified by the DOFs*

$$\Xi : V_k(P) \rightarrow \mathbb{R}$$

grouped in the four following sets

$$\left\{ \begin{array}{l}
 \bullet \text{ the value of } v \text{ at the vertices of } P \\
 \bullet \text{ for each edge } E \text{ of } P, \text{ the value of } v \text{ at the } k-1 \text{ internal points of the } (k+1)\text{-point} \\
 \quad \text{Gauss-Lobatto quadrature rule on } E \\
 \bullet \text{ for each face } F \text{ of } P, \text{ the } n_{k-2} \text{ moments up to order } k-2 \text{ of } v \text{ in } F \\
 \\
 \frac{1}{|F|} \int_F v m_\alpha d\Sigma, \quad \alpha = 1, \dots, n_{k-2} \quad (2.38) \\
 \\
 \text{where } m_\alpha \text{ are the scaled monomials defined in (2.1) and } n_{k-2} \text{ in (2.2)} \\
 \bullet \text{ the } \nu_{k-2} \text{ scaled moments up to order } k-2 \text{ of } v \text{ in } P: \\
 \\
 \frac{1}{|P|} \int_P v \mu_\alpha d\Omega, \quad \alpha = 1, \dots, \nu_{k-2} \quad (2.39) \\
 \\
 \text{where } \mu_\alpha \text{ are the scaled monomials defined in (2.4) and } \nu_{k-2} \text{ in (2.5)}
 \end{array} \right.$$

The proof of Proposition 2.2 can be found in [1]. The dimension of the space  $V_k(P)$  coincides with the number of corresponding degrees of freedom  $N_{DOF}$

$$\dim V_k(P) = N_{DOF} = N_V + N_E(k-1) + N_F n_{k-2} + \nu_{k-2} \quad (2.39)$$

where  $N_V$ ,  $N_E$  and  $N_F$  are respectively the number of vertices, edges and faces belonging to the polyhedron  $P$ . The space decomposition of the space  $V_k(P)$  into its basis functions  $\varphi$  through Lagrangian interpolation obviously still applies in an equivalent manner as described in (2.24) and (2.25) for the 2D case. A visual representation of the degrees of freedom defined in Proposition 2.2 is shown in Figure 2.13.

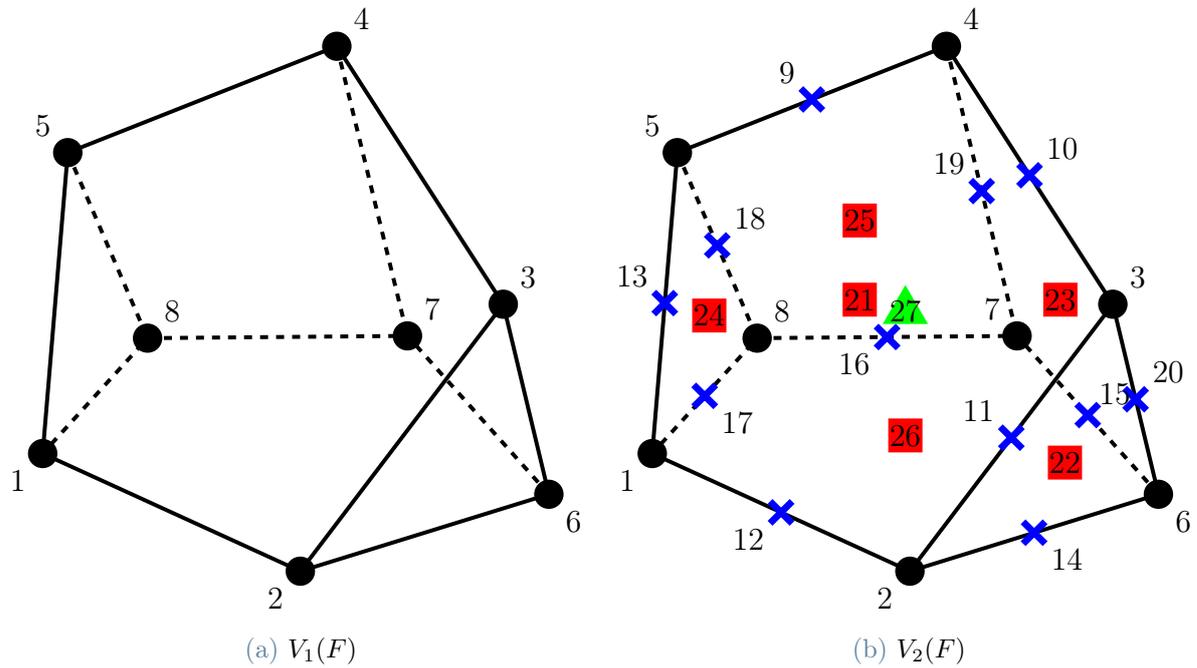


Figure 2.13: Local degrees of freedom for the two virtual element spaces  $V_1(P)$  and  $V_2(P)$  in a pentagonal wedge  $P$ . The black dots correspond to vertices DOFs, blue crosses to edge DOFs, red squares to internal face DOFs and green triangles to internal volume DOFs. Note that while the first two sets match with the function evaluation at the precise location shown in the figure, the last two do not have a geometric punctual representation and are displayed respectively inside the faces and the polyhedron for the sake of simplicity.

As done in the two dimensional setting for the purpose of visualizing the shape functions, one could explicitly solve the boundary value PDE for a given DOF in Proposition 2.2. However, the procedure is more cumbersome with respect to 2D domains since on every face of the polyhedron a PDE has to be solved first to recover the restriction of the shape function on that face, and this requires to compute the projection of the shape function on the face because its restriction belongs to the enhanced space and not the standard virtual space. Figure 2.14 renders a shape function and clarifies the steps to compute it. Once again, it is important to remember that these explicit computations are never required by the VEM and are here shown for clarity only.

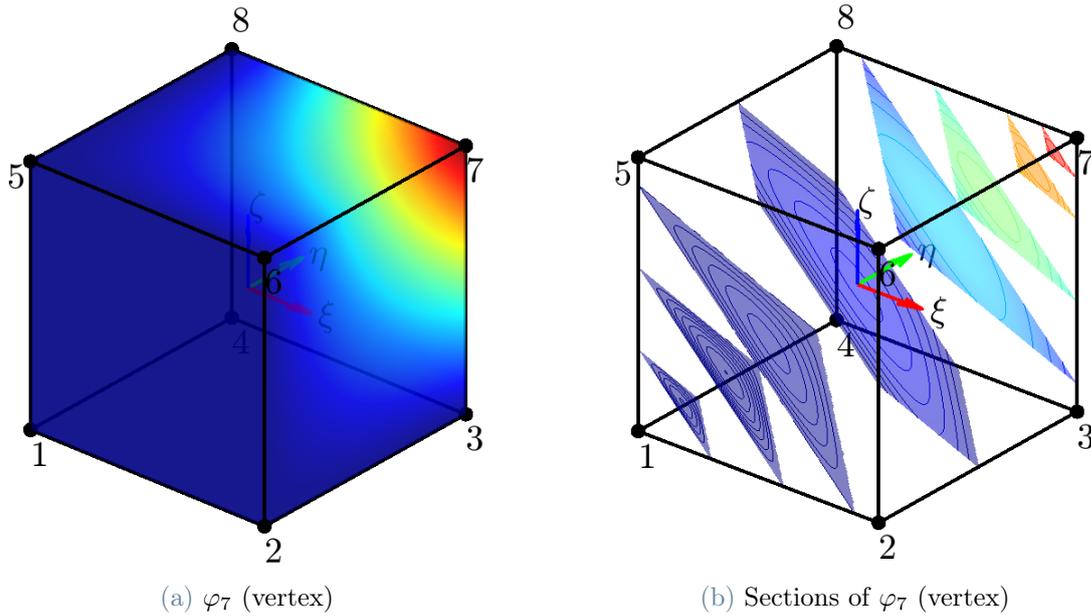


Figure 2.14: Shape function  $\varphi_7$  (a) and sections showing the isolines (b) for a linear virtual hexahedral element. The restriction on the skeleton of the element is linear and nonzero on the edges  $\overline{V_3V_7}$ ,  $\overline{V_6V_7}$  and  $\overline{V_7V_8}$ . Note that the shape function is harmonic inside the element ( $\Delta\varphi_7 = 0$ ). The restriction  $\varphi_7|_F$  on each face  $F$  is found by solving  $\Delta\varphi_7|_F = a\xi_f + b\eta_f + c$  where the coefficients  $a$ ,  $b$  and  $c$  are set so that the 0<sup>th</sup> and 1<sup>st</sup> order moments of the projection  $\Pi_{F,1}^\nabla\varphi_7|_F$  coincide with those of  $\varphi_7|_F$ .

# 3 | Virtual elements for elastostatics in 3D

With the tools presented in Chapter 2 it is now possible to tackle the three-dimensional linear elastostatic problem with the virtual element method. Many choices, including the use of scaled monomials, the selection of the degrees of freedom to describe the virtual element space and the use of the direct projection operator for faces only will be clear as the scheme presented in Subsection 1.3.7 is followed. The polytopic mesh described in Section 2.1 and its conventions to characterize the entities will be adopted.

## 3.1. Definition of the displacement and strain models

Each of the three components of the unknown local displacement vector field will be approximated by the virtual element space presented in Section 2.5. As described in Subsection 1.3.1, we recall that it is possible to collect the basis functions, here represented with the symbol  $N^u$  to cope with usual structural mechanics notation, in the following  $[3 \times n_u]$  matrix  $\mathbf{N}_u$

$$\mathbf{N}_u = \begin{bmatrix} N_1^u & 0 & 0 & N_2^u & 0 & 0 & \dots & N_{N_{DOF}}^u & 0 & 0 \\ 0 & N_1^u & 0 & 0 & N_2^u & 0 & \dots & 0 & N_{N_{DOF}}^u & 0 \\ 0 & 0 & N_1^u & 0 & 0 & N_2^u & \dots & 0 & 0 & N_{N_{DOF}}^u \end{bmatrix} \quad (3.1)$$

where the first  $N_V$   $[3 \times 3]$  blocks correspond to vertex nodal DOFs, the second  $N_E(k-1)$   $[3 \times 3]$  blocks to edge DOFs, the third  $N_F n_{k-2}$   $[3 \times 3]$  blocks to face moment DOFs and the last  $\nu_{k-2}$   $[3 \times 3]$  blocks are associated with volume moment internal DOFs. Matrix  $\mathbf{N}_u$  realizes the map between the local DOFs described in Proposition 2.2, gathered in the vector  $\hat{\mathbf{u}}$ , and the continuous displacement field inside the element  $P$ . Once again, we stress that the shape functions  $N_j^u$  are not explicitly known, unless the corresponding partial differential equation is solved, and will remain unknown even after the solution of the virtual element problem. As anticipated in Section 2.3, the projection operator for the space  $V_k(P)$  will be encapsulated in the definition of the strain field, being this

the symmetric gradient of the displacement. Following the general procedure for mixed finite elements of Chapter 1, the virtual element projection operator is embedded in the compatibility matrix  $\mathbf{C}$ , *projecting* the displacement field (which contains polynomials of degree  $k$  plus other additional functions) into the strain field (made of polynomials up to degree  $k - 1$ ). Namely, matrix  $\mathbf{C}$  projects the local displacement DOFs  $\hat{\mathbf{u}}$  into the parameters  $\hat{\boldsymbol{\varepsilon}}$  of the polynomial strain field and it is the discrete counterpart of the projection

$$\Pi_{P,k} : V_k(P) \rightarrow \mathcal{P}_{k-1}(P) \quad (3.2)$$

In simple terms,  $\mathbf{C}$  encapsulates the gradient operator and the projection operator in one single matrix mapping the discrete parameters. In view of the above considerations, the local strain field is *a priori* defined as it contains a polynomial of order  $k - 1$ . Therefore, following (2.5), matrix  $\mathbf{N}_\varepsilon$  has dimensions  $[6 \times 6\nu_{k-1}]$ , as it is composed by flanking  $\nu_{k-1}$   $[6 \times 6]$  diagonal blocks with the entries of the  $[\nu_{k-1} \times 1]$  vector  $\boldsymbol{\mu}_k$  defined in (2.6)

$$\mathbf{N}_\varepsilon = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \xi & 0 & 0 & 0 & 0 & 0 & \dots & \mu_{\nu_{k-1}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \xi & 0 & 0 & 0 & 0 & \dots & 0 & \mu_{\nu_{k-1}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \xi & 0 & 0 & 0 & \dots & 0 & 0 & \mu_{\nu_{k-1}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \xi & 0 & 0 & \dots & 0 & 0 & 0 & \mu_{\nu_{k-1}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \xi & 0 & \dots & 0 & 0 & 0 & 0 & \mu_{\nu_{k-1}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \xi & \dots & 0 & 0 & 0 & 0 & 0 & \mu_{\nu_{k-1}} \end{bmatrix} \quad (3.3)$$

If the lowest order  $k = 1$  is adopted, the strain model reduces to the  $[6 \times 6]$  identity matrix.

## 3.2. Local stiffness matrix

### 3.2.1. Consistent part of the local stiffness matrix

Having defined the strain and displacement model, it is possible to assembly the consistent part of the local stiffness matrix, starting from the computation of  $\mathbf{C}$ . We recall from (1.40) that we need the two matrices  $\mathbf{G}$  and  $\mathbf{A}$ . The first one, given in (1.37), expands to

the  $[\nu_{k-1} \times \nu_{k-1}]$

$$\begin{aligned}
\mathbf{G} &= \int_P \mathbf{N}_\varepsilon^\top \mathbf{N}_\varepsilon d\Omega = \int_P \begin{bmatrix} \mathbf{I} \\ \xi \mathbf{I} \\ \eta \mathbf{I} \\ \zeta \mathbf{I} \\ \vdots \\ \mu_{\nu_{k-1}} \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \xi \mathbf{I} & \eta \mathbf{I} & \zeta \mathbf{I} & \dots & \mu_{\nu_{k-1}} \mathbf{I} \end{bmatrix} d\Omega = \\
&= \int_P \begin{bmatrix} \mathbf{I} & \xi \mathbf{I} & \eta \mathbf{I} & \zeta \mathbf{I} & \dots & \mu_{\nu_{k-1}} \mathbf{I} \\ \xi \mathbf{I} & \xi^2 \mathbf{I} & \xi \eta \mathbf{I} & \xi \zeta \mathbf{I} & \dots & \xi \mu_{\nu_{k-1}} \mathbf{I} \\ \eta \mathbf{I} & \xi \eta \mathbf{I} & \eta^2 \mathbf{I} & \eta \zeta \mathbf{I} & \dots & \eta \mu_{\nu_{k-1}} \mathbf{I} \\ \zeta \mathbf{I} & \xi \zeta \mathbf{I} & \eta \zeta \mathbf{I} & \zeta^2 \mathbf{I} & \dots & \zeta \mu_{\nu_{k-1}} \mathbf{I} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_{\nu_{k-1}} \mathbf{I} & \mu_{\nu_{k-1}} \xi \mathbf{I} & \mu_{\nu_{k-1}} \eta \mathbf{I} & \mu_{\nu_{k-1}} \zeta \mathbf{I} & \dots & \mu_{\nu_{k-1}}^2 \mathbf{I} \end{bmatrix} d\Omega \quad (3.4)
\end{aligned}$$

where  $\mathbf{I}$  is the  $[6 \times 6]$  identity matrix. The integrals in (3.4) are computable *exactly* (up to machine precision) by means of numerical integration techniques over polyhedral domains. A brief discussion of the available methods is due and reported here.

- A straightforward approach is to *sub-tetrahedralize* the polyhedron, adopt Gaussian integration over each tetrahedron and sum the result of the integrals. This however, requires a considerable computational effort as the number of Gauss points increases with cubic power of the order of the method  $k$ . In fact,  $n$ -point Gaussian quadrature in one-dimensional domains guarantees exact integration of polynomials of order  $2n - 1$ .

$$k = 2n - 1 \implies n = \left\lceil \frac{k+1}{2} \right\rceil \quad (3.5)$$

Exploiting tensor product, the rule is able to integrate exactly polynomials up to order  $k$  for  $d$ -dimensional domains if  $\left\lceil \frac{k+1}{2} \right\rceil^d$  points are used, where  $\lceil (\cdot) \rceil$  is the ceiling operator applied to the quantity  $(\cdot)$

$$n = \left\lceil \frac{k+1}{2} \right\rceil^d \quad (3.6)$$

Assuming the polyhedron has  $N_F$  faces, a simple tratrahedralization (only valid for star-shaped<sup>1</sup> polyhedra) splits the faces in triangles and generates the corresponding tetrahedra with an internal chosen common point, the center of the star. We indicate

<sup>1</sup>A set  $\Omega$  in the Euclidean space  $\mathbb{R}^d$  is a star-shaped domain (or radially convex set) if there exists an  $\mathbf{x}_0 \in \Omega$ , the center of the star, such that for all  $\mathbf{x} \in \Omega$ , the line segment  $\overline{\mathbf{x}_0 \mathbf{x}}$  lies in  $\Omega$ .

with  $N_\Delta$  the number of triangles resulting from face triangulations, given by

$$N_\Delta = \sum_{f=1}^{N_F} (N_{f,E} - 2)$$

where  $N_{f,E}$  are the numbers of edges of the face indexed with  $f$ . As an example, a hexahedron, whose 6 faces are quadrilaterals, has  $N_\Delta = \sum_{f=1}^6 (4 - 2) = 12$ , hence its faces will be decomposed in 12 triangles. Following the above notation, the number of points required to exactly integrate a polynomial of order  $k$  described in (3.6) becomes, for a polyhedral domain,

$$n = N_\Delta \left\lceil \frac{k+1}{2} \right\rceil^3 \quad (3.7)$$

Now, a generic  $[n \times n]$  symmetric matrix has  $\frac{n(n+1)}{2}$  independent entries. The  $[6\nu_{k-1} \times 6\nu_{k-1}]$  matrix  $\mathbf{G}$  is block-symmetric and thus requires to compute in general the  $\frac{\nu_{k-1}(\nu_{k-1}+1)}{2}$  independent components. These are monomials up to order  $2(k-1)$  and hence require to compute

$$n = N_\Delta \left\lceil \frac{2(k-1)+1}{2} \right\rceil^3 = N_\Delta \left\lceil \frac{2k-1}{2} \right\rceil^3 = N_\Delta k^3 \quad \forall k \in \mathbb{N} \quad (3.8)$$

Gauss points for each polyhedron  $P$  and for each set of these points they require to perform  $\frac{\nu_{k-1}(\nu_{k-1}+1)}{2}$  function evaluations to fill the matrix  $\mathbf{G}$ . To better understand this, we apply (3.8) to a hexahedron and compute the number of Gauss points required following the above-presented method in Table 3.1.

	Independent entries of $\mathbf{G}$	Gauss points	Function evaluations
$k = 1$	1	12	12
$k = 2$	10	96	960
$k = 3$	55	324	17820
$k = 4$	210	768	161280

**Table 3.1:** Computational cost to produce the entries of matrix  $\mathbf{G}$  using Gauss sub-tetrahedralization for a hexahedron as a function of the order  $k$  of the VEM.

Already with  $k = 2$ , the number of function evaluations to populate one single matrix  $\mathbf{G}$  sensibly increases, underlying the weakness of Gaussian quadrature for

this purpose.

- Some variants of the above technique manage to reduce the number of points needed for the integration over tetrahedra (e.g. [41]), though still needing to process sub-tetrahedralization to handle polyhedra. Other rules avoid the need of subdivision of the integration domain in tetrahedra, as presented in [49] and in the brand new [58], or belong to the *compressed* integration techniques, which manage to reduce the number of evaluation points (as in [8]). However, the latter are mostly available for 2D domains, and the computational cost required to produce such schemes is not negligible.
- The choice adopted in this work follows a recursive algorithm presented in [4], a quadrature-free<sup>2</sup> integration scheme developed *ad hoc* to build stiffness matrices for discontinuous Galerkin methods on general polytopic meshes (see Appendix B).

The second matrix involved in the computation of  $\mathbf{C}$  is the  $[6\nu_{k-1} \times n_u]$  matrix  $\mathbf{A}$ , defined in (1.41). Performing integration by-parts, one obtains

$$\begin{aligned} \mathbf{A} &= \int_P \mathbf{N}_\varepsilon^\top \mathbf{S} \mathbf{N}_u d\Omega = \int_{\partial P} (\mathbb{N}_P \mathbf{N}_\varepsilon)^\top \mathbf{N}_u d\Sigma - \int_P (\mathbf{S}^\top \mathbf{N}_\varepsilon)^\top \mathbf{N}_u d\Omega = \\ &= \sum_{F \in \partial P} \left[ \int_F (\mathbb{N}_F \mathbf{N}_\varepsilon)^\top \mathbf{N}_u d\Sigma \right] - \int_P (\mathbf{S}^\top \mathbf{N}_\varepsilon)^\top \mathbf{N}_u d\Omega \end{aligned} \quad (3.9)$$

where  $\mathbb{N}_P$  is the  $[3 \times 6]$  matrix collecting the direction cosines of the outward normal unit vector on the surface  $\partial P$ , and  $\mathbb{N}_F$  the corresponding one for a single face  $F$ . It is convenient to separate the two quantities of (3.9), and analyze their properties.

$$\mathbf{A}_1 = \sum_{F \in \partial P} \left[ \int_F (\mathbb{N}_F \mathbf{N}_\varepsilon)^\top \mathbf{N}_u d\Sigma \right] \quad (3.10)$$

$$\mathbf{A}_2 = - \int_P (\mathbf{S}^\top \mathbf{N}_\varepsilon)^\top \mathbf{N}_u d\Omega \quad (3.11)$$

Each term in the summation of  $\mathbf{A}_1$  is a matrix made of integrals *over a polygon* of a polynomial of at most degree  $k - 1$  multiplied with the unknown basis functions. This is one of the key differences in implementing VE in 3D with respect to a two-dimensional setting, where the boundary of the polygonal element is made by its *edges*, where the basis functions are *explicitly known* from the degrees of freedom. Conversely, in three dimensions, only the integrals of the shape functions multiplied by a polynomial up to

---

<sup>2</sup>A quadrature-free integration scheme does not adopt integration points, exploiting only analytical-type evaluations of the geometry of the domain of integration.

order  $k - 2$  can be directly addressed by the internal face DOFs. To tackle the problem for the remaining  $(k - 1)$ -degree monomials, we can draw from Theorem 2.1

$$\int_F m_{\alpha} N_j^u d\Sigma = \int_F m_{\alpha} \Pi_{F,k}^{\nabla} N_j^u d\Sigma \quad \forall \alpha : |\alpha| = k - 1 \quad (3.12)$$

and we can explicitly compute the integrals through the projection  $\Pi_{F,k}^{\nabla} N_j^u$ , whose coefficients come from solving the linear algebraic system (2.31). Expanding  $\mathbf{A}_1$  leads to

$$\begin{aligned} \mathbf{A}_1 &= \sum_{F \in \partial P} \left[ \int_F (\mathbb{N}_F \mathbf{N}_{\varepsilon})^T \mathbf{N}_u d\Sigma \right] = \\ &= \sum_{F \in \partial P} \left[ \int_F \begin{bmatrix} \mathbb{N}_F \\ \xi \mathbb{N}_F \\ \eta \mathbb{N}_F \\ \zeta \mathbb{N}_F \\ \vdots \\ \mu_{\nu_{k-1}} \mathbb{N}_F \end{bmatrix} \begin{bmatrix} N_1^u \mathbf{I} & N_2^u \mathbf{I} & \dots & N_{N_{DOF}}^u \mathbf{I} \end{bmatrix} d\Sigma \right] = \\ &= \sum_{F \in \partial P} \left[ \int_F \begin{bmatrix} \mathbb{N}_F N_1^u & \mathbb{N}_F N_2^u & \dots & \mathbb{N}_F N_{N_{DOF}}^u \\ \xi \mathbb{N}_F N_1^u & \xi \mathbb{N}_F N_2^u & \dots & \xi \mathbb{N}_F N_{N_{DOF}}^u \\ \eta \mathbb{N}_F N_1^u & \eta \mathbb{N}_F N_2^u & \dots & \eta \mathbb{N}_F N_{N_{DOF}}^u \\ \zeta \mathbb{N}_F N_1^u & \zeta \mathbb{N}_F N_2^u & \dots & \zeta \mathbb{N}_F N_{N_{DOF}}^u \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{\nu_{k-1}} \mathbb{N}_F N_1^u & \mu_{\nu_{k-1}} \mathbb{N}_F N_2^u & \dots & \mu_{\nu_{k-1}} \mathbb{N}_F N_{N_{DOF}}^u \end{bmatrix} d\Sigma \right] \quad (3.13) \end{aligned}$$

where  $\mathbf{I}$  is the  $[3 \times 3]$  identity matrix. Each  $[6 \times 3]$  submatrix in between rows  $6(i - 1) + 1$  and  $6i$  and between columns  $3(j - 1) + 1$  and  $3j$  contains the direction cosines of the outward normal unit vector of the face multiplied by the  $i^{\text{th}}$  scaled monomial  $m_i$  and  $j^{\text{th}}$  shape function  $N_j^u$ . Explicitly expanded, each block reads

$$\mu_i \mathbb{N}_F N_j^u = \begin{bmatrix} \mu_i n_x N_j^u & 0 & 0 \\ 0 & \mu_i n_y N_j^u & 0 \\ 0 & 0 & \mu_i n_z N_j^u \\ \mu_i n_y N_j^u & \mu_i n_x N_j^u & 0 \\ 0 & \mu_i n_z N_j^u & \mu_i n_y N_j^u \\ \mu_i n_z N_j^u & 0 & \mu_i n_x N_j^u \end{bmatrix}$$

According to the order  $k$  of the VEM and the type of shape function being integrated,

the entries of  $\mathbf{A}_1$  can be directly computed from the virtual DOFs or require a projection operation through (3.12). First we focus on vertex-type, edge-type and face-type DOFs for the entries of  $\mathbf{A}_1$ . For each face, the  $[6 \times 3]$  blocks up to row  $6\nu_{k-2}$  contain restrictions on polygons of monomials up to degree  $k-2$  in  $\mathbb{R}^3$  multiplied by shape functions, which can be transformed into a combination of the virtual degrees of freedom through the change of coordinates described in (2.10). Lagrangian interpolation property ensures that in these first  $6\nu_{k-2}$  rows, only the columns corresponding to face-type DOFs have non-zero entries. For the remaining rows, from  $6\nu_{k-2} + 1$  to  $6\nu_{k-1}$ , the projection operation described in Section 2.3 has to be applied to find the coefficients of the monomials up to order  $k$  of the polynomial  $\Pi_{F,k}^\nabla N_j^u$  appearing in the identity (3.12). Once these are found, the last entries of  $\mathbf{A}_1$  are obtained by integrating the resulting polynomial of degree  $(k-1) + k = 2k-1$ . The polyhedron-type DOFs were left out in  $\mathbf{A}_1$  and indeed their contribution is zero. In fact, by inspecting the right hand side (2.33) of the projection linear algebraic system, one notices that for a polyhedron-type shape function  $N_j^u$

$$\frac{1}{|F|} \int_F N_j^u d\Sigma = 0$$

thanks to Lagrangian interpolation property, which ensures that the above quantity is 1 if and only if  $j$  corresponds to the so-defined face-type DOF. Similarly, after performing the change of coordinates described in (2.10), the quantity

$$\int_F \Delta m_\alpha N_j^u d\Sigma = 0 \quad \forall \alpha : |\alpha| = k-1$$

again due to Lagrangian interpolation property with respect to face-type DOFs. Lastly,

$$\int_E \nabla m_\alpha \cdot \mathbf{n} N_j^u d\Gamma = 0$$

due to Lagrangian interpolation property with respect to vertex-type and edge-type DOFs, and since the shape functions are *exactly* polynomials on the edges of the polyhedron.

Matrix  $\mathbf{A}_2$  appears only when the order of the VEM  $k \geq 2$ , since the derivatives of the constant strain field vanish for  $k = 1$ . The divergence of the strain field in  $[3 \times 6\nu_{k-1}]$

matrix form  $\mathbf{S}^T \mathbf{N}_\varepsilon$  expands to

$$\begin{aligned}
\mathbf{S}^T \mathbf{N}_\varepsilon &= \begin{bmatrix} \partial_x & 0 & 0 & \partial_y & 0 & \partial_z \\ 0 & \partial_y & 0 & \partial_x & \partial_z & 0 \\ 0 & 0 & \partial_z & 0 & \partial_y & \partial_x \end{bmatrix} \begin{bmatrix} \mathbf{I} & \xi \mathbf{I} & \eta \mathbf{I} & \zeta \mathbf{I} & \dots & \mu_{\nu_{k-1}} \mathbf{I} \end{bmatrix} = \\
&= \frac{1}{h_P} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \dots & \partial_\xi \mu_{\nu_{k-1}} & 0 & 0 & \partial_\eta \mu_{\nu_{k-1}} & 0 & \partial_\zeta \mu_{\nu_{k-1}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 & \partial_\eta \mu_{\nu_{k-1}} & 0 & \partial_\xi \mu_{\nu_{k-1}} & \partial_\zeta \mu_{\nu_{k-1}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \dots & 0 & 0 & \partial_\zeta \mu_{\nu_{k-1}} & 0 & \partial_\eta \mu_{\nu_{k-1}} & \partial_\xi \mu_{\nu_{k-1}} & 0 & 0 \end{bmatrix} = \\
&= \frac{1}{h_P} \tilde{\mathbf{M}} \tag{3.14}
\end{aligned}$$

where  $\mathbf{I}$  is the  $[6 \times 6]$  identity matrix. The  $[3 \times 6\nu_{k-1}]$  matrix  $\tilde{\mathbf{M}}$  is constant for all the elements of the virtual element program and can also be expressed as a combination of the scaled monomials  $\mu_{k-2}$

$$\tilde{\mathbf{M}} = \sum_{i=1}^{\nu_{k-2}} \tilde{\mathbf{M}}_i \mu_i \tag{3.15}$$

where  $\tilde{\mathbf{M}}_i$  gathers the coefficients of the scaled monomial  $\mu_i$ . Substituting (3.15) in (3.14) and plugging back in (3.11) one obtains

$$\begin{aligned}
\mathbf{A}_2 &= - \int_P (\mathbf{S}^T \mathbf{N}_\varepsilon)^T \mathbf{N}_u d\Omega = - \frac{1}{h_P} \int_P \sum_{i=1}^{\nu_{k-2}} \left[ \tilde{\mathbf{M}}_i^T \mu_i \right] \mathbf{N}_u d\Omega = \\
&= - \frac{1}{h_P} \sum_{i=1}^{\nu_{k-2}} \left[ \tilde{\mathbf{M}}_i^T \left( \int_P \mu_i \mathbf{N}_u d\Omega \right) \right] \tag{3.16}
\end{aligned}$$

By Lagrangian interpolation property, the quantities in round brackets assume values equal to the volume  $|P|$  of the element if the shape function refers to the corresponding internal polyhedron-type DOF, and zero otherwise.

Having assembled  $\mathbf{G}$  and  $\mathbf{A}$ , the  $[6\nu_{k-1} \times n_u]$  compatibility matrix  $\mathbf{C}$  is computed according to (1.40), and the  $[n_u \times n_u]$  local consistent part of the stiffness matrix  $\mathbf{K}_\varepsilon^c$ , following (1.46),

requires the  $[\nu_{k-1} \times \nu_{k-1}]$  elastic matrix  $\mathbf{E}$ , which expands to the fully computable

$$\begin{aligned} \mathbf{E} &= \int_P \mathbf{N}_\varepsilon^\top \mathbf{D} \mathbf{N}_\varepsilon d\Omega = \int_P \begin{bmatrix} \mathbf{I} \\ \xi \mathbf{I} \\ \eta \mathbf{I} \\ \zeta \mathbf{I} \\ \vdots \\ \mu_{\nu_{k-1}} \mathbf{I} \end{bmatrix} \mathbf{D} \begin{bmatrix} \mathbf{I} & \xi \mathbf{I} & \eta \mathbf{I} & \zeta \mathbf{I} & \dots & \mu_{\nu_{k-1}} \mathbf{I} \end{bmatrix} d\Omega = \\ &= \int_P \begin{bmatrix} \mathbf{D} & \xi \mathbf{D} & \eta \mathbf{D} & \zeta \mathbf{D} & \dots & \mu_{\nu_{k-1}} \mathbf{D} \\ \xi \mathbf{D} & \xi^2 \mathbf{D} & \xi \eta \mathbf{D} & \xi \zeta \mathbf{D} & \dots & \xi \mu_{\nu_{k-1}} \mathbf{D} \\ \eta \mathbf{D} & \xi \eta \mathbf{D} & \eta^2 \mathbf{D} & \eta \zeta \mathbf{D} & \dots & \eta \mu_{\nu_{k-1}} \mathbf{D} \\ \zeta \mathbf{D} & \xi \zeta \mathbf{D} & \eta \zeta \mathbf{D} & \zeta^2 \mathbf{D} & \dots & \zeta \mu_{\nu_{k-1}} \mathbf{D} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_{\nu_{k-1}} \mathbf{D} & m_{\nu_{k-1}} \xi \mathbf{D} & \mu_{\nu_{k-1}} \eta \mathbf{D} & \mu_{\nu_{k-1}} \zeta \mathbf{D} & \dots & \mu_{\nu_{k-1}}^2 \mathbf{D} \end{bmatrix} d\Omega \end{aligned} \quad (3.17)$$

### 3.2.2. Stabilizing part of the local stiffness matrix

The computation of the stabilizing matrix requires the assembly of the  $[n_u \times n_u]$  hourglass matrix  $\mathbf{H}$ , described in (1.62), which in turn reduces to the computation of the  $[n_u \times n_{D+R}]$  matrix  $\mathbf{T}_{D+R}$ . In the context of the virtual element method, the rank deficiency of the consistent local stiffness matrix  $\mathbf{K}_\varepsilon^c$  is due to the extra non-polynomial functions allowable in the displacement field. If the displacement model was made of polynomials up to degree  $k$  only, then the  $k - 1$  polynomial strain field would capture all possible deforming modes and no stabilization would be required as hourglass modes would not arise. Therefore, the number of hourglass modes is equal to  $n_u - 3\nu_k$ , where indeed the number of parameters  $n_u$  (coinciding with the dimension of the virtual space for the displacement field) has been subtracted by the dimension  $3\nu_k$  of the space spanned by the monomials up to degree  $k$  in three dimensions. Following the arguments above, the approximate displacement field purified from the hourglass modes can be expressed as

$$\mathbf{u}_{D+R}(\boldsymbol{\xi}) = \mathbf{N}_k(\boldsymbol{\xi}) \hat{\mathbf{p}}_{D+R} = \mathbf{N}_u(\boldsymbol{\xi}) \hat{\mathbf{u}}_{D+R} \quad (3.18)$$

where the  $[3 \times 3\nu_k]$  matrix  $\mathbf{N}_k$  gathers the scaled monomials  $\boldsymbol{\mu}_k$

$$\mathbf{N}_k = \begin{bmatrix} 1 & 0 & 0 & \xi & 0 & 0 & \dots & \mu_{\nu_k} & 0 & 0 \\ 0 & 1 & 0 & 0 & \xi & 0 & \dots & 0 & \mu_{\nu_k} & 0 \\ 0 & 0 & 1 & 0 & 0 & \xi & \dots & 0 & 0 & \mu_{\nu_k} \end{bmatrix} \quad (3.19)$$

Recalling from (1.52) that

$$\hat{\mathbf{u}}_{D+R} = \mathbf{T}_{D+R} \hat{\mathbf{p}}_{D+R}$$

the identity (3.18) becomes

$$\mathbf{N}_k(\boldsymbol{\xi}) \hat{\mathbf{p}}_{D+R} = \mathbf{N}_u(\boldsymbol{\xi}) \mathbf{T}_{D+R} \hat{\mathbf{p}}_{D+R} \quad (3.20)$$

Applying each one of the maps of the local degrees of freedom described in Proposition 2.2 to the above identity, the entries of matrix  $\mathbf{T}_{D+R}$  are found. In other words, the procedure exploits the evaluation of each of the DOFS at the coordinates contained in matrix  $\mathbf{N}_k$  and exploits the Lagrangian interpolation property to assess the entries of matrix  $\mathbf{N}_u$ . Explicitly, if the first DOF of vertex-type is evaluated, indicating with  $\mathbf{I}$  the  $[3 \times 3]$  identity matrix, one has

$$\begin{bmatrix} \mathbf{I} & \xi_1 \mathbf{I} & \dots & \mu_{\nu_k 1} \mathbf{I} \end{bmatrix} \hat{\mathbf{p}}_{D+R} = \begin{bmatrix} \mathbf{I} & \mathbf{0I} & \dots & \mathbf{0I} \end{bmatrix} \mathbf{T}_{D+R} \hat{\mathbf{p}}_{D+R}$$

since  $N_1^u(\boldsymbol{\xi}_j) = \delta_{1j} \quad \forall j = 1, \dots, N_{N_{DOF}}$ . Similarly, evaluating a generic vertex-type or edge-type DOF labelled with  $j$ , one has

$$\begin{bmatrix} \mathbf{I} & \xi_j \mathbf{I} & \dots & \mu_{\nu_k j} \mathbf{I} \end{bmatrix} \hat{\mathbf{p}}_{D+R} = \begin{bmatrix} \mathbf{0I} & \dots & \mathbf{I} & \dots & \mathbf{0I} \end{bmatrix} \mathbf{T}_{D+R} \hat{\mathbf{p}}_{D+R} \quad (3.21)$$

where the only three non-zero entries in the matrix on the right hand side lie in  $j^{\text{th}}$   $[3 \times 3]$  block. Evaluating a face-type DOF  $j$  yields to

$$\begin{bmatrix} f_F m_j \mathbf{I} & f_F \xi m_j \mathbf{I} & \dots & f_F \mu_{\nu_k} m_j \mathbf{I} \end{bmatrix} \hat{\mathbf{p}}_{D+R} = \begin{bmatrix} \mathbf{0I} & \dots & \mathbf{I} & \dots & \mathbf{0I} \end{bmatrix} \mathbf{T}_{D+R} \hat{\mathbf{p}}_{D+R} \quad (3.22)$$

where the only three non-zero entries in the matrix on the right hand side lie in  $j^{\text{th}}$   $[3 \times 3]$  block after the blocks corresponding to vertex-type and edge-type and the symbol  $f_\Omega(\cdot)$  stands for the averaged integral of  $(\cdot)$

$$f_\Omega(\cdot) = \frac{1}{|\Omega|} \int_\Omega (\cdot) d\Omega$$

Finally, if a polyhedron-type DOF  $j$  is evaluated<sup>3</sup>, the following identity is obtained

$$\begin{bmatrix} f_P \mu_j \mathbf{I} & f_P \xi \mu_j \mathbf{I} & \dots & f_P \mu_{\nu_k} \mu_j \mathbf{I} \end{bmatrix} \hat{\mathbf{p}}_{D+R} = \begin{bmatrix} \mathbf{0I} & \dots & \mathbf{I} & \dots & \mathbf{0I} \end{bmatrix} \mathbf{T}_{D+R} \hat{\mathbf{p}}_{D+R} \quad (3.23)$$

---

<sup>3</sup>In the example  $j$  is not the last DOF of the collection, otherwise in the right hand side the last  $[3 \times 3]$  block is the one containing the only non-zero entries.

where again the only three non-zero entries in the matrix on the right hand side lie in  $j^{\text{th}}$   $[3 \times 3]$  block after the blocks corresponding to face-type DOFs. In order to satisfy (3.21), (3.22) and (3.23) for every corresponding DOF, the matrix  $\mathbf{T}_{D+R}$  has to contain the respective evaluation of  $\mathbf{N}_k$  displayed on the left hand sides of the identities at the corresponding  $j^{\text{th}}$  row-block. Hence, the  $[n_u \times 3\nu_k]$  matrix  $\mathbf{T}_{D+R}$  explicitly becomes

$$\mathbf{T}_{D+R} = \begin{bmatrix} \mathbf{I} & \xi_1 \mathbf{I} & \eta_1 \mathbf{I} & \zeta_1 \mathbf{I} & \dots & \mu_{\nu_k 1} \mathbf{I} \\ \mathbf{I} & \xi_2 \mathbf{I} & \eta_2 \mathbf{I} & \zeta_2 \mathbf{I} & \dots & \mu_{\nu_k 2} \mathbf{I} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ f_F \mathbf{I} & f_F \xi \mathbf{I} & f_F \eta \mathbf{I} & f_F \zeta \mathbf{I} & \dots & f_F \mu_{\nu_k} \mathbf{I} \\ f_F \xi_f \mathbf{I} & f_F \xi_f \xi \mathbf{I} & f_F \xi_f \eta \mathbf{I} & f_F \xi_f \zeta \mathbf{I} & \dots & f_F \xi_f \mu_{\nu_k} \mathbf{I} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ f_P \mathbf{I} & f_P \xi \mathbf{I} & f_P \eta \mathbf{I} & f_P \zeta \mathbf{I} & \dots & f_P \mu_{\nu_k} \mathbf{I} \\ f_P \xi \mathbf{I} & f_P \xi^2 \mathbf{I} & f_P \xi \eta \mathbf{I} & f_P \xi \zeta \mathbf{I} & \dots & f_P \xi \mu_{\nu_k} \mathbf{I} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix} \quad (3.24)$$

where  $\mathbf{I}$  is again the  $[3 \times 3]$  identity matrix. Computing the hourglass matrix  $\mathbf{H}$  requires the inversion of  $(\mathbf{T}_{D+R})^T \mathbf{T}_{D+R}$ , which however is computationally acceptable considering the inversion has to be performed once per element, the matrices are sparse and their dimension is relatively small.

### 3.3. Equivalent nodal forces vector

Recalling the expression of the equivalent nodal forces vector reported in (1.42) clearly establishes how the explicit computation is unfeasible. A suitable projection has to be performed to tackle both the body forces and the surface tractions, when these are applied. For the latter

$$\mathbf{F}_e^p = \int_{\partial_p P} \mathbf{N}_u^T \mathbf{p} d\sigma \quad (3.25)$$

where  $\mathbf{p}$  is the  $[3 \times 1]$  vector gathering the surface tractions, the computation can be done by exploiting the projections of  $\mathbf{N}_u$  already computed for the entries of  $\mathbf{A}_1$ . For the former, we recall the definition

$$\mathbf{F}_e^b = \int_P \mathbf{N}_u^T \mathbf{b} d\Omega \quad (3.26)$$

where  $\mathbf{b}$  is the  $[3 \times 1]$  vector gathering the body forces per unit volume. Different choices are available to approximate the above integral. In [1] it is shown that optimal error estimate in the  $H^1$  and  $L^2$  norm is obtained if the shape functions  $N_i^u$  are projected into

the space of polynomials of degree  $k$  through the projector  $\Pi_{P,k}^0$ <sup>4</sup>. In the same [1] it is also proved that optimal convergence rates are still achieved if the projector  $\Pi_{P,k-1}^0$  is applied to the displacement basis  $N_i^u$  for VEM with order  $k = 1, 2$  and if the projector  $\Pi_{P,k-2}^0$  is applied for order  $k > 2$ . In this work the approximation of the equivalent nodal forces vector is distinguished in the cases  $k = 1$  and  $k \geq 1$ .

### 3.3.1. Approximation of body forces for first order VEM

If  $k = 1$ , the approximation of the body forces can be achieved by projecting the shape functions onto the space of constants. The technique is exactly the same as the one introduced in the first condition of (2.27) for the computation of the right hand side of the algebraic system for  $\Pi_{F,1}^\nabla$ . Given the  $[3 \times 1]$  vector  $\mathbf{f}$

$$\mathbf{f} = \begin{Bmatrix} f_x \\ f_y \\ f_z \end{Bmatrix} = \int_P \begin{Bmatrix} b_x \\ b_y \\ b_z \end{Bmatrix} d\Omega = \int_P \mathbf{b} d\Omega$$

The equivalent nodal forces vector for the body components can be computed as

$$\begin{aligned} \mathbf{F}_e^b &= \int_P \mathbf{N}_u^T \mathbf{b} d\Omega \approx \int_P [\Pi_{P,0}^0 \mathbf{N}_u]^T \mathbf{b} d\Omega = \\ &= \int_P \begin{bmatrix} \frac{1}{N_V} \mathbf{I} & \frac{1}{N_V} \mathbf{I} & \dots & \frac{1}{N_V} \mathbf{I} \end{bmatrix}^T \mathbf{b} d\Omega = \\ &= \frac{1}{N_V} \left\{ f_x \quad f_y \quad f_z \quad \dots \quad f_x \quad f_y \quad f_z \right\}^T \end{aligned} \quad (3.27)$$

where  $\mathbf{I}$  is the  $[3 \times 3]$  identity matrix and  $N_V$  the number of vertices of the element. In other words, the body forces are uniformly distributed to the vertices of the polyhedron.

### 3.3.2. Approximation of body forces for higher order VEM

If  $k \geq 2$  the body forces  $\mathbf{b}$  are projected in the space of polynomials of degree  $k - 2$ , so that the approximated body forces  $\mathbf{b}^h$  can be expressed as

$$\mathbf{b}^h = \Pi_{P,k-2}^0 \mathbf{b} = \sum_{i=1}^{\nu_{k-2}} \mu_i \hat{\mathbf{b}}_i^h \quad (3.28)$$

---

<sup>4</sup>The projection  $\Pi_{P,k}^0$  is similarly achieved as described in Section 2.3 for the operator  $\Pi_{F,k}^\nabla$ , but applied so that the orthogonality conditions hold between the function and the monomial instead of between their gradients, and the domain is polyhedral. Moreover, the enhance property of Theorem 2.1 is exploited to compute the right hand side vector of the system leading to the coefficients of the polynomial projection.

where  $\hat{\mathbf{b}}_i^h$  is the  $[3 \times 1]$  vector gathering the coefficients for the monomial  $\mu_i$ . These can be computed in a similar manner as presented in Section 2.3. The orthogonality condition in this case reads

$$\int_P \mu_j (\mathbf{b}^h - \mathbf{b}) d\Omega = \mathbf{0} \quad \forall j \leq \nu_{k-2} \quad (3.29)$$

which, after substituting (3.28) in place of  $\mathbf{b}^h$ , translates into

$$\int_P \mu_j \left[ \sum_{i=1}^{\nu_{k-2}} (\mu_i \hat{\mathbf{b}}_i^h) - \mathbf{b} \right] d\Omega = \mathbf{0} \quad \forall j \leq \nu_{k-2}, \quad \forall i = 1, \dots, \dim V_k(F) \quad (3.30)$$

and rearranging leads to

$$\sum_{i=1}^{\nu_{k-2}} \left[ \left( \int_P \mu_j \mu_i d\Omega \right) \hat{\mathbf{b}}_i^h \right] = \int_P \mu_j \mathbf{b} d\Omega \quad \forall j \leq \nu_{k-2} \quad (3.31)$$

which is an algebraic linear system in the unknown  $[3\nu_{k-2} \times 1]$  vector  $\hat{\mathbf{b}}^h$ . Explicitly, the above linear system reads

$$\begin{bmatrix} \left( \int_P 1 \right) \mathbf{I} & \left( \int_P \xi \right) \mathbf{I} & \cdots & \left( \int_P \mu_{\nu_{k-2}} \right) \mathbf{I} \\ \left( \int_P \xi \right) \mathbf{I} & \left( \int_P \xi^2 \right) \mathbf{I} & \cdots & \left( \int_P \xi \mu_{\nu_{k-2}} \right) \mathbf{I} \\ \vdots & \vdots & \ddots & \vdots \\ \left( \int_P \mu_{\nu_{k-2}} \right) \mathbf{I} & \left( \int_P \xi \mu_{\nu_{k-2}} \right) \mathbf{I} & \cdots & \left( \int_P \mu_{\nu_{k-2}}^2 \right) \mathbf{I} \end{bmatrix} \begin{Bmatrix} \hat{\mathbf{b}}_1^h \\ \hat{\mathbf{b}}_2^h \\ \vdots \\ \hat{\mathbf{b}}_{\nu_{k-2}}^h \end{Bmatrix} = \begin{Bmatrix} \int_P \mathbf{b} \\ \int_P \xi \mathbf{b} \\ \vdots \\ \int_P \mu_{\nu_{k-2}} \mathbf{b} \end{Bmatrix} \quad (3.32)$$

where  $\mathbf{I}$  is the  $[3 \times 3]$  identity matrix and the differentials  $d\Omega$  have been omitted for conciseness. Solving for  $\hat{\mathbf{b}}^h$  allows to find the direct coefficients to associate with the equivalent nodal forces vector. Indeed, plugging the approximation (3.28) in (3.26) leads to

$$\mathbf{F}_e^b = \int_P \mathbf{N}_u^T \mathbf{b} d\Omega \approx \int_P \mathbf{N}_u^T \mathbf{b}^h d\Omega = \sum_{i=1}^{\nu_{k-2}} \left[ \left( \int_P \mu_i \mathbf{N}_u^T d\Omega \right) \hat{\mathbf{b}}_i^h \right] \quad (3.33)$$

The quantities in round brackets are known directly from the polyhedron-type internal DOFs, available if the VEM order is  $k \geq 2$ . Specifically, by the Lagrangian interpolation property, the vector of local equivalent nodal forces, contains all zeros up to the previous index corresponding to the first polyhedron-type DOF, and then it contains exactly the entries of  $\hat{\mathbf{b}}^h$ . Explicitly

$$\mathbf{F}_e^b = \left\{ 0 \ 0 \ 0 \ \dots \ \hat{\mathbf{b}}_1^h \ \hat{\mathbf{b}}_2^h \ \dots \ \hat{\mathbf{b}}_{\nu_{k-2}}^h \right\}^T \quad (3.34)$$



# 4 | Numerical tests of the virtual element method for 3D elastostatics

This chapter presents numerical results obtained by a MATLAB ([40]) implementation of the virtual element method for three-dimensional linear elastostatics described in Chapter 3, up to order 2. Henceforth, the units of measurement are not specified and can be arbitrarily taken provided they are consistent (e.g.,  $N/mm^3$  for body forces;  $N/mm^2$  for stresses, surface tractions, Young's modulus and Lamé parameters;  $mm$  for lengths).

## 4.1. Data of the problem

We recall the linear elastostatic problem in the unknown displacements  $\mathbf{u}(\mathbf{x})$

$$\begin{cases} -\mathbf{S}^T [\mathbf{D}(\mathbf{S}\mathbf{u})] = \mathbf{b} & \text{in } \Omega \\ \mathbb{N}[\mathbf{D}(\mathbf{S}\mathbf{u})] = \mathbf{p} & \text{on } \partial_p\Omega \\ \mathbf{u} = \bar{\mathbf{u}} & \text{on } \partial_u\Omega \end{cases} \quad (4.1)$$

where the following data is chosen

- simply connected cubic unit domain  $\Omega = [0, 1]^3$
- Lamé parameters  $\lambda = 1$  and  $\mu = 1$ , so that Young's modulus  $E = 2.5$  and Poisson ratio  $\nu = 0.25$
- homogeneous Dirichlet boundary conditions are enforced on the full boundary ( $\bar{\mathbf{u}} = \mathbf{0}$ ,  $\partial_u\Omega = \partial\Omega$ ,  $\partial_p\Omega = \emptyset$ )

- trigonometric body forces  $\mathbf{b}(\mathbf{x})$  in  $\Omega$

$$\mathbf{b}(x, y, z) = C \begin{Bmatrix} -\pi^2 [(\lambda + \mu) \cos(\pi x) \sin(\pi y + \pi z) - (\lambda + 4\mu) \sin(\pi x) \sin(\pi y) \sin(\pi z)] \\ -\pi^2 [(\lambda + \mu) \cos(\pi y) \sin(\pi x + \pi z) - (\lambda + 4\mu) \sin(\pi x) \sin(\pi y) \sin(\pi z)] \\ -\pi^2 [(\lambda + \mu) \cos(\pi z) \sin(\pi x + \pi y) - (\lambda + 4\mu) \sin(\pi x) \sin(\pi y) \sin(\pi z)] \end{Bmatrix}$$

where  $C$  is a constant which is taken to be 0.1.

So that problem (4.1) becomes

$$\begin{cases} -\mathbf{S}^T [\mathbf{D}(\mathbf{S}\mathbf{u})] = \mathbf{b} & \text{in } \Omega = [0, 1]^3 \\ \mathbf{u} = \mathbf{0} & \text{on } \partial\Omega \end{cases} \quad (4.2)$$

The body forces above  $\mathbf{b}$  are found by selecting a displacement field  $\mathbf{u}(\mathbf{x})$  satisfying the boundary conditions for the problem (4.2) and plugging such field in the first equation. The solver is then expected to find the solution  $\mathbf{u}$  which was specifically picked. For the present case the displacement  $\mathbf{u}(\mathbf{x})$ , solution of (4.2), is

$$\mathbf{u}(x, y, z) = C \sin(\pi x) \sin(\pi y) \sin(\pi z) \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix} \quad (4.3)$$

rendered in Figure 4.1.

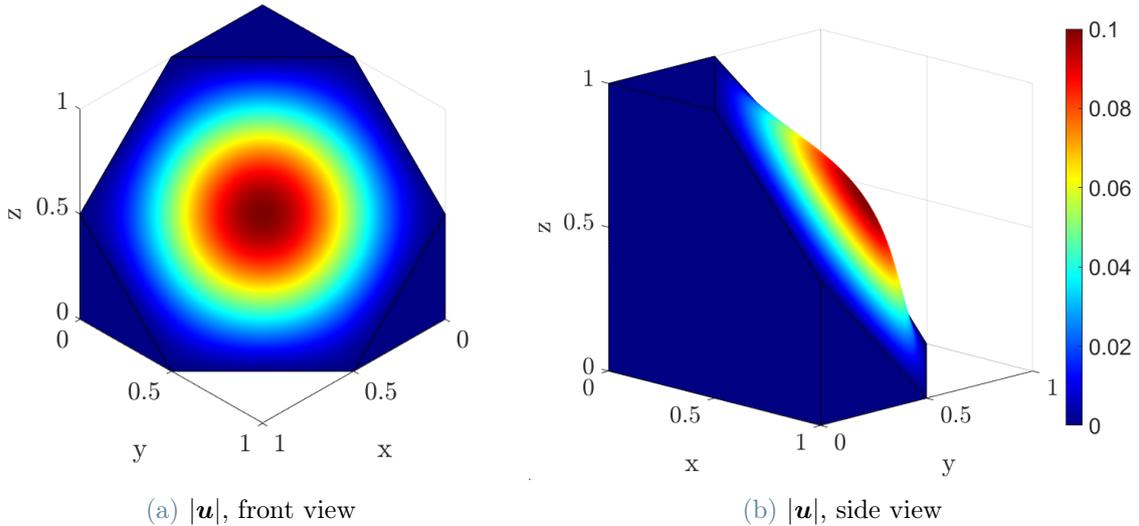


Figure 4.1: Exact solution of problem (4.2) shown in the deformed configuration where a cut has been performed to see the interior of the  $[1 \times 1 \times 1]$  domain. The color correspond to the displacement magnitude scalar field.

The corresponding strain field, shown in Figures 4.2, 4.3, 4.4, 4.5, 4.6, and 4.7 is given by

$$\boldsymbol{\varepsilon}(x, y, z) = C \begin{pmatrix} \pi \cos(\pi x) \sin(\pi y) \sin(\pi z) \\ \pi \sin(\pi x) \cos(\pi y) \sin(\pi z) \\ \pi \sin(\pi x) \sin(\pi y) \cos(\pi z) \\ \pi [\sin(\pi x) \cos(\pi y) \sin(\pi z) + \cos(\pi x) \sin(\pi y) \sin(\pi z)] \\ \pi [\sin(\pi x) \sin(\pi y) \cos(\pi z) + \sin(\pi x) \cos(\pi y) \sin(\pi z)] \\ \pi [\sin(\pi x) \sin(\pi y) \cos(\pi z) + \cos(\pi x) \sin(\pi y) \sin(\pi z)] \end{pmatrix} \quad (4.4)$$

which will be used to assess the error (4.6).

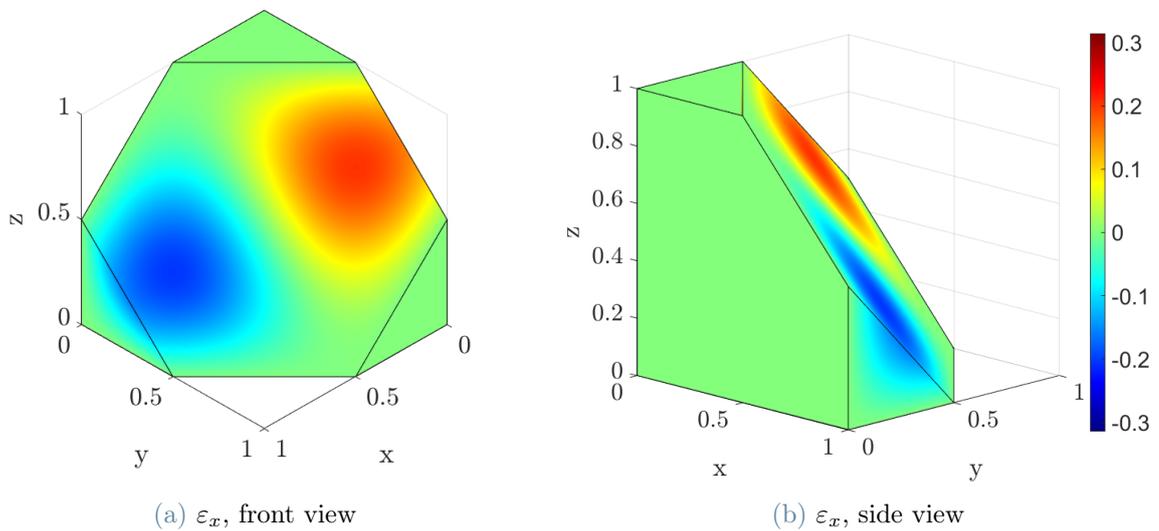


Figure 4.2: Exact  $\varepsilon_x$  strain of problem (4.2) shown in the reference configuration where a cut has been performed to see the interior of the  $[1 \times 1 \times 1]$  domain.

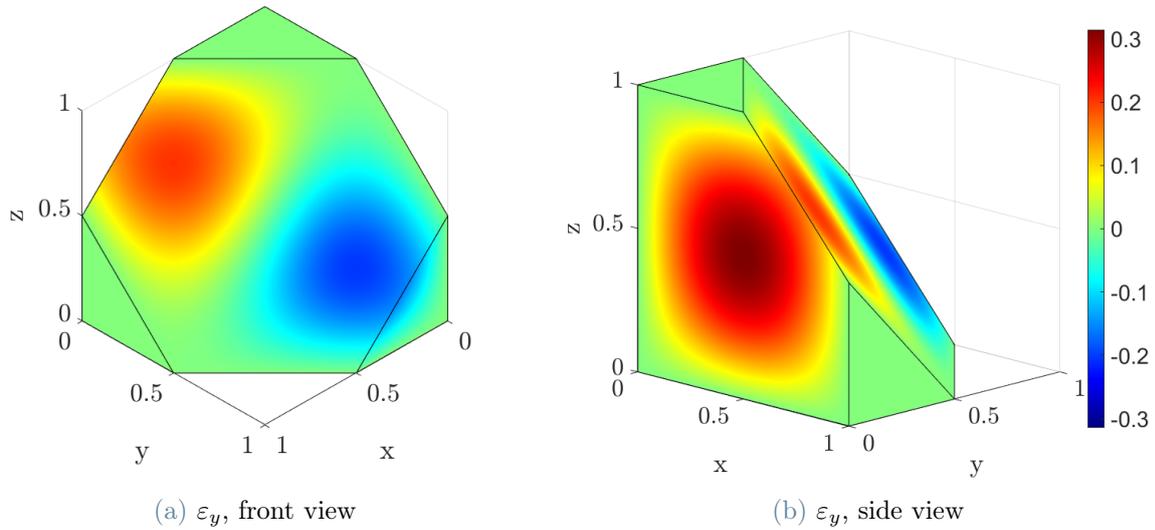


Figure 4.3: Exact  $\varepsilon_y$  strain of problem (4.2) shown in the reference configuration where a cut has been performed to see the interior of the  $[1 \times 1 \times 1]$  domain.

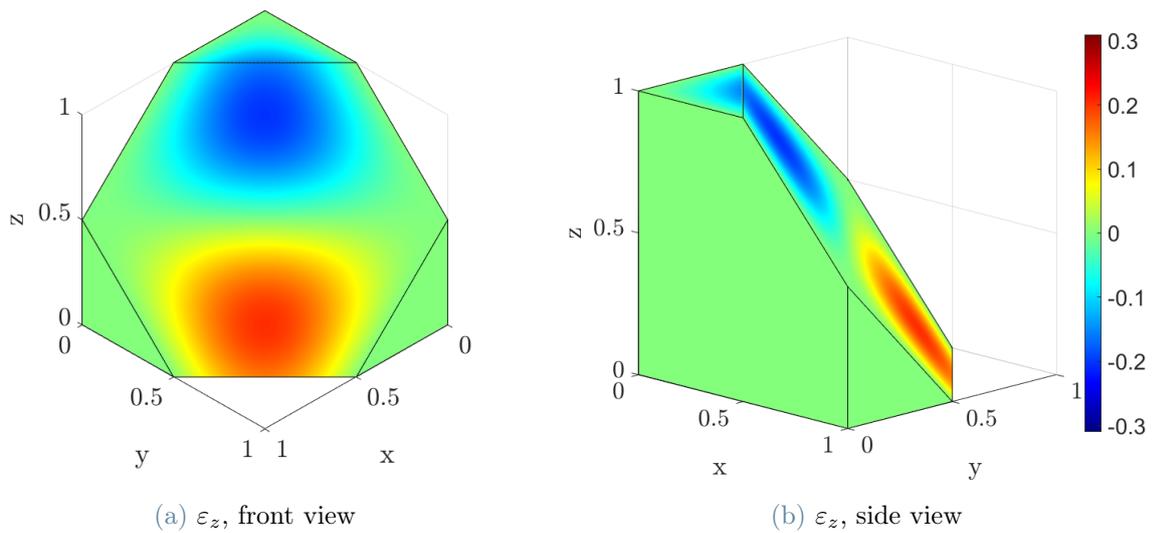


Figure 4.4: Exact  $\varepsilon_z$  strain of problem (4.2) shown in the reference configuration where a cut has been performed to see the interior of the  $[1 \times 1 \times 1]$  domain.

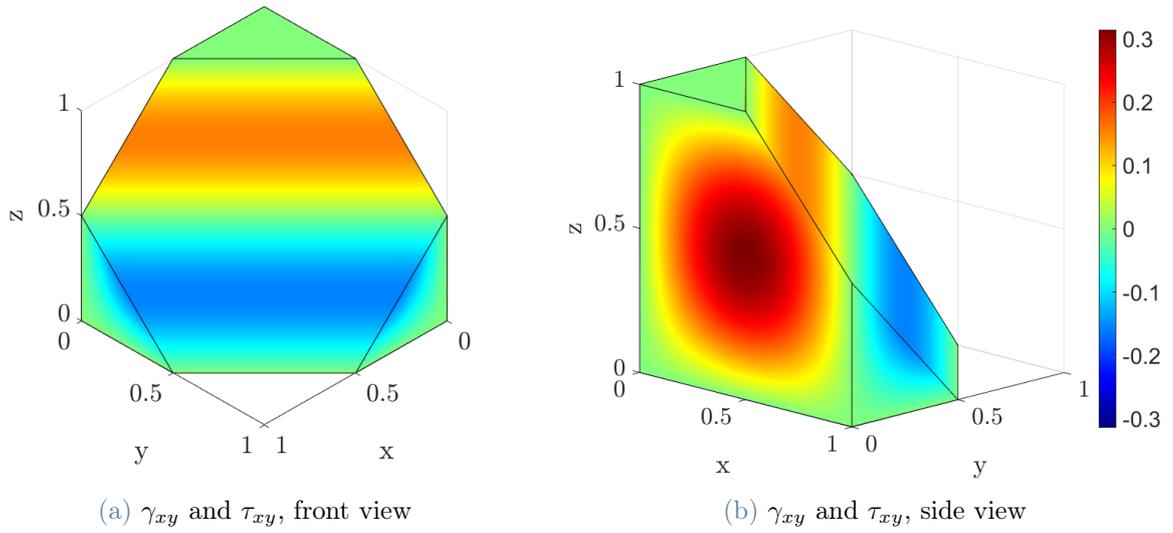


Figure 4.5: Exact  $\gamma_{xy}$  strain and  $\tau_{xy}$  stress of problem (4.2) shown in the reference configuration where a cut has been performed to see the interior of the  $[1 \times 1 \times 1]$  domain.

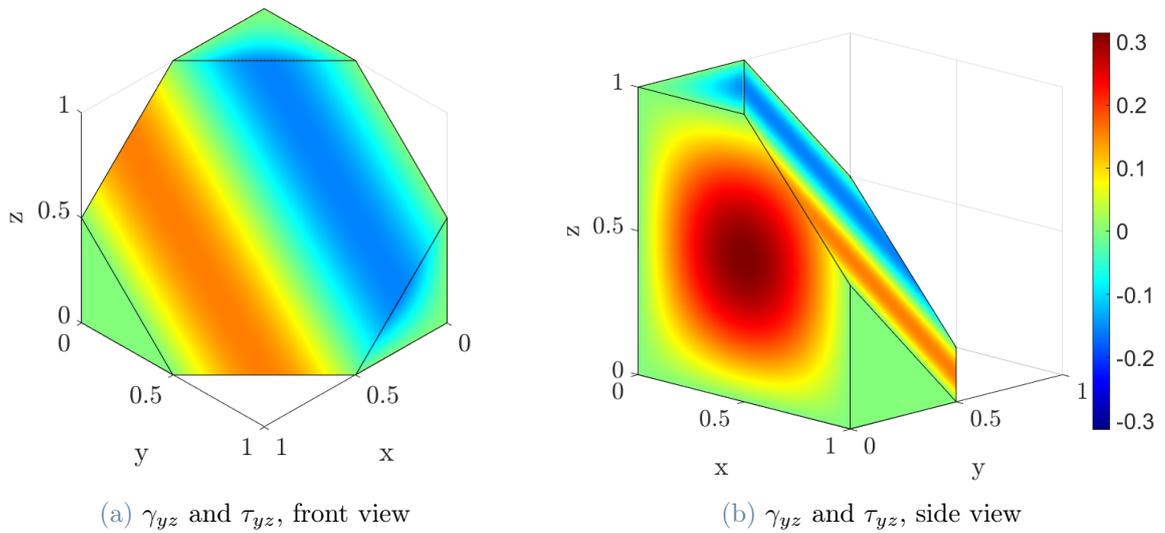


Figure 4.6: Exact  $\gamma_{yz}$  strain and  $\tau_{yz}$  stress of problem (4.2) shown in the reference configuration where a cut has been performed to see the interior of the  $[1 \times 1 \times 1]$  domain.

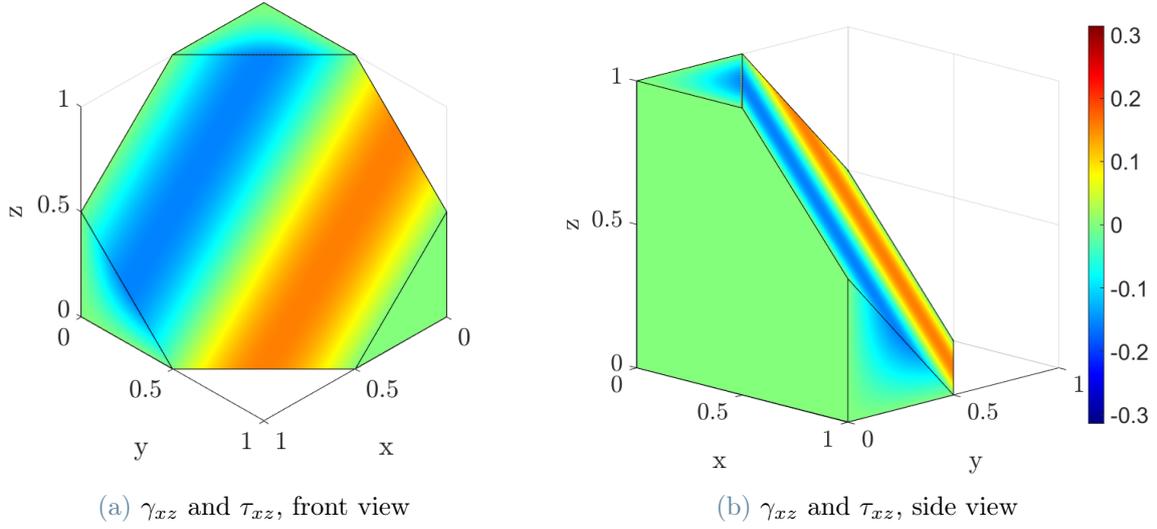


Figure 4.7: Exact  $\gamma_{xz}$  strain and  $\tau_{xz}$  stress of problem (4.2) shown in the reference configuration where a cut has been performed to see the interior of the  $[1 \times 1 \times 1]$  domain.

The exact stress field  $\boldsymbol{\sigma}$ , obtained by applying the material stiffness matrix  $\mathbf{D}$  to the strain field  $\boldsymbol{\varepsilon}$  is

$$\boldsymbol{\sigma}(x, y, z) = C \left\{ \begin{array}{l} \pi \left[ \begin{array}{l} 3 \cos(\pi x) \sin(\pi y) \sin(\pi z) + \\ + \sin(\pi x) \cos(\pi y) \sin(\pi z) + \\ + \sin(\pi x) \sin(\pi y) \cos(\pi z) \end{array} \right] \\ \pi \left[ \begin{array}{l} \cos(\pi x) \sin(\pi y) \sin(\pi z) + \\ + 3 \sin(\pi x) \cos(\pi y) \sin(\pi z) + \\ + \sin(\pi x) \sin(\pi y) \cos(\pi z) \end{array} \right] \\ \pi \left[ \begin{array}{l} \cos(\pi x) \sin(\pi y) \sin(\pi z) + \\ + \sin(\pi x) \cos(\pi y) \sin(\pi z) + \\ + 3 \sin(\pi x) \sin(\pi y) \cos(\pi z) \end{array} \right] \\ \pi [\sin(\pi x) \cos(\pi y) \sin(\pi z) + \cos(\pi x) \sin(\pi y) \sin(\pi z)] \\ \pi [\sin(\pi x) \sin(\pi y) \cos(\pi z) + \sin(\pi x) \cos(\pi y) \sin(\pi z)] \\ \pi [\sin(\pi x) \sin(\pi y) \cos(\pi z) + \cos(\pi x) \sin(\pi y) \sin(\pi z)] \end{array} \right. \quad (4.5)$$

The first three components are shown in Figures 4.8, 4.9 and 4.10, while the last three numerically coincide with the shear strains  $\boldsymbol{\gamma}$  of Figures 4.5, 4.6 and 4.7.

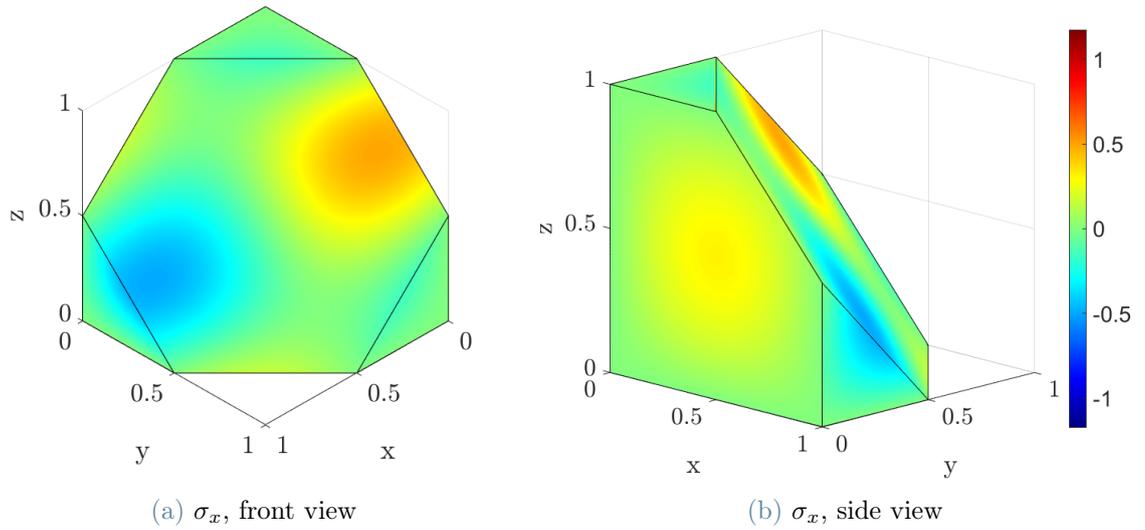


Figure 4.8: Exact  $\sigma_x$  stress of problem (4.2) shown in the reference configuration where a cut has been performed to see the interior of the  $[1 \times 1 \times 1]$  domain.

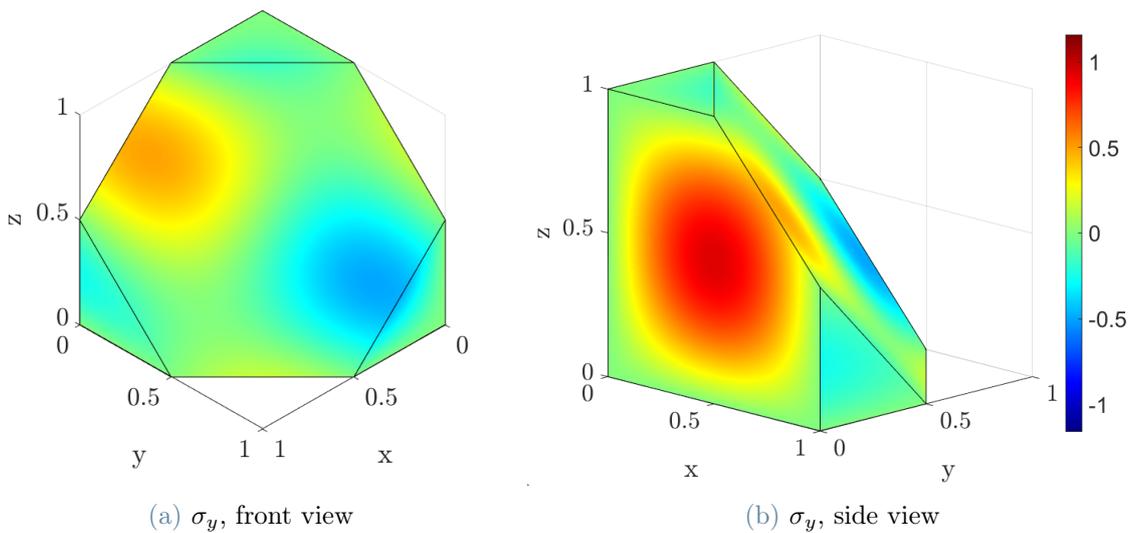


Figure 4.9: Exact  $\sigma_y$  stress of problem (4.2) shown in the reference configuration where a cut has been performed to see the interior of the  $[1 \times 1 \times 1]$  domain.

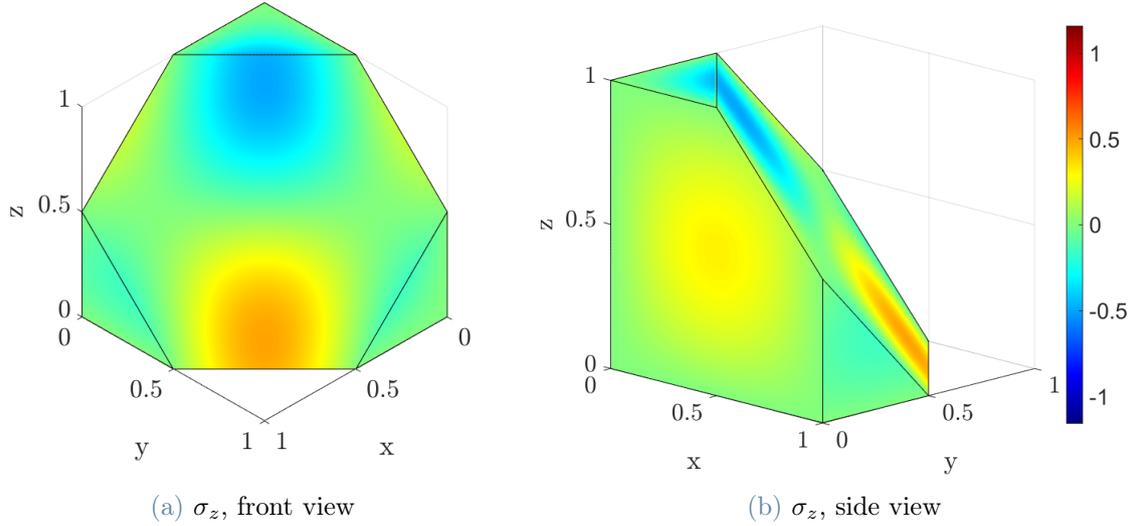


Figure 4.10: Exact  $\sigma_z$  stress of problem (4.2) shown in the reference configuration where a cut has been performed to see the interior of the  $[1 \times 1 \times 1]$  domain.

## 4.2. Meshes

Six different meshes are adopted for the computational domain of the problem and progressively refined to test the order of convergence.

- Tetrahedral mesh: the most common elemental geometry adopted in finite element meshers (Figures 4.11a, 4.12).
- Hexahedral mesh (Figure 4.13), made of *brick* elements (Figure 4.11b).
- Honeycomb-type mesh (Figure 4.14), made by elements obtained by extruding an hexagon (Figure 4.11c).
- Mesh made by truncated octahedra (Figure 4.15), i.e. elements obtained by removing the six pyramids corresponding to the six vertices of a regular octahedron. It has 14 faces (8 regular hexagons and 6 squares), 36 edges, and 24 vertices (Figure 4.11d).
- Mesh obtained by Voronoi tessellation (Figure 4.16) of the domain through random seeding the points generating the Voronoi cells (Figure 4.11e). A detailed overview of the method can be found in [52].
- Non-convex mesh (Figure 4.17), made by hexahedra where 2 additional vertices have been inserted in the centroid of two opposite faces and whose coordinates have been shifted, so to allow periodicity (Figure 4.11f).

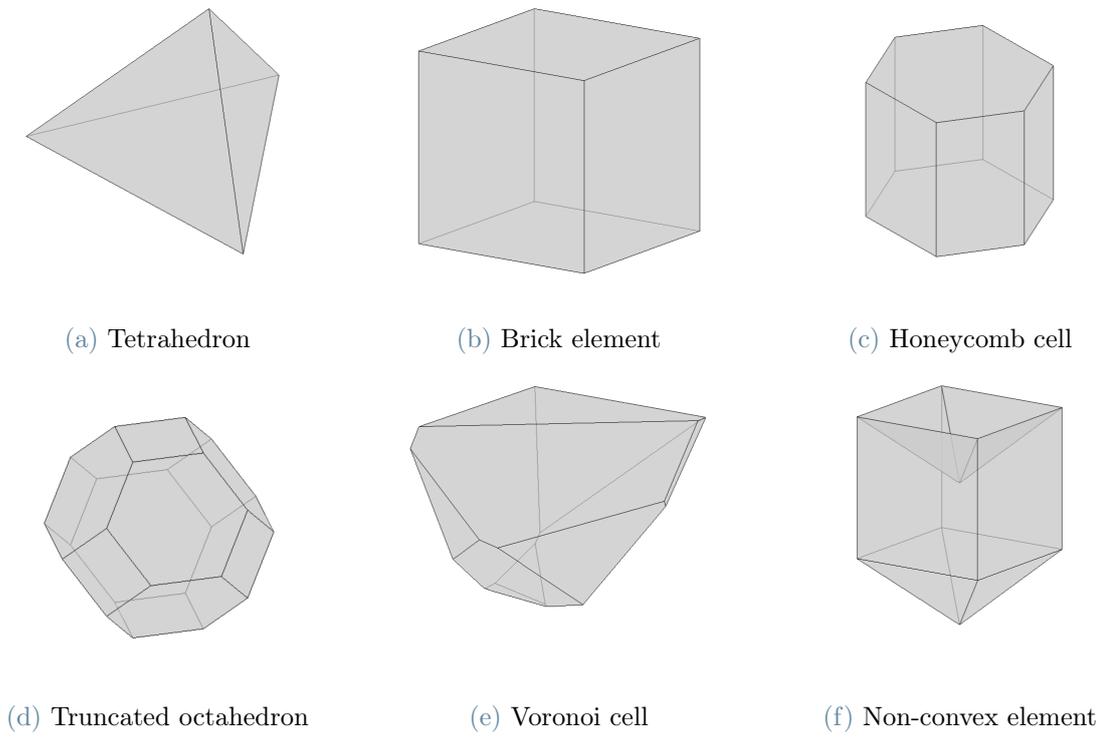


Figure 4.11: Polyhedral elements for the considered meshes of problem (4.2).

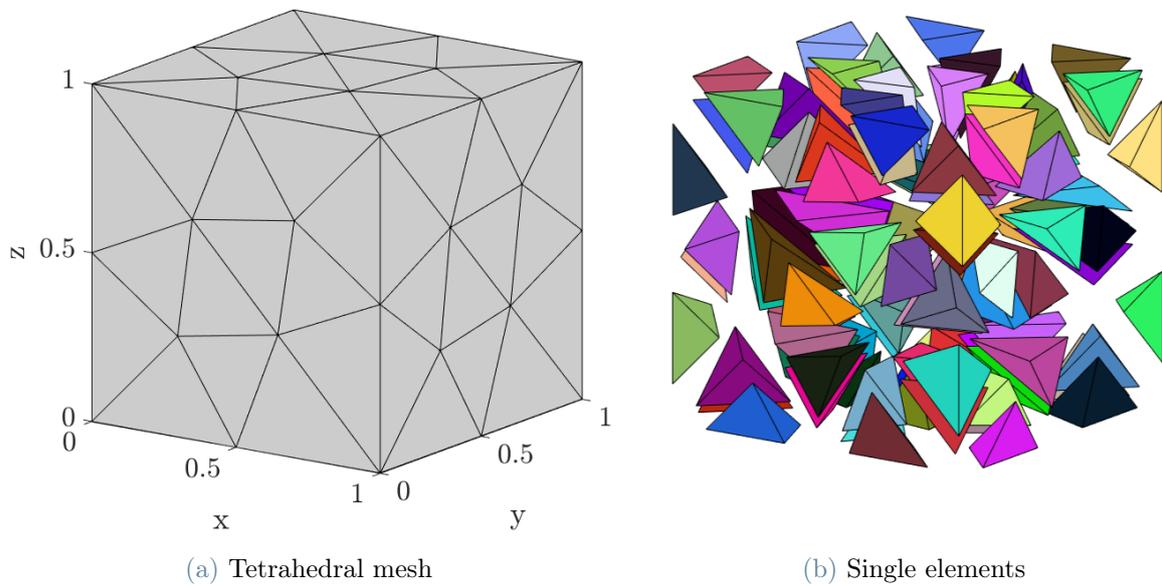


Figure 4.12: Tetrahedral mesh of the  $[1 \times 1 \times 1]$  domain.

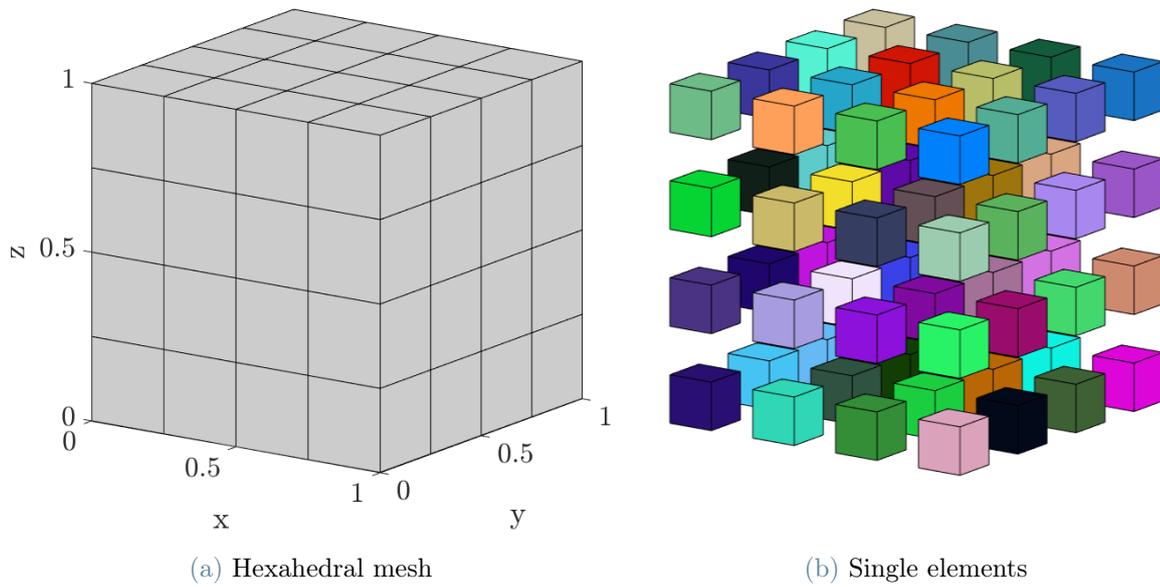


Figure 4.13: Hexahedral mesh of the  $[1 \times 1 \times 1]$  domain.

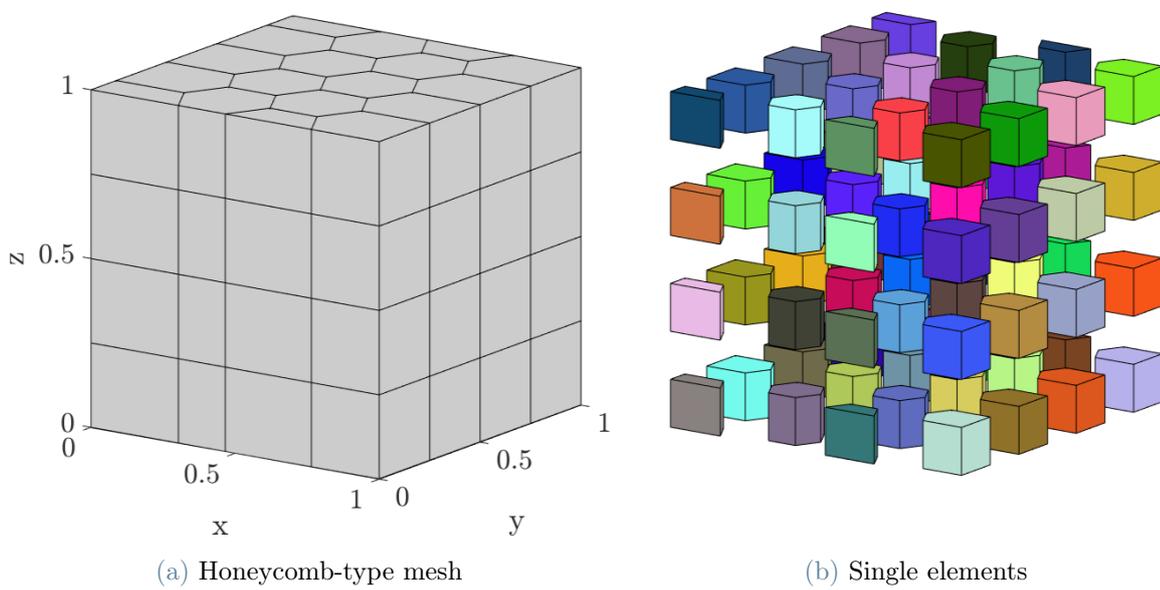


Figure 4.14: Honeycomb-type mesh of the  $[1 \times 1 \times 1]$  domain  $\Omega$ . Note that on the boundary  $\partial\Omega$  the elements have been cut to comply with the geometry.

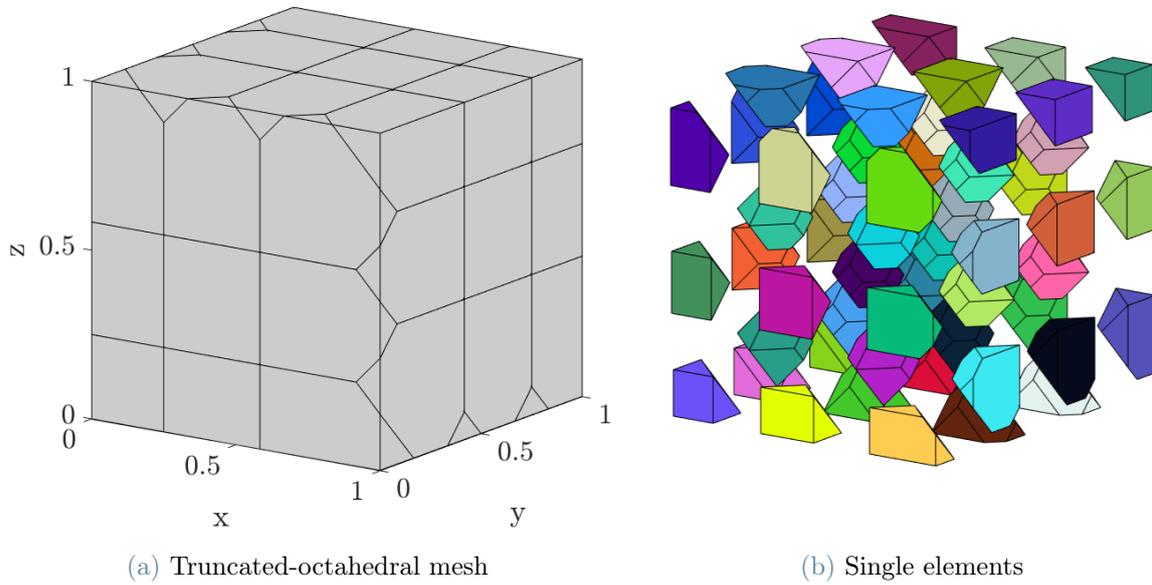


Figure 4.15: Truncated-octahedral mesh of the  $[1 \times 1 \times 1]$  domain  $\Omega$ . Note that on the boundary  $\partial\Omega$  the elements have been cut to comply with the geometry, so that the boundaries of the elements of Figure 4.11d are actually all contained in the interior of  $\Omega$ .

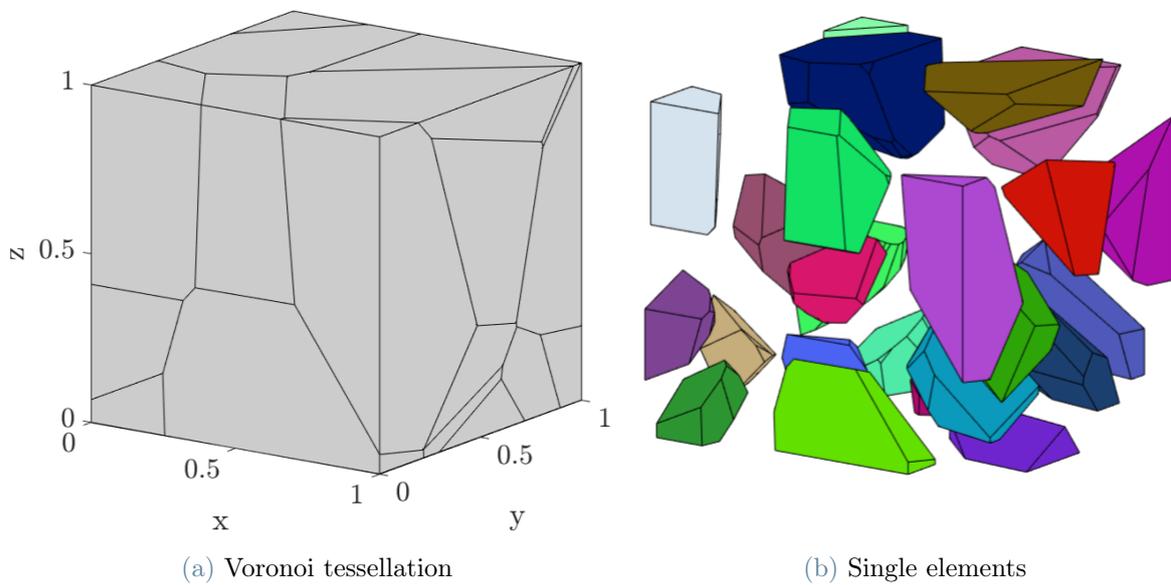


Figure 4.16: Mesh obtained by Voronoi tessellation of the  $[1 \times 1 \times 1]$  domain by random seeding the centers of the Voronoi cells.

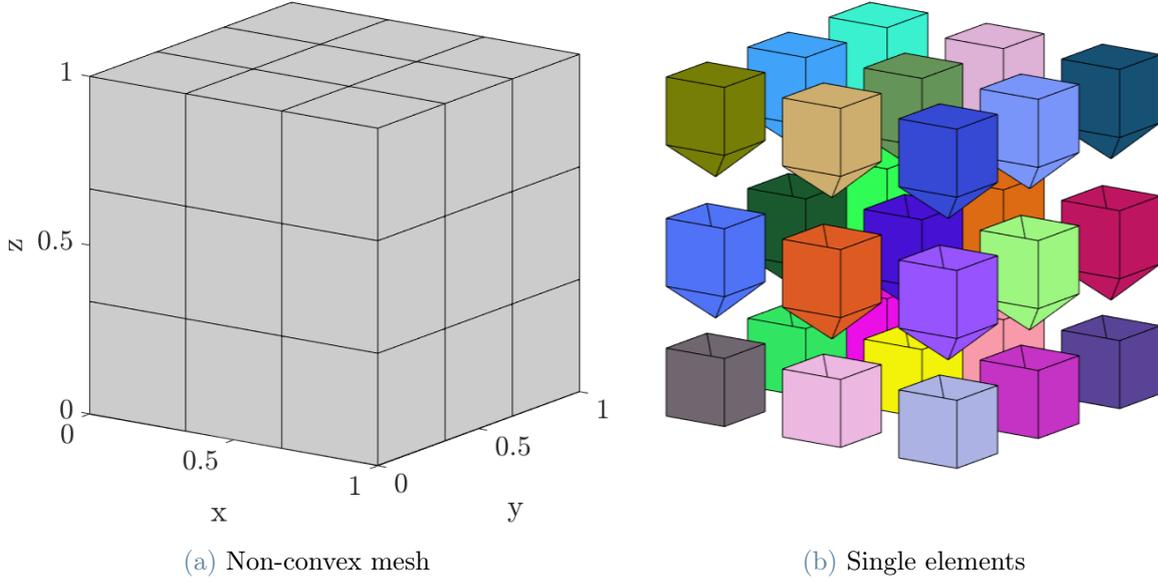


Figure 4.17: Non-convex mesh of the  $[1 \times 1 \times 1]$  domain.

### 4.3. h-refinement convergence tests

In this section h-refinement tests with known analytical solution will be conducted in a cubic unit domain  $\Omega = [0, 1]^3$  with various element types to show the robustness of the VEM. Other cases with different boundary conditions and body forces will be jointly presented with numerical results for *deltahedra* and *self-stabilizing* schemes proposed in Chapter 6. The convergence of the method will be evaluated in terms of the  $L^2$ -norm strain error, defined as

$$\|e_\varepsilon\|_{L^2} = \sqrt{\sum_{P \in \mathcal{P}} \int_P \|\varepsilon - \varepsilon^h\|^2 d\Omega} \quad (4.6)$$

under a progressive refinement of the mesh size  $h$ , obtained by averaging the element sizes  $h_P$  defined in (1.20), generally halved at each refinement step. The strain error in the  $L^2$ -norm computed according to (4.6) can be estimated with the following

$$\|e_\varepsilon\|_{L^2(\Omega)} = C(\mathbf{u}, k)h^k \quad (4.7)$$

where  $C(\mathbf{u}, k)$  is a constant depending only on the solution and on the order  $k$  of the VEM. Taking (4.7) for two different mesh sizes  $h_1$  and  $h_2$  leads to

$$\left. \begin{aligned} \|e_\varepsilon(h_1)\|_{L^2(\Omega)} &= C(\mathbf{u}, k)h_1^k \\ \|e_\varepsilon(h_2)\|_{L^2(\Omega)} &= C(\mathbf{u}, k)h_2^k \end{aligned} \right\} \implies \frac{\|e_\varepsilon(h_2)\|_{L^2(\Omega)}}{\|e_\varepsilon(h_1)\|_{L^2(\Omega)}} = \frac{C(\mathbf{u}, k)h_2^k}{C(\mathbf{u}, k)h_1^k} = \left(\frac{h_2}{h_1}\right)^k$$

and by taking the logarithm

$$k = \log_{\left(\frac{h_2}{h_1}\right)} \left[ \frac{\|e_\varepsilon(h_2)\|_{L^2(\Omega)}}{\|e_\varepsilon(h_1)\|_{L^2(\Omega)}} \right] = \frac{\log \left( \frac{\|e_\varepsilon(h_2)\|_{L^2(\Omega)}}{\|e_\varepsilon(h_1)\|_{L^2(\Omega)}} \right)}{\log \left( \frac{h_2}{h_1} \right)}$$

Therefore, it is convenient to plot the  $L^2$ -norm strain errors in a log-log plane as a function of the mesh size  $h$ , and check if the points align on a straight line whose slope corresponds to the order of the method  $k$ .

### 4.3.1. First order virtual elements

Figure 4.18 illustrates the convergence rates exhibited under  $h$ -refinement by the implemented virtual element program of order  $k = 1$  for the six meshes presented in Section 4.2.

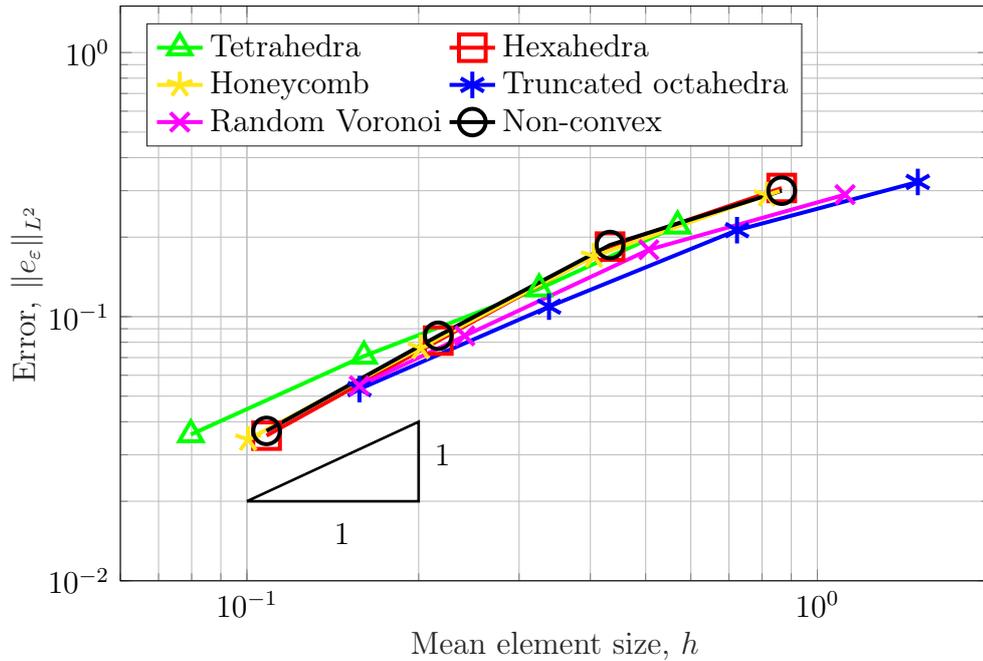


Figure 4.18:  $h$ -refinement convergence test for 1<sup>st</sup> order VEM.

All the convergence lines correctly align with the expected slope already with coarse meshes, showing an  $L^2$ -norm strain error  $\|e_\varepsilon\|_{L^2} = \mathcal{O}(h)$ .

For conciseness, the solved displacements are rendered for a few significant cases: two tetrahedral meshes (Figure 4.19), three hexahedral meshes (Figure 4.20) and two Voronoi tessellations (Figure 4.21). Moreover, stress contours are plotted for the finer Voronoi mesh (Figures 4.22 and 4.23), to allow a qualitative comparison with the exact fields

displayed in Figures 4.8, 4.9, 4.10, 4.5, 4.6 and 4.7. It should be noted that even though the elemental stress field is constant for  $k = 1$  VEM, the stress contours are smoothed inside each element from the nodal stress values obtained by volume averaging the field over the neighboring elements, as already mentioned in Section 1.3.6.

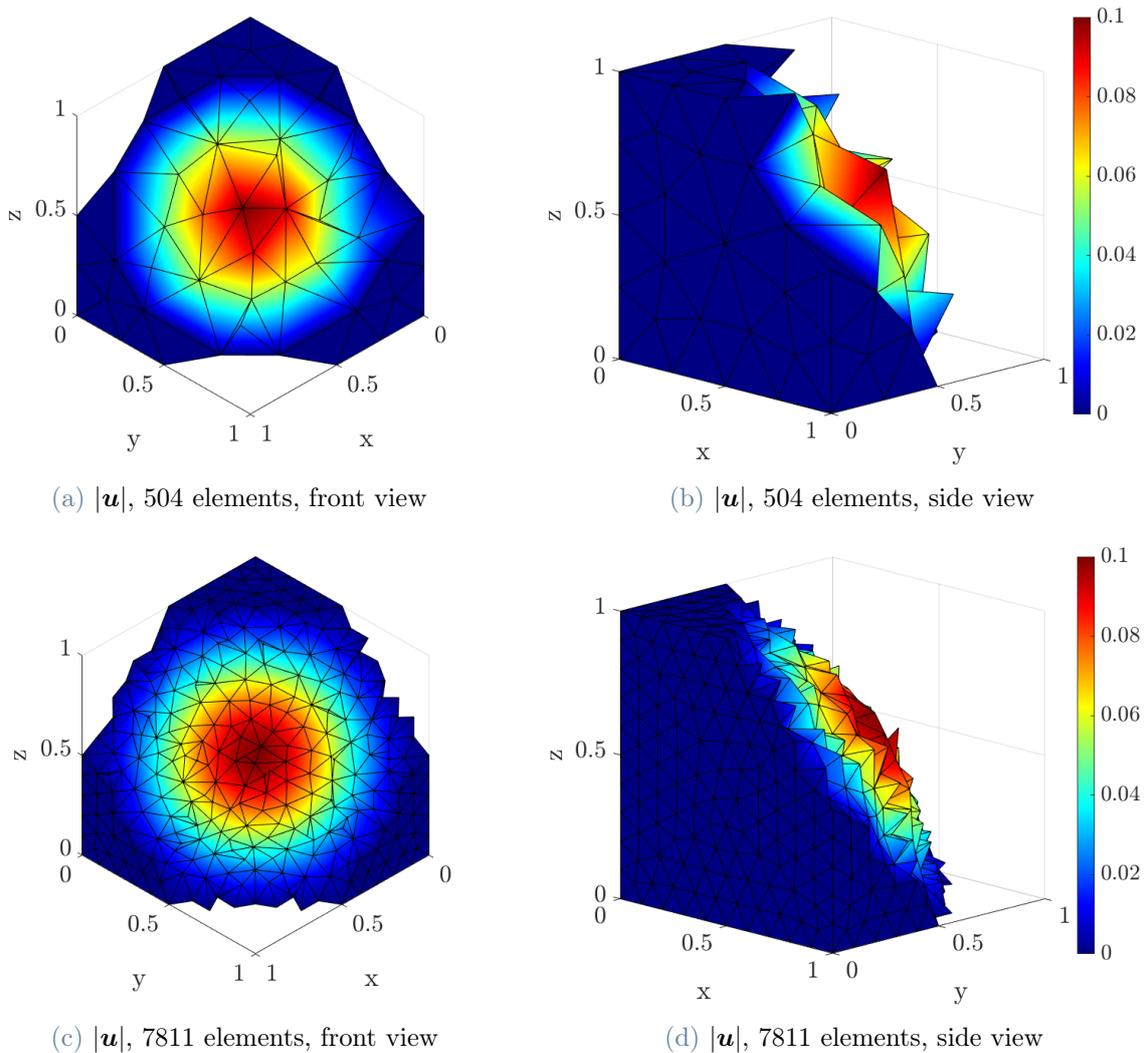


Figure 4.19: Numerical  $k = 1$  VEM solution of problem (4.2) shown in the deformed configuration where a cut has been performed to see the interior of the mesh, made by 504 (a, b) and 7811 (c, d) tetrahedra. The color represents the displacement magnitude.

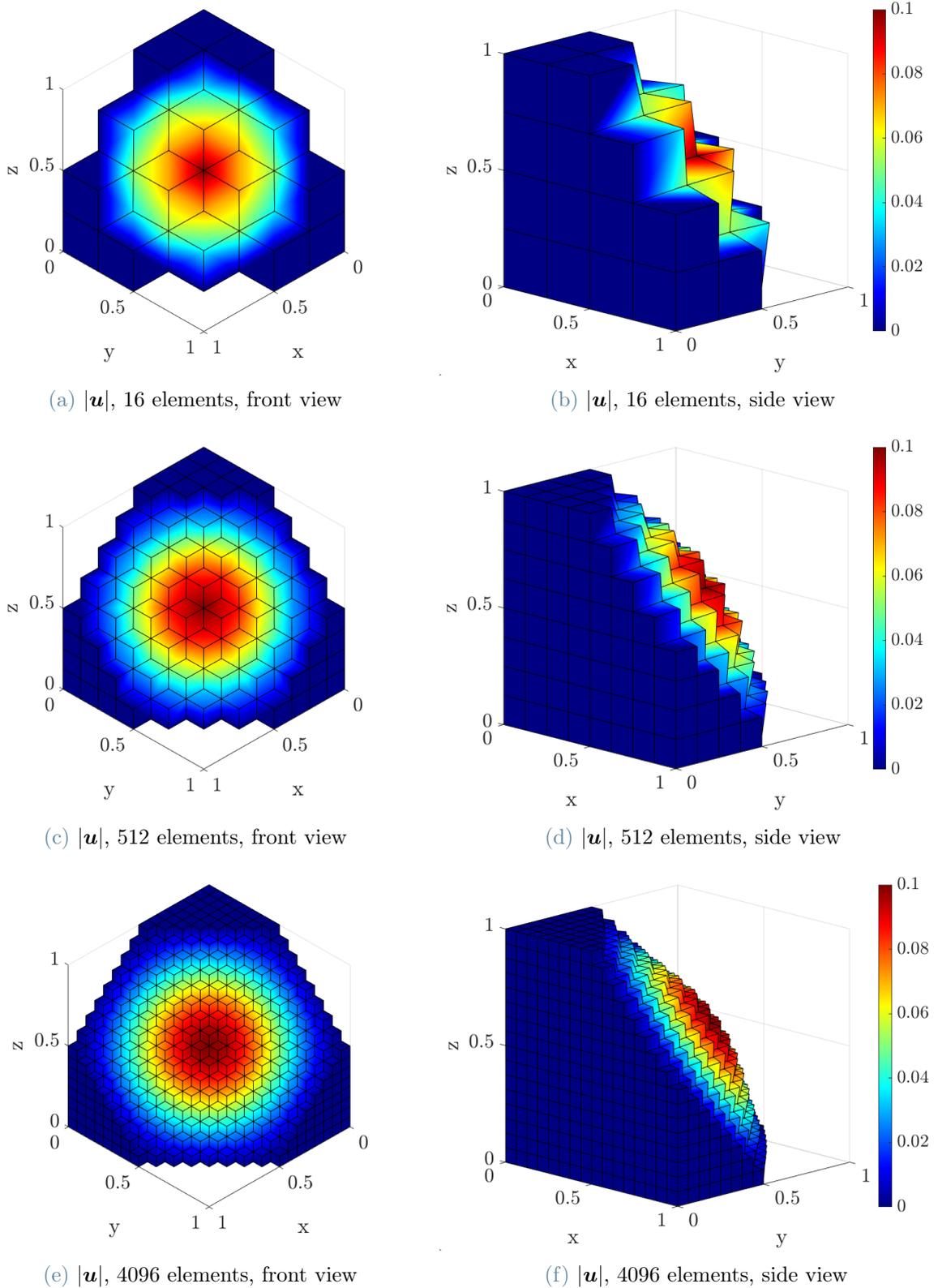


Figure 4.20: Numerical  $k = 1$  VEM solution of problem (4.2) shown in the deformed configuration where a cut allows to see the interior of the mesh, made by 16 (a, b), 512 (c, d) and 4096 (e, f) brick elements. The color represent the displacement magnitude.

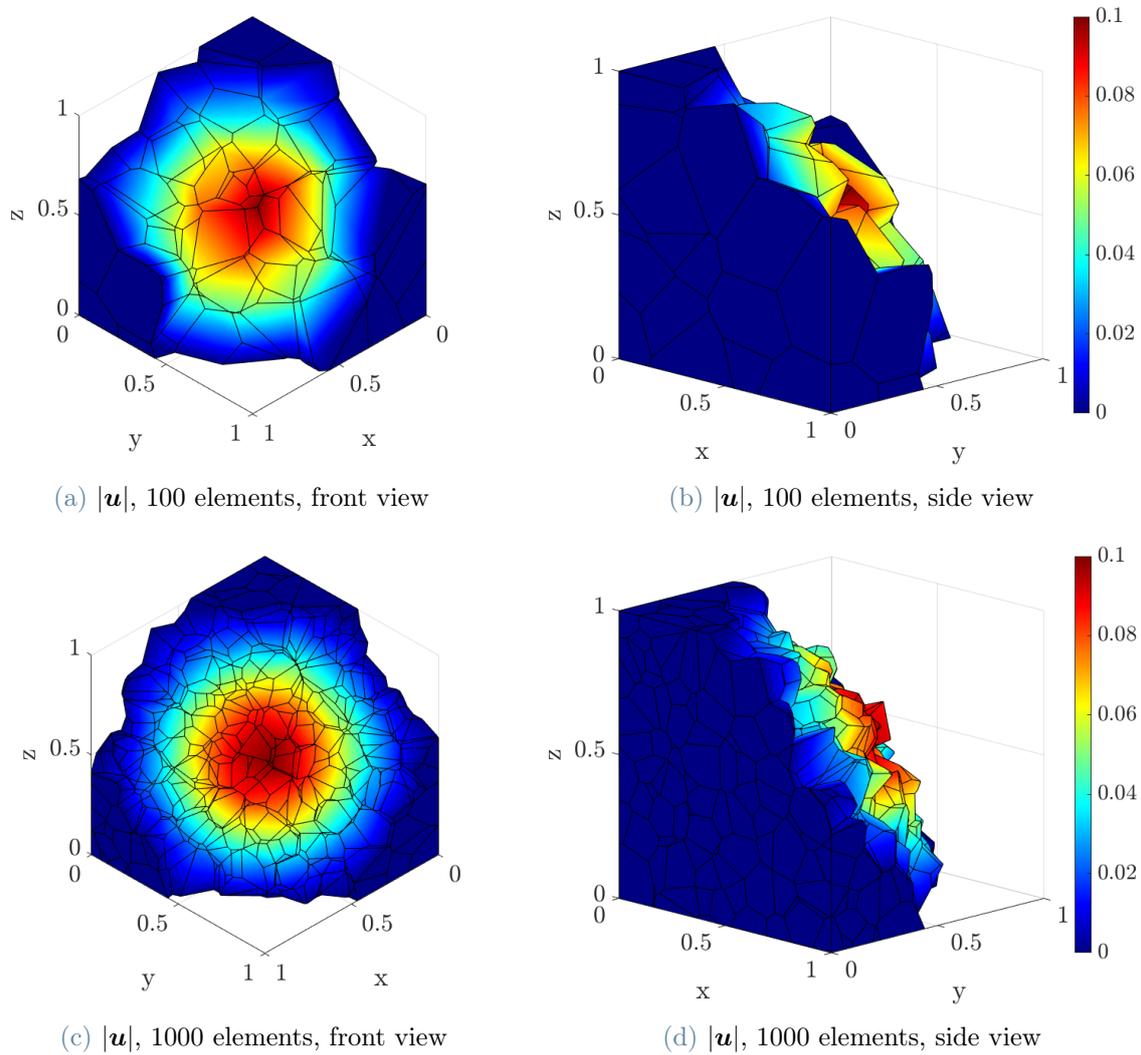


Figure 4.21: Numerical  $k = 1$  VEM solution of problem (4.2) shown in the deformed configuration where a cut has been performed to see the interior of the mesh, made by 100 (a, b) and 1000 (c, d) Voronoi cells. The color represents the displacement magnitude.

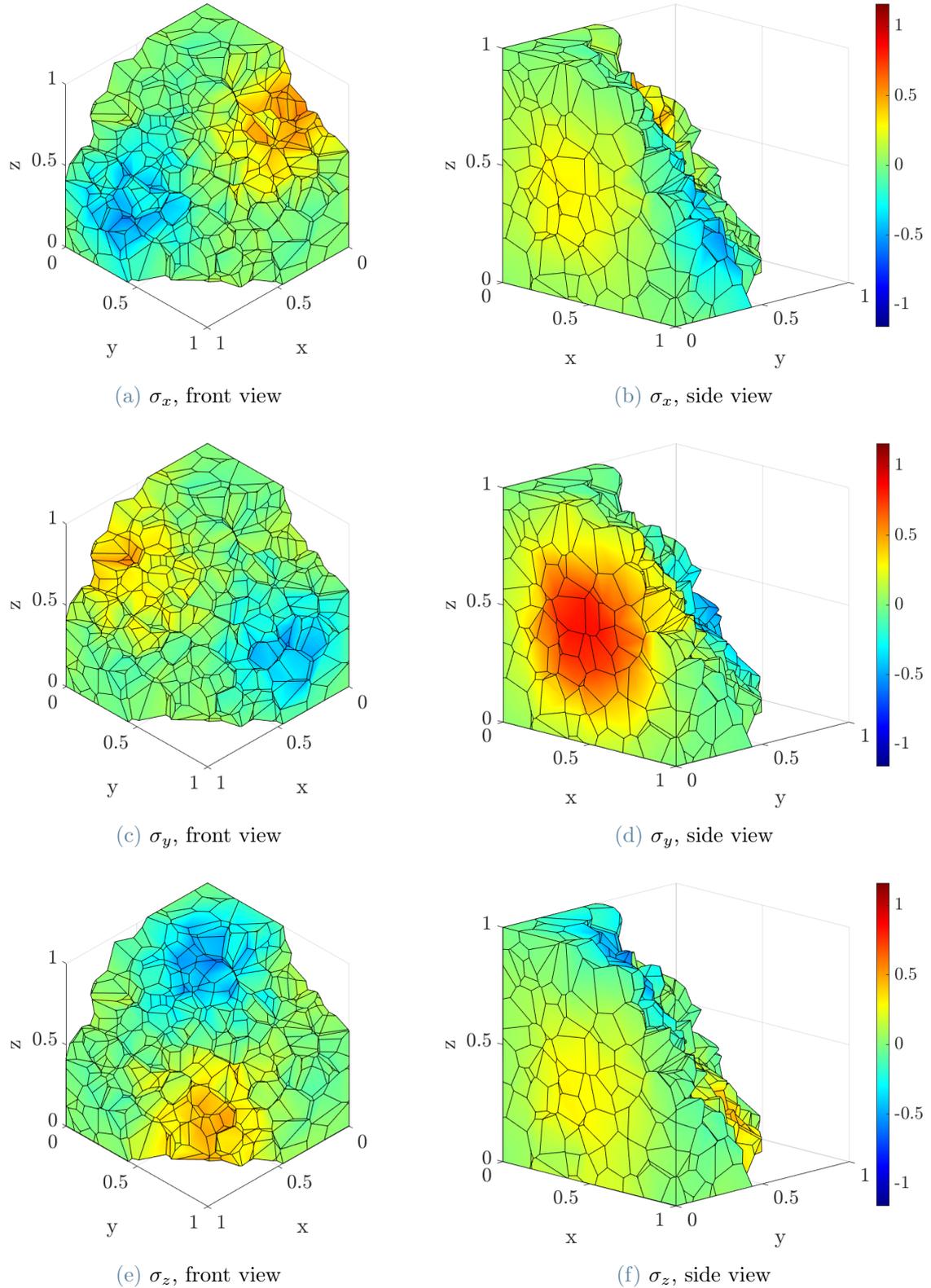


Figure 4.22: Numerical  $k = 1$  VEM normal stresses  $\sigma_x$  (a, b),  $\sigma_y$  (c, d) and  $\sigma_z$  (e, f) of problem (4.2) shown in the reference configuration where a cut allows to see the interior of the mesh, made by 1000 Voronoi cells.

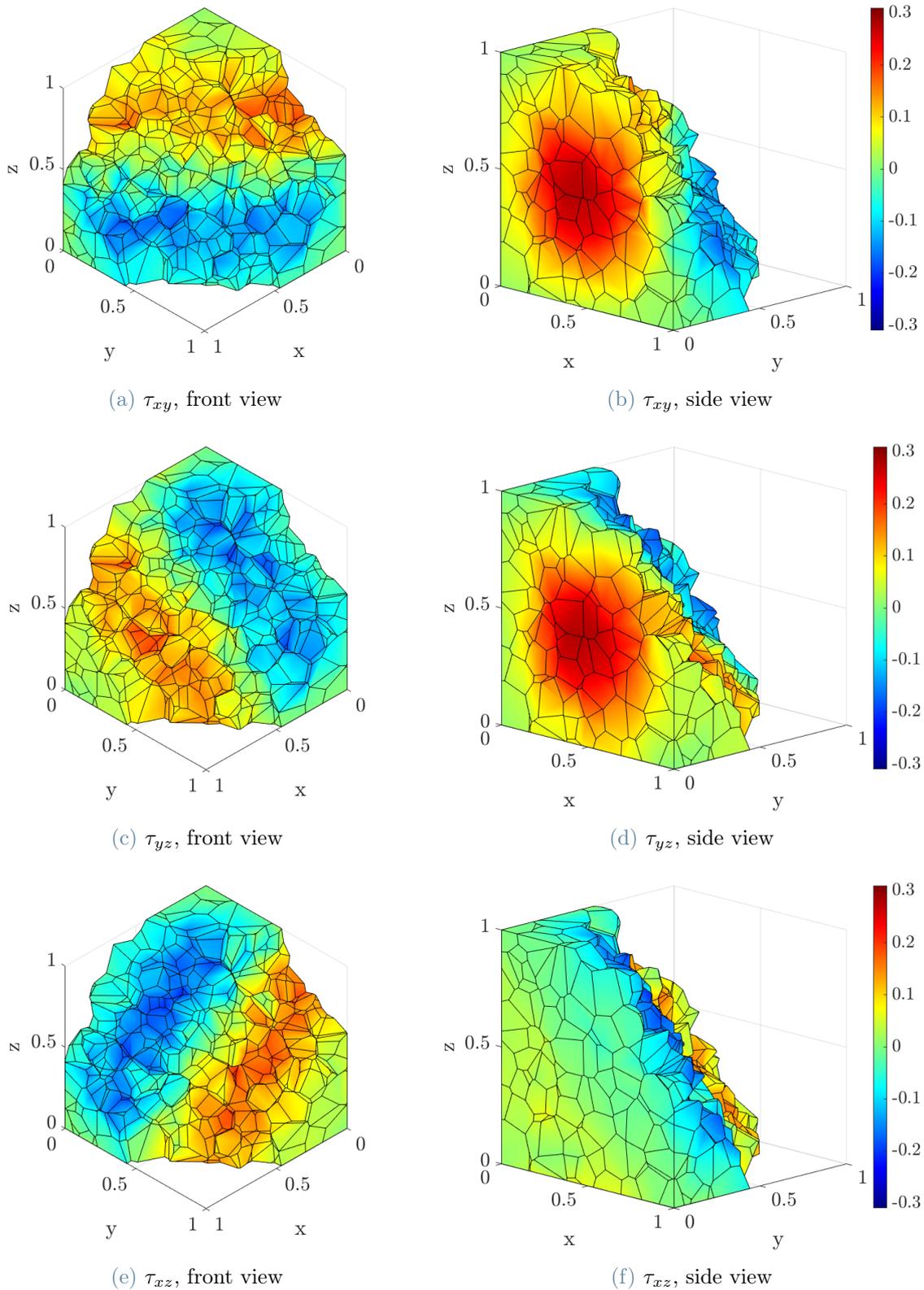


Figure 4.23: Numerical  $k = 1$  VEM shear stresses  $\tau_{xy}$  (a, b),  $\tau_{yz}$  (c, d) and  $\tau_{xz}$  (e, f) of problem (4.2) shown in the reference configuration where a cut allows to see the interior of the mesh, made by 1000 Voronoi cells.

### 4.3.2. Second order virtual elements

Figure 4.24 illustrates the convergence rates exhibited under  $h$ -refinement by the implemented virtual element program of order  $k = 2$  for the six meshes presented in Section 4.2.

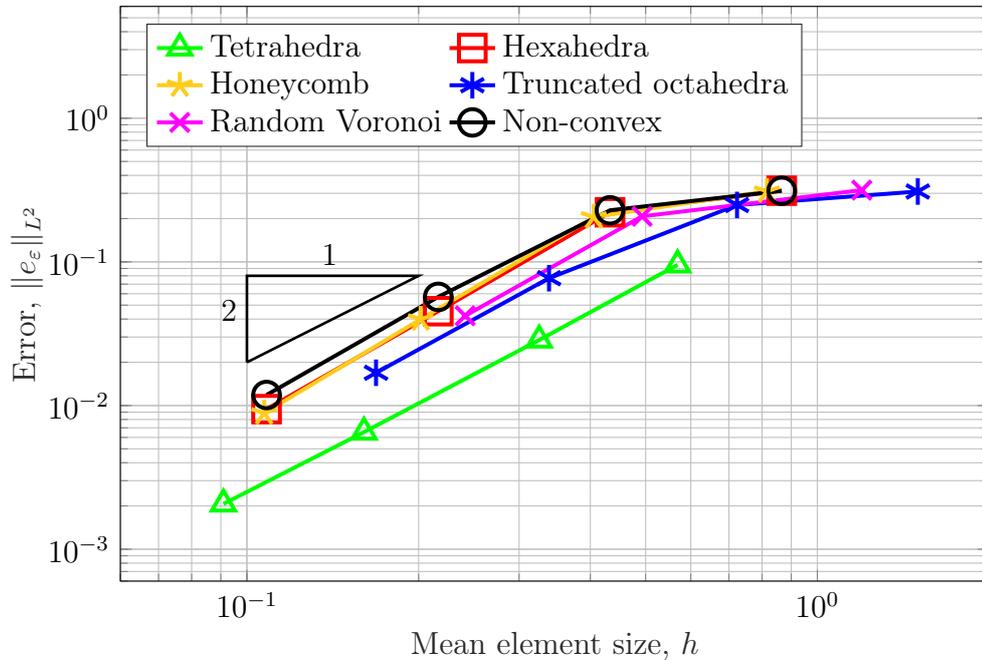


Figure 4.24:  $h$ -refinement convergence test for 2<sup>nd</sup> order VEM.

The convergence lines correctly align with the expected slope, showing an  $L^2$ -norm strain error  $\|e_\varepsilon\|_{L^2} = \mathcal{O}(h^2)$ . Further refinement of the mesh generated by a Voronoi tessellation through random seeding would confirm the correct slope already shown for such mesh, but the elements complexity indeed grows very fast, without significantly changing the mesh size  $h$  (elements with  $> 20$  faces are easily obtained).



# 5 | Deltahedra and enhanced strain VEM

This chapter presents a possible improvement of the standard virtual element method applied to three dimensional domains, in order to overcome two major drawbacks:

- the heavy additional computations required by the fact that the restrictions of the shape functions on the boundary of each element (its faces) are not polynomials as in 2D, where the boundary of each element is made by its edges, but belong to a proper virtual space;
- a stabilizing part is needed to suppress hourglass modes naturally present in the standard VEM consistent stiffness matrices, and even though many choices are available for its implementation, stabilization is often considered one of the bottleneck preventing virtual elements to be employed in commercial softwares.

As it is often the case, high order Galerkin techniques are not employed in common engineering applications, and  $h$ -refinements are always preferred when performing analyses. For this reason, an improvement of the first order standard VEM has been investigated, starting with the geometrical structure of the polyhedra.

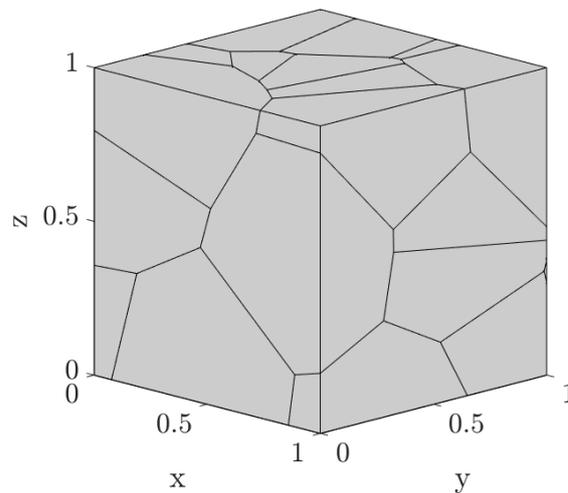
## 5.1. Deltahedral mesh first order VEM ( $\Delta$ VEM)

We recall from Definition 2.4 the 4<sup>th</sup> requirement for an element  $v$  to belong to the local virtual element space in  $\mathbb{R}^3$   $V_k(P)$ :

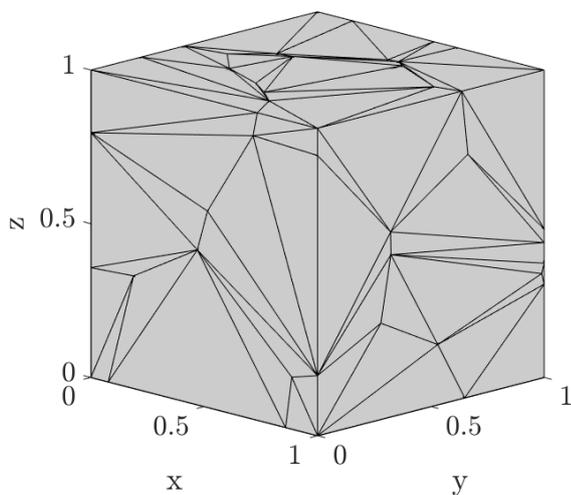
$$\text{for every face } F \text{ in } \partial P, v|_F \in W_k(F)$$

where  $W_k(F)$  is the enhanced local virtual element space in  $\mathbb{R}^2$  described in Section 2.4. As a consequence, the restriction on the faces of shape functions  $\varphi$  are also non-polynomial as  $\varphi|_F \in W_k(F)$ , leading to cumbersome projection computations necessary to assemble the boundary matrix  $\mathbf{A}_1$  explained in (3.13). However, for virtual elements of order  $k = 1$ , if the faces of the polyhedron are triangles instead of general polygons, projecting

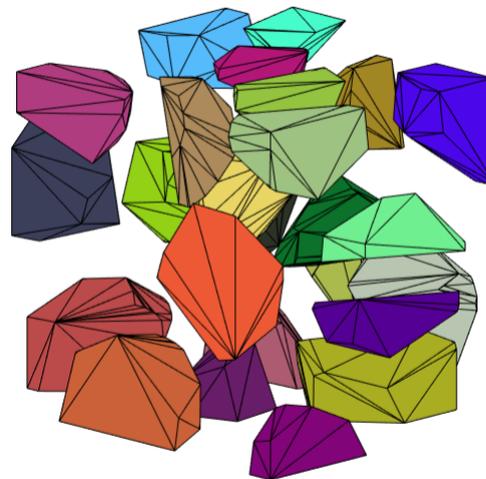
the restrictions of the shape functions over the faces is not necessary anymore as these become linear functions. Hence, *deltahedra* will be adopted, namely polyhedral elements whose faces are all triangles.



(a) Voronoi tessellation



(b) Deltahedral mesh



(c) Single deltahedra

Figure 5.1: Deltahedral mesh (b) obtained by triangulating the faces of a Voronoi tessellation (a) of the  $[1 \times 1 \times 1]$  domain.

### 5.1.1. Deltahedra

In this subsection, the properties of employing deltahedra are given and reasoned, starting with a simple, yet remarkable proposition.

**Proposition 5.1.** *Deltahedra property.* Given a virtual space  $V_1(F)$  and the corresponding enhanced virtual space  $W_1(F)$  of order  $k = 1$  embedded in a triangular domain  $F \in \mathbb{R}^2$ , the elements  $v \in V_1(F)$  and  $w \in W_1(F)$ :

- are linear polynomials in  $F$  ( $v, w \in \mathcal{P}_1(F)$ )
- coincide ( $v = w$ ).

*Proof.* For  $k = 1$ , taking an element  $v \in V_1(F)$  implies that

$$\Delta v \in \mathcal{P}_{-1} \implies \Delta v = 0$$

The values of  $v$  at the vertices are the complete set of degrees of freedom uniquely identifying an element in  $V_1(F)$  whose Laplacian is zero and whose value restricted on the boundary  $v|_{\partial F}$  is a piecewise linear, globally continuous function. If  $F$  is a triangle, the number of DOFs is three, corresponding to the three vertices, and they also uniquely identify a polynomial  $p \in \mathcal{P}_1(F)$ . Clearly, it holds  $v|_{\partial F} = p|_{\partial F}$ , and  $v$  being unique, together with  $\Delta p = 0$ , implies  $v = p$  in  $F$  as well. Hence,  $v$  is a linear polynomial and it obviously coincides with its polynomial projection  $\Pi_{F,1}^\nabla v$ .

Analogously, taking an element  $w \in W_1(F)$  implies that

$$\Delta w \in \mathcal{P}_1$$

The values of  $w$  at the vertices are the complete set of degrees of freedom uniquely identifying an element in  $W_1(F)$  whose value restricted on the boundary  $w|_{\partial F}$  is a piecewise linear, globally continuous function and with the additional conditions enforced by the enhanced space property (Theorem 2.1) to account for the fact that the Laplacian of  $w$  is a linear polynomial

$$\begin{aligned} \int_F w \, d\Sigma &= \int_F \Pi_{F,1}^\nabla w \, d\Sigma \\ \int_F w \xi_f \, d\Sigma &= \int_F [\Pi_{F,1}^\nabla w] \xi_f \, d\Sigma \\ \int_F w \eta_f \, d\Sigma &= \int_F [\Pi_{F,1}^\nabla w] \eta_f \, d\Sigma \end{aligned} \tag{5.1}$$

If  $F$  is a triangle, the polynomial projection  $\Pi_{F,1}^\nabla w$  evaluated at the DOFs assumes the same values as  $w$  evaluated at the DOFs, and consequently  $w|_{\partial F} = [\Pi_{F,1}^\nabla w]|_{\partial F}$ . Being  $w$  unique, we notice that taking  $w = \Pi_{F,1}^\nabla w$ , the three conditions (5.1) are obviously satisfied and  $\Delta w = 0 \in \mathcal{P}_1$ , hence  $w$  indeed equals its projection  $\Pi_{F,1}^\nabla w$ .

To prove that  $v$  coincide with  $w$  it is enough to remember from Section 2.4 that whenever

$v \in V_k(F)$  and  $w \in W_k(F)$  share the same values for the DOFs, then they have the exact same projection  $\Pi_{F,k}^\nabla v = \Pi_{F,k}^\nabla w$ . Hence

$$v = \Pi_{F,1}^\nabla v = \Pi_{F,1}^\nabla w = w$$

concluding the proof.  $\square$

In view of Proposition 5.1, the restrictions on the polyhedral boundary of the virtual shape functions  $\varphi|_F$  of order  $k = 1$  are also linear, and the projection  $\Pi_{F,1}^\nabla$  is not required anymore. The process of partitioning a polygonal area into a set of triangles, namely *polygon triangulation*, is well known and established in computational geometry and can be carried out for any simple polygon<sup>1</sup> with almost linear time complexity. The existence of such triangulation is guaranteed by the *two ears theorem*, here briefly reported from [47].

**Theorem 5.1.** *Two ears theorem. Every simple polygon  $F$  that is not itself a triangle has at least two ears, i.e. two triangles made by three consecutive vertices  $V_1, V_2$  and  $V_3$  such that the edge  $\overline{V_1V_3}$  is entirely contained in  $F$ .*

Removing one ear from a given simple polygon  $F_0$  produces another simple polygon  $F_1$ , which could be further reduced iterating the procedure, called *ear clipping method* (Figure 5.2). The method allows to partition a polygon with  $n$  vertices in  $(n - 2)$  triangles without adding extra points in  $\mathcal{O}(n^2)$  time. Faster algorithms guaranteeing that no points are added exhibit  $\mathcal{O}(n \log^* n)$  time ([50]). If the polygon is convex, triangulation is trivial as it can be obtained by adding diagonals from a randomly picked vertex to its non-neighboring vertices (*fan triangulation*, Figure 5.3).

---

<sup>1</sup>A simple polygon is a polygon which does not intersect itself and has no holes.

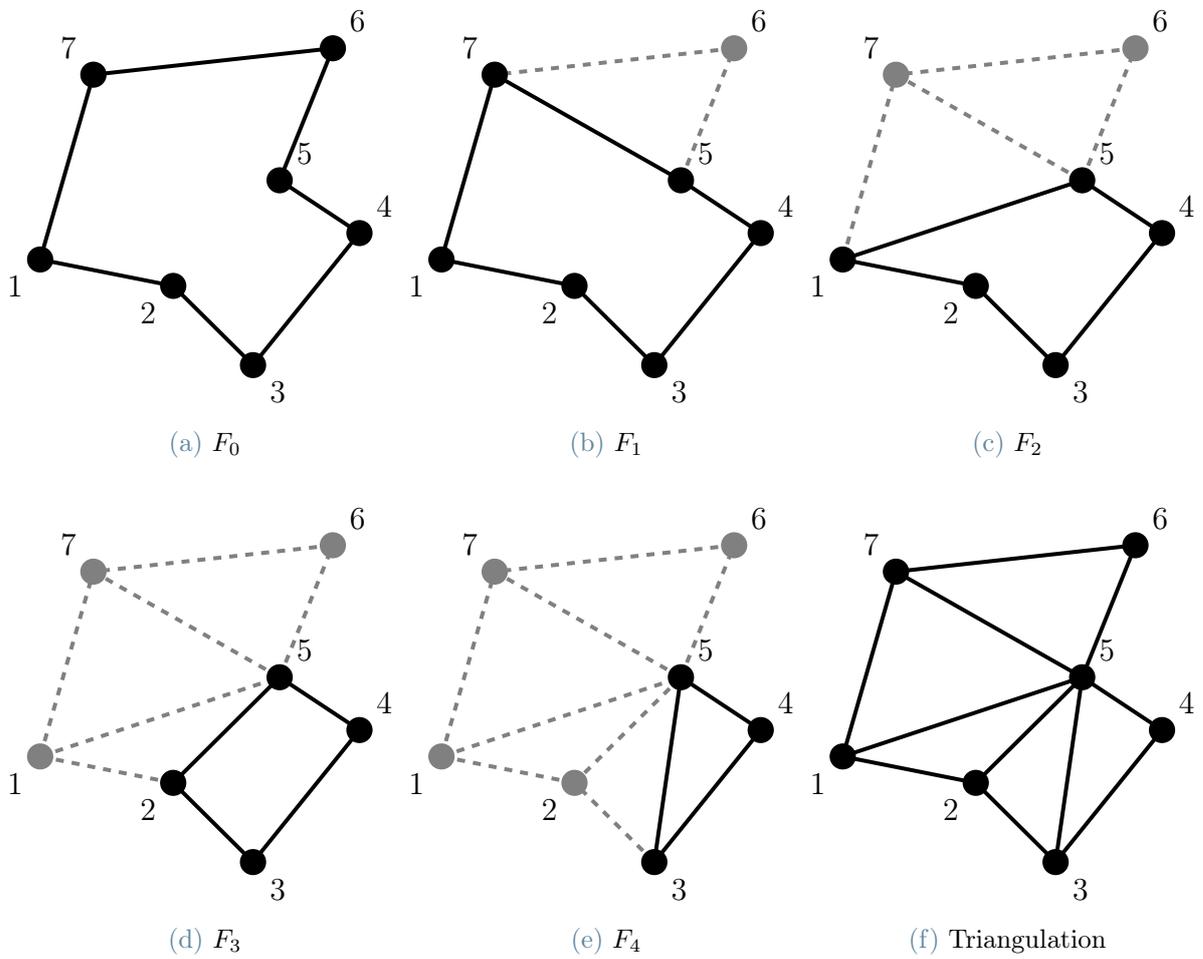


Figure 5.2: A possible triangulation of a simple polygon via the ear clipping method. At each iteration  $i$ , an ear is removed from the polygon  $F_{i-1}$ , obtaining a new simple polygon  $F_i$ , until one single triangle is obtained. Notice that the triangulation shown is not unique.

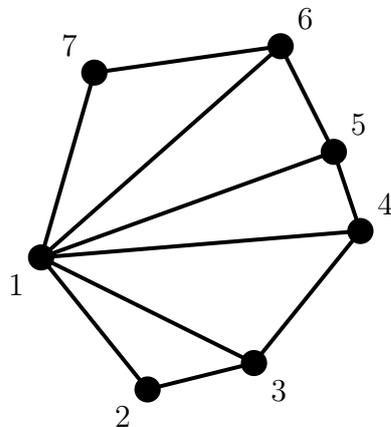


Figure 5.3: Fan triangulation of a convex polygon.

### 5.1.2. Summary of the $\Delta$ VEM scheme

The advantages of adopting deltahedra for VEM of order  $k = 1$  are evident below, where the VEM matrices presented in Chapter 4 are explicitly reported and adapted for these peculiar elements.

- The displacement model contains only vertex-type DOFs, as with standard VEM of order  $k = 1$

$$\mathbf{N}_u = \begin{bmatrix} N_1^u & 0 & 0 & N_2^u & 0 & 0 & \dots & N_{N_V}^u & 0 & 0 \\ 0 & N_1^u & 0 & 0 & N_2^u & 0 & \dots & 0 & N_{N_V}^u & 0 \\ 0 & 0 & N_1^u & 0 & 0 & N_2^u & \dots & 0 & 0 & N_{N_V}^u \end{bmatrix}$$

- The strain model is described by the usual  $[6 \times 6]$  identity matrix for VEM of order  $k = 1$

$$\mathbf{N}_\varepsilon = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- Matrix  $\mathbf{G}$  reduces to the  $[6 \times 6]$  identity matrix multiplied by the volume of the element  $|P|$ , as with VEM of order  $k = 1$

$$\mathbf{G} = \int_P \mathbf{N}_\varepsilon^T \mathbf{N}_\varepsilon d\Omega = \int_P \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} d\Omega = |P| \mathbf{I}$$

- Matrix  $\mathbf{A}_1$  greatly simplifies

$$\begin{aligned} \mathbf{A}_1 &= \sum_{F \in \partial P} \left[ \int_F (\mathbb{N}_F \mathbf{N}_\varepsilon)^\top \mathbf{N}_u d\Sigma \right] = \sum_{F \in \partial P} \left[ \int_F \left[ \mathbb{N}_F N_1^u \quad \mathbb{N}_F N_2^u \quad \dots \quad \mathbb{N}_F N_{N_V}^u \right] d\Sigma \right] = \\ &= \sum_{F \in \partial P} \left[ \int_F \begin{bmatrix} n_x N_1^u & 0 & 0 & \dots & n_x N_{N_V}^u & 0 & 0 \\ 0 & n_y N_1^u & 0 & \dots & 0 & n_y N_{N_V}^u & 0 \\ 0 & 0 & n_z N_1^u & \dots & 0 & 0 & n_z N_{N_V}^u \\ n_y N_1^u & n_x N_1^u & 0 & \dots & n_y N_{N_V}^u & n_x N_{N_V}^u & 0 \\ 0 & n_z N_1^u & n_y N_1^u & \dots & 0 & n_z N_{N_V}^u & n_y N_{N_V}^u \\ n_z N_1^u & 0 & n_x N_1^u & \dots & n_z N_{N_V}^u & 0 & n_x N_{N_V}^u \end{bmatrix} d\Sigma \right] \end{aligned}$$

where each entry can be easily computed exploiting Lagrangian-type interpolation, the fact that the shape functions are linear on  $F$  and  $F$  being a triangle

$$\int_F n_i N_j^u d\Sigma = \begin{cases} n_i \frac{|F|}{3} & \text{if vertex } j \text{ is on face } F \\ 0 & \text{otherwise} \end{cases}$$

The coefficient  $\frac{|F|}{3}$  accounts for exact integration over  $F$  of the linear restriction of the shape function  $N_j^u|_F$

- Matrix  $\mathbf{A}_2$  is null as for standard VEM of order  $k = 1$

$$\mathbf{A}_2 = \mathbf{0}$$

- Matrix  $\mathbf{E}$  reduces to the material stiffness matrix multiplied by the volume of the element  $|P|$  as it happens in standard VEM of order  $k = 1$

$$\mathbf{E} = \int_P \mathbf{N}_\varepsilon^\top \mathbf{D} \mathbf{N}_\varepsilon d\Omega = |P| \mathbf{D}$$

- The matrix  $\mathbf{T}_{D+R}$  gathering deformative and rigid body modes reduces to the fol-

lowing  $[3N_V \times 12]$  matrix as in standard VEM of order  $k = 1$

$$\mathbf{T}_{D+R} = \begin{bmatrix} 1 & 0 & 0 & \xi_1 & 0 & 0 & \eta_1 & 0 & 0 & \zeta_1 & 0 & 0 \\ 0 & 1 & 0 & 0 & \xi_1 & 0 & 0 & \eta_1 & 0 & 0 & \zeta_1 & 0 \\ 0 & 0 & 1 & 0 & 0 & \xi_1 & 0 & 0 & \eta_1 & 0 & 0 & \zeta_1 \\ 1 & 0 & 0 & \xi_2 & 0 & 0 & \eta_2 & 0 & 0 & \zeta_2 & 0 & 0 \\ 0 & 1 & 0 & 0 & \xi_2 & 0 & 0 & \eta_2 & 0 & 0 & \zeta_2 & 0 \\ 0 & 0 & 1 & 0 & 0 & \xi_2 & 0 & 0 & \eta_2 & 0 & 0 & \zeta_2 \\ \vdots & \vdots \\ 1 & 0 & 0 & \xi_{N_V} & 0 & 0 & \eta_{N_V} & 0 & 0 & \zeta_{N_V} & 0 & 0 \\ 0 & 1 & 0 & 0 & \xi_{N_V} & 0 & 0 & \eta_{N_V} & 0 & 0 & \zeta_{N_V} & 0 \\ 0 & 0 & 1 & 0 & 0 & \xi_{N_V} & 0 & 0 & \eta_{N_V} & 0 & 0 & \zeta_{N_V} \end{bmatrix}$$

where  $\{\xi_i \ \eta_i \ \zeta_i\}^T$  are the coordinates of vertex  $V_i$

- The local equivalent nodal forces vector  $\mathbf{F}_e^b$  is computed as for standard VEM for  $k = 1$  following (3.27)

The above scheme highlights how convenient the adoption of deltahedra is, from both the implementation and computation point of view.

## 5.2. Hu-Washizu-based enhanced strain VEM

The second drawback of the virtual element method concerns the requirement of stabilization. As mentioned in the introduction of this chapter, the need of a stabilizing part in the stiffness matrix is seen as one of the major weaknesses of the VEM. Moreover, adding a fictitious stiffness to suppress hourglass modes can result in completely wrong results in the very unlucky cases where the forcing term activates exactly those modes. Indeed, even though under  $h$ -refinement the approximate solution would still converge to the exact one, with coarse meshes the VEM would produce a solution which depend on the freedom in choosing the coefficient for stabilization (e.g., the usual  $\frac{1}{2}$  for scalar-based stabilization (1.65) or  $\alpha_0$  in diagonal matrix-based stabilization (1.66)). This section discuss a possible way of intrinsically suppressing hourglass modes in the consistent part of the stiffness matrix, hence avoiding the need of stabilization. The general idea comes from augmenting the strain parameters as in [21], but applying the concept to virtual elements instead of finite elements, as successfully done for the two dimensional case in [42] and [43]. However, in three dimensions the proper matching between strain and displacement parameters becomes more subtle and harder to achieve, as will be clear in the

next section. The general procedure to build an enhanced strain virtual element scheme is presented in this section while the choice of special element, a *brick deltahedron* and its peculiar features will be described in the following one.

### 5.2.1. Enhanced strain field

The key idea of enhanced strain methods is to augment the strain field expressed in terms of *generalized* parameters of (1.22), here indicated with  $\hat{\boldsymbol{\varepsilon}}^g$ , with additional non-compatible ones, namely the *enhanced strain parameters*  $\hat{\boldsymbol{\varepsilon}}^{en}$ . The approximation (1.22) then becomes

$$\boldsymbol{\varepsilon}(\boldsymbol{\xi}) \approx \boldsymbol{\varepsilon}^h(\boldsymbol{\xi}) = \mathbf{N}_\varepsilon(\boldsymbol{\xi})\hat{\boldsymbol{\varepsilon}} = \begin{bmatrix} \mathbf{N}_\varepsilon^g(\boldsymbol{\xi}) & \mathbf{N}_\varepsilon^{en}(\boldsymbol{\xi}) \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{\varepsilon}}^g \\ \hat{\boldsymbol{\varepsilon}}^{en} \end{Bmatrix} \quad (5.2)$$

where the  $[6 \times n_\varepsilon^g]$  matrix  $\mathbf{N}_\varepsilon^g$  gathers the shape functions mapping the generalized parameters  $\hat{\boldsymbol{\varepsilon}}^g$  and the  $[6 \times n_\varepsilon^{en}]$  matrix  $\mathbf{N}_\varepsilon^{en}$  maps the enhanced strain parameters  $\hat{\boldsymbol{\varepsilon}}^{en}$ . The matrices  $\mathbf{N}_\varepsilon^g$  and  $\mathbf{N}_\varepsilon^{en}$  must satisfy two conditions:

- they must *span* different strain spaces without intersections

$$\mathbf{N}_\varepsilon^g \hat{\boldsymbol{\varepsilon}}^g \neq \mathbf{N}_\varepsilon^{en} \hat{\boldsymbol{\varepsilon}}^{en} \quad \forall \hat{\boldsymbol{\varepsilon}}^g \neq \mathbf{0}, \hat{\boldsymbol{\varepsilon}}^{en} \neq \mathbf{0} \quad (5.3)$$

which implies

$$\begin{aligned} (\hat{\boldsymbol{\varepsilon}}^g \mathbf{N}_\varepsilon^g)^\top \mathbf{N}_\varepsilon^g \hat{\boldsymbol{\varepsilon}}^g &\neq (\hat{\boldsymbol{\varepsilon}}^g \mathbf{N}_\varepsilon^g)^\top \mathbf{N}_\varepsilon^{en} \hat{\boldsymbol{\varepsilon}}^{en} && \forall \hat{\boldsymbol{\varepsilon}}^g \neq \mathbf{0}, \hat{\boldsymbol{\varepsilon}}^{en} \neq \mathbf{0} \\ 1 &\neq (\hat{\boldsymbol{\varepsilon}}^g \mathbf{N}_\varepsilon^g)^\top \mathbf{N}_\varepsilon^{en} \hat{\boldsymbol{\varepsilon}}^{en} && \forall \hat{\boldsymbol{\varepsilon}}^g \neq \mathbf{0}, \hat{\boldsymbol{\varepsilon}}^{en} \neq \mathbf{0} \\ |\Omega_e| &\neq (\hat{\boldsymbol{\varepsilon}}^g)^\top \left[ \int_{\Omega_e} (\mathbf{N}_\varepsilon^g)^\top \mathbf{N}_\varepsilon^{en} d\Omega \right] \hat{\boldsymbol{\varepsilon}}^{en} && \forall \Omega_e, \hat{\boldsymbol{\varepsilon}}^g \neq \mathbf{0}, \hat{\boldsymbol{\varepsilon}}^{en} \neq \mathbf{0} \\ \implies &\int_{\Omega_e} (\mathbf{N}_\varepsilon^g)^\top \mathbf{N}_\varepsilon^{en} d\Omega = \mathbf{0} && (5.4) \end{aligned}$$

- they must not be orthogonal with respect to the elastic matrix  $\mathbf{D}$

$$\int_{\Omega_e} (\mathbf{N}_\varepsilon^g)^\top \mathbf{D} \mathbf{N}_\varepsilon^{en} d\Omega \neq \mathbf{0} \quad (5.5)$$

### 5.2.2. Stress and displacement field

The stress  $\boldsymbol{\sigma}$  and displacement  $\mathbf{u}$  field are modelled as in (1.23) and (1.21) respectively, here recalled.

$$\begin{aligned}\boldsymbol{\sigma}(\boldsymbol{\xi}) &\approx \boldsymbol{\sigma}^h(\boldsymbol{\xi}) = \mathbf{N}_\sigma(\boldsymbol{\xi})\hat{\boldsymbol{\sigma}} \\ \mathbf{u}(\boldsymbol{\xi}) &\approx \mathbf{u}^h(\boldsymbol{\xi}) = \mathbf{N}_u(\boldsymbol{\xi})\hat{\mathbf{u}}\end{aligned}$$

The number  $n_\sigma$  of stress parameters collected in vector  $\hat{\boldsymbol{\sigma}}$  is equal to the number  $n_\varepsilon^g$  of generalized strain parameters collected in  $\hat{\boldsymbol{\varepsilon}}^g$  so that scalar product can be computed to enforce the generalized variables condition in the sense specified by (1.33), leading to

$$\hat{\boldsymbol{\sigma}}^T \hat{\boldsymbol{\varepsilon}}^g = \int_{\Omega_e} \boldsymbol{\sigma}^T \boldsymbol{\varepsilon}^g d\Omega = \hat{\boldsymbol{\sigma}}^T \left( \int_{\Omega_e} \mathbf{N}_\sigma^T \mathbf{N}_\varepsilon^g d\Omega \right) \hat{\boldsymbol{\varepsilon}}^g \implies \int_{\Omega_e} \mathbf{N}_\sigma^T \mathbf{N}_\varepsilon^g d\Omega = \mathbf{I} \quad (5.6)$$

Following the choice made in (1.36), to satisfy (5.6) the local stress model can be taken as

$$\mathbf{N}_\sigma = \mathbf{N}_\varepsilon^g \left( \int_{\Omega_e} (\mathbf{N}_\varepsilon^g)^T \mathbf{N}_\varepsilon^g d\Omega \right)^{-1} = \mathbf{N}_\varepsilon^g (\mathbf{G}^g)^{-1} \quad (5.7)$$

where the square  $[n_\varepsilon^g \times n_\varepsilon^g]$  invertible matrix  $\mathbf{G}^g$  is given by

$$\mathbf{G}^g = \int_{\Omega_e} (\mathbf{N}_\varepsilon^g)^T \mathbf{N}_\varepsilon^g d\Omega \quad (5.8)$$

Moreover, remembering (5.4) and being  $\hat{\boldsymbol{\sigma}}$  and  $\hat{\boldsymbol{\varepsilon}}^g$  generalized variables, the local approximate stresses are orthogonal to the local enhanced strains

$$\begin{aligned}\int_{\Omega_e} \boldsymbol{\sigma}^T \boldsymbol{\varepsilon}^{en} d\Omega &= \hat{\boldsymbol{\sigma}}^T \left( \int_{\Omega_e} \mathbf{N}_\sigma^T \mathbf{N}_\varepsilon^{en} d\Omega \right) \hat{\boldsymbol{\varepsilon}}^{en} = \\ &= \hat{\boldsymbol{\sigma}}^T (\mathbf{G}^g)^{-T} \left( \int_{\Omega_e} (\mathbf{N}_\varepsilon^{en})^T \mathbf{N}_\varepsilon^{en} d\Omega \right) \hat{\boldsymbol{\varepsilon}}^{en} = 0 \quad \forall \hat{\boldsymbol{\sigma}} \neq \mathbf{0}, \hat{\boldsymbol{\varepsilon}}^{en} \neq \mathbf{0} \quad (5.9)\end{aligned}$$

which implies

$$\int_{\Omega_e} \mathbf{N}_\sigma^T \mathbf{N}_\varepsilon^{en} d\Omega = 0 \quad (5.10)$$

### 5.2.3. Discretized mixed functional

Recalling the local Hu-Washizu functional  $\Pi_e$  defined in (1.18)

$$\begin{aligned}\Pi_e(\mathbf{u}, \boldsymbol{\varepsilon}, \boldsymbol{\sigma}) &= \frac{1}{2} \int_{\Omega_e} \boldsymbol{\varepsilon}^T(\boldsymbol{\xi}) \mathbf{D} \boldsymbol{\varepsilon}(\boldsymbol{\xi}) d\Omega - \int_{\Omega_e} \boldsymbol{\sigma}^T(\boldsymbol{\xi}) \left( \boldsymbol{\varepsilon}(\boldsymbol{\xi}) - \mathbf{S} \mathbf{u}(\boldsymbol{\xi}) \right) d\Omega + \\ &\quad - \int_{\Omega_e} \mathbf{u}^T(\boldsymbol{\xi}) \mathbf{b}(\boldsymbol{\xi}) d\Omega - \int_{\partial_p \Omega_e} \mathbf{u}^T(\boldsymbol{\xi}) \mathbf{p}(\boldsymbol{\xi}) d\Sigma\end{aligned}$$

and substituting the assumed fields (1.21), (5.2) and (1.23), one has the corresponding discrete functional  $\Pi_e^h$

$$\begin{aligned} \Pi_e^h(\hat{\mathbf{u}}, \hat{\boldsymbol{\varepsilon}}, \hat{\boldsymbol{\sigma}}) &= \frac{1}{2} \hat{\boldsymbol{\varepsilon}}^T \left( \int_{\Omega_e} \begin{bmatrix} (\mathbf{N}_\varepsilon^g)^T \\ (\mathbf{N}_\varepsilon^{en})^T \end{bmatrix} \mathbf{D} \begin{bmatrix} \mathbf{N}_\varepsilon^g & \mathbf{N}_\varepsilon^{en} \end{bmatrix} d\Omega \right) \hat{\boldsymbol{\varepsilon}} + \\ &\quad - \hat{\boldsymbol{\sigma}}^T \left( \int_{\Omega_e} \mathbf{N}_\sigma^T \left( \begin{bmatrix} \mathbf{N}_\varepsilon^g & \mathbf{N}_\varepsilon^{en} \end{bmatrix} \hat{\boldsymbol{\varepsilon}} - \mathbf{S} \mathbf{N}_u \hat{\mathbf{u}} \right) d\Omega \right) + \\ &\quad - \hat{\mathbf{u}}^T \left( \int_{\Omega_e} \mathbf{N}_u^T \mathbf{b} d\Omega + \int_{\partial_p \Omega_e} \mathbf{N}_u^T \mathbf{p} d\Sigma \right) \end{aligned} \quad (5.11)$$

which, in view of (5.6) and (5.10), reduces to

$$\Pi_e^h(\hat{\mathbf{u}}, \hat{\boldsymbol{\varepsilon}}, \hat{\boldsymbol{\sigma}}) = \frac{1}{2} \hat{\boldsymbol{\varepsilon}}^T \mathbf{E} \hat{\boldsymbol{\varepsilon}} - \hat{\boldsymbol{\sigma}}^T (\hat{\boldsymbol{\varepsilon}}^g - \mathbf{C} \hat{\mathbf{u}}) - \hat{\mathbf{u}}^T \mathbf{F}_e \quad (5.12)$$

where the following expressions have been introduced

- $[n_\varepsilon \times n_\varepsilon]$  elastic matrix  $\mathbf{E}$

$$\begin{aligned} \mathbf{E} &= \int_{\Omega_e} \begin{bmatrix} (\mathbf{N}_\varepsilon^g)^T \\ (\mathbf{N}_\varepsilon^{en})^T \end{bmatrix} \mathbf{D} \begin{bmatrix} \mathbf{N}_\varepsilon^g & \mathbf{N}_\varepsilon^{en} \end{bmatrix} d\Omega = \\ &= \int_{\Omega_e} \begin{bmatrix} (\mathbf{N}_\varepsilon^g)^T \mathbf{D} \mathbf{N}_\varepsilon^g & (\mathbf{N}_\varepsilon^g)^T \mathbf{D} \mathbf{N}_\varepsilon^{en} \\ (\mathbf{N}_\varepsilon^{en})^T \mathbf{D} \mathbf{N}_\varepsilon^g & (\mathbf{N}_\varepsilon^{en})^T \mathbf{D} \mathbf{N}_\varepsilon^{en} \end{bmatrix} d\Omega = \\ &= \begin{bmatrix} \mathbf{E}^{g,g} & \mathbf{E}^{g,en} \\ \mathbf{E}^{en,g} & \mathbf{E}^{en,en} \end{bmatrix} \end{aligned} \quad (5.13)$$

- $[n_\varepsilon^g \times n_u]$  compatibility matrix  $\mathbf{C}$

$$\mathbf{C} = \int_{\Omega_e} \mathbf{N}_\sigma^T \mathbf{S} \mathbf{N}_u d\Omega = (\mathbf{G}^g)^{-T} \int_{\Omega_e} (\mathbf{N}_\varepsilon^g)^T \mathbf{S} \mathbf{N}_u d\Omega = \mathbf{G}^{-1} \mathbf{A} \quad (5.14)$$

where the symmetry of matrix  $\mathbf{G}$  has been exploited and the  $[n_\varepsilon^g \times n_u]$  matrix  $\mathbf{A}$  is

$$\mathbf{A} = \int_{\Omega_e} (\mathbf{N}_\varepsilon^g)^T \mathbf{S} \mathbf{N}_u d\Omega \quad (5.15)$$

- $[n_u \times 1]$  local equivalent nodal forces vector, defined as in (1.42)

$$\mathbf{F}_e = \int_{\Omega_e} \mathbf{N}_u^T \mathbf{b} d\Omega + \int_{\partial_p \Omega_e} \mathbf{N}_u^T \mathbf{p} d\Sigma$$

### 5.2.4. Stationarity of the mixed functional

Following what has been done in Section 1.3.2, by enforcing the stationarity of the mixed discrete functional (5.12) with respect to the variables  $\hat{\mathbf{u}}$ ,  $\hat{\boldsymbol{\varepsilon}}^g$ ,  $\hat{\boldsymbol{\varepsilon}}^{en}$  and  $\hat{\boldsymbol{\sigma}}$ , one obtains the algebraic governing equations

- equilibrium

$$\partial_{\hat{\mathbf{u}}}\Pi_e^h = \mathbf{0} \implies \mathbf{C}^T \hat{\boldsymbol{\sigma}} = \mathbf{F}_e \quad (5.16)$$

- constitutive law

$$\partial_{\hat{\boldsymbol{\varepsilon}}^g}\Pi_e^h = \mathbf{0} \implies \hat{\boldsymbol{\sigma}} = \mathbf{E}^{g,g} \hat{\boldsymbol{\varepsilon}}^g + \mathbf{E}^{g,en} \hat{\boldsymbol{\varepsilon}}^{en} \quad (5.17)$$

- condition on enhanced strains

$$\partial_{\hat{\boldsymbol{\varepsilon}}^{en}}\Pi_e^h = \mathbf{0} \implies \mathbf{E}^{en,g} \hat{\boldsymbol{\varepsilon}}^g + \mathbf{E}^{en,en} \hat{\boldsymbol{\varepsilon}}^{en} = \mathbf{0} \quad (5.18)$$

- kinematic compatibility

$$\partial_{\hat{\boldsymbol{\sigma}}}\Pi_e^h = \mathbf{0} \implies \hat{\boldsymbol{\varepsilon}}^g = \mathbf{C} \hat{\mathbf{u}} \quad (5.19)$$

Equation (5.18) justifies the requirement (5.5) between  $\mathbf{N}_\varepsilon^g$  and  $\mathbf{N}_\varepsilon^{en}$ . Indeed, if  $\mathbf{N}_\varepsilon^g$  and  $\mathbf{N}_\varepsilon^{en}$  were orthogonal with respect to the material stiffness matrix, then  $\mathbf{E}_g^{en} = \mathbf{E}_{en}^g = \mathbf{0}$  and equation (5.18) would imply  $\hat{\boldsymbol{\varepsilon}}^{en} = \mathbf{0}$  since  $\mathbf{E}^{en,en}$  is non-singular by construction. Isolating from (5.18)  $\hat{\boldsymbol{\varepsilon}}^{en}$

$$\hat{\boldsymbol{\varepsilon}}^{en} = -(\mathbf{E}^{en,en})^{-1} \mathbf{E}^{en,g} \hat{\boldsymbol{\varepsilon}}^g \quad (5.20)$$

and substituting in (5.17) yields to

$$\hat{\boldsymbol{\sigma}} = \mathbf{E}^{g,g} \hat{\boldsymbol{\varepsilon}}^g - \mathbf{E}^{g,en} (\mathbf{E}^{en,en})^{-1} \mathbf{E}^{en,g} \hat{\boldsymbol{\varepsilon}}^g = \tilde{\mathbf{E}} \hat{\boldsymbol{\varepsilon}}^g \quad (5.21)$$

where the modified elastic stiffness matrix  $\tilde{\mathbf{E}}$  has been introduced

$$\tilde{\mathbf{E}} = \mathbf{E}^{g,g} - \mathbf{E}^{g,en} (\mathbf{E}^{en,en})^{-1} \mathbf{E}^{en,g}$$

Finally, substituting (5.17) in (5.21) and (5.21) in (5.16) the final local algebraic system is obtained

$$\tilde{\mathbf{K}}_e \hat{\mathbf{u}} = \mathbf{F}_e \quad (5.22)$$

where the  $[n_u \times n_u]$  symmetric positive semi-definite matrix  $\tilde{\mathbf{K}}_e$  is the *enhanced strain local stiffness matrix*, given by

$$\tilde{\mathbf{K}}_e = \mathbf{C}^T \tilde{\mathbf{E}} \mathbf{C} \quad (5.23)$$

Following the remark on the rank of the local stiffness matrix pointed out in Section 1.3.2, if  $n_u - n_\varepsilon \leq 6$ , then matrix  $\tilde{\mathbf{K}}_e$  has the correct degree of singularity and no stabilization is required. A proper choice of the geometry and of the strain model is hence envisaged to achieve a self-stabilizing element.

### 5.2.5. Assembly, boundary conditions and solution

The assembly process follows the same steps presented in Section 1.3.4, so that the global stiffness matrix becomes

$$\tilde{\mathbf{K}} = \bigwedge_{e=1}^{n_e} \tilde{\mathbf{K}}_e$$

and the equivalent nodal forces vector

$$\mathbf{F} = \bigwedge_{e=1}^{n_e} \mathbf{F}_e$$

Once the global system

$$\tilde{\mathbf{K}} \mathbf{U} = \mathbf{F}$$

is obtained, boundary conditions can be enforced according to the standard way presented in Section 1.3.5 and the solution retrieval follows the same concepts explained in the above-mentioned section.

### 5.2.6. Strains and stresses recovery

Once the unknown discrete field  $\mathbf{U}$  is found, it is possible to recover the local degrees of freedom  $\hat{\mathbf{u}}$ . From these, the local generalized strain parameters  $\hat{\varepsilon}^g$  are obtained through the compatibility equation in (5.19)

$$\hat{\varepsilon}^g = \mathbf{C} \hat{\mathbf{u}}$$

and, consequently, the local enhanced strain parameters  $\hat{\varepsilon}^{en}$  through (5.20)

$$\hat{\varepsilon}^{en} = -(\mathbf{E}^{en,en})^{-1} \mathbf{E}^{en,g} \hat{\varepsilon}^g$$

so that the local strain field  $\boldsymbol{\varepsilon}(\boldsymbol{\xi})$  can be extrapolated from (5.2) as

$$\boldsymbol{\varepsilon}(\boldsymbol{\xi}) = \mathbf{N}_\varepsilon^g(\boldsymbol{\xi})\hat{\boldsymbol{\varepsilon}}^g + \mathbf{N}_\varepsilon^{en}(\boldsymbol{\xi})\hat{\boldsymbol{\varepsilon}}^{en}$$

The local stress parameters  $\hat{\boldsymbol{\sigma}}$  are computed from the constitutive law (5.17)

$$\hat{\boldsymbol{\sigma}} = \tilde{\mathbf{E}}\hat{\boldsymbol{\varepsilon}}^g$$

so that the local stress field  $\boldsymbol{\sigma}(\boldsymbol{\xi})$  can be retrieved as

$$\boldsymbol{\sigma}(\boldsymbol{\xi}) = \mathbf{N}_\sigma(\boldsymbol{\xi})\hat{\boldsymbol{\sigma}}$$

### 5.3. Enhanced strain fields choices for the VEM

In the following, three particular choices for the enhanced strain VEM formulation discussed in Section 5.2 are proposed. Before presenting the peculiar choices of the fields, some important remarks are due concerning the general procedure described in the previous section.

- The geometry of the element is chosen to be a *brick-type deltahedron*, i.e. a polyhedron with 8 nodes, 26 edges and 12 faces. A particular case can be easily obtained by triangulating the 6 faces of a hexahedron, obtaining 6 couples of coplanar faces. However, the following formulations work with the general 8-nodes deltahedron.
- The VEM order will be  $k = 1$ , in the sense that only vertex-type DOFs will appear in the formulation, namely 24 DOFs per element. Moreover, the restriction of the virtual displacement field on the skeleton of the element (i.e., its edges) is piecewise linear and globally continuous, as well as on the faces, in view of the deltahedra property (Proposition 5.1).
- The enhanced incompatible part of the strain field  $\mathbf{N}_\varepsilon^{en}\hat{\boldsymbol{\varepsilon}}^{en}$  has no effect on the stability of the system. Therefore, it is convenient to enrich the generalized part of the strain field  $\mathbf{N}_\varepsilon^g\hat{\boldsymbol{\varepsilon}}^g$ , which will coincide with the total strain field itself  $\mathbf{N}_\varepsilon\hat{\boldsymbol{\varepsilon}}$ . In the VEM context, the operation of enriching the strain model translates into projecting the strain field into the space of polynomials of degree higher than one degree less of the VEM order. Namely, the strain model  $\mathbf{N}_\varepsilon$  is chosen to describe a polynomial of order  $p$ , not necessarily complete

$$\mathbf{N}_\varepsilon(\boldsymbol{\xi}) \in \mathcal{P}_p, \quad p > 0$$

- Performing by parts integration on matrix  $\mathbf{A}$  defined in (5.15) yields to

$$\begin{aligned} \mathbf{A} &= \int_P (\mathbf{N}_\varepsilon^g)^\top \mathbf{S} \mathbf{N}_u d\Omega = \int_{\partial P} (\mathbb{N}_P \mathbf{N}_\varepsilon^g)^\top \mathbf{N}_u d\Sigma - \int_P (\mathbf{S}^\top \mathbf{N}_\varepsilon^g)^\top \mathbf{N}_u d\Omega = \\ &= \sum_{F \in \partial P} \left[ \int_F (\mathbb{N}_F \mathbf{N}_\varepsilon^g)^\top \mathbf{N}_u d\Sigma \right] - \int_P (\mathbf{S}^\top \mathbf{N}_\varepsilon^g)^\top \mathbf{N}_u d\Omega \quad (5.24) \end{aligned}$$

The first summation of integrals over the faces  $F$ ,  $\mathbf{A}_1$ , can be computed exploiting the deltahedra property of Proposition 5.1, since the  $j^{\text{th}}$  shape function  $N_u^j|_F$  is an explicitly known linear function  $\forall j = 1, \dots, 8$  and the strain model  $\mathbf{N}_\varepsilon^g$  is a-priori chosen. The second integral over the volume of the element can be computed if internal moment polyhedron-type DOFs are available or if the strain field is divergence-free. In this latter option  $\mathbf{A}_2 = \mathbf{0}$ , as it happens in standard VEM of order  $k = 1$ , where the strain model is constant (a particular case where its divergence is null). In the enhanced formulation, the strain field is no longer a constant function and to be able to compute all the components of matrix  $\mathbf{A}$  without the need of introducing internal moments one has to chose a divergence-free field  $\mathbf{N}_\varepsilon^g$ .

### 5.3.1. Linear strain field

Keeping in mind that the vertex-type DOFs are 24 for the element adopted, the number of strain parameters  $n_\varepsilon$  has to be not less than 18, so that constraining the 6 rigid body motions will reestablish the full rank of the stiffness matrix  $\mathbf{K}_e$ . The first immediate choice for enhancing the strain field to achieve self-stability is therefore to increase the order of the complete polynomial strain to  $p = 1$ , i.e. one degree more than standard  $k = 1$  VEM. This leads to a strain model  $\mathbf{N}_\varepsilon$  made by  $6\nu_1 = 24$  parameters.

$$\mathbf{N}_\varepsilon = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \xi & 0 & 0 & 0 & 0 & 0 & \eta & 0 & 0 & 0 & 0 & 0 & \zeta & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \xi & 0 & 0 & 0 & 0 & 0 & \eta & 0 & 0 & 0 & 0 & 0 & \zeta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \xi & 0 & 0 & 0 & 0 & 0 & \eta & 0 & 0 & 0 & 0 & 0 & \zeta & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \xi & 0 & 0 & 0 & 0 & 0 & \eta & 0 & 0 & 0 & 0 & 0 & \zeta & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \xi & 0 & 0 & 0 & 0 & 0 & \eta & 0 & 0 & 0 & 0 & 0 & \zeta & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \xi & 0 & 0 & 0 & 0 & 0 & \eta & 0 & 0 & 0 & 0 & 0 & \zeta \end{bmatrix} \quad (5.25)$$

It was noted that the above model generates a  $[24 \times 24]$  matrix  $\mathbf{A}_1$  of rank 18, which would lead to  $\text{rank}(\mathbf{K}_e) = 18$ . However, matrix  $\mathbf{A}_2$  would not be computable as the field in (5.25) is not divergence-free and no moments DOFs are available. Since the strain field components are already described by a complete polynomial, namely  $\mathbf{N}_\varepsilon \in [\mathcal{P}_1]^6$ , there is no way to make the field divergence-free, while keeping the number of strain

parameters  $n_\varepsilon = 18$  (see (5.26)). Indeed adding linear parameters would mean adding a linear combination of the parameters already included in the complete field, which would not restore the rank of  $\mathbf{A}_1$ .

$$\mathbf{N}_\varepsilon = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \eta & 0 & 0 & \zeta & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \xi & 0 & 0 & 0 & 0 & 0 & 0 & \zeta & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \xi & 0 & 0 & \eta & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \zeta \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \xi & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \eta & 0 & 0 & 0 \end{bmatrix} \quad (5.26)$$

A second option is to add three 0<sup>th</sup>-order internal moments DOFs, increasing the number of local DOFs to  $n_u = 27$ . It was noted that this choice, while formally allowing to compute the entries of  $\mathbf{A}_2$ , does not increase the rank of  $\mathbf{A}$ , still remaining  $\text{rank}(\mathbf{K}_e) = 18$ , which is now insufficient to self-stabilize the element, having the number of DOFs increased to 27. The only remaining choice is to further increase the polynomial degree of the strain model.

### 5.3.2. 31-strain-parameters 24-DOFs self-stabilized VE

Addressing the issues highlighted in Section 5.3.1, a divergence-free 2<sup>nd</sup>-degree polynomial strain field is tested. Clearly, such field won't be complete anymore as some parameters are suppressed by the divergence-free condition  $\mathbf{S}^T \mathbf{N}_\varepsilon = \mathbf{0}$ . A way to produce this field is to exploit a complete 4<sup>th</sup>-order polynomial *Airy function*  $\phi(\xi, \eta, \zeta) \in \mathcal{P}_4$  and compute the components of the strain field such that it becomes divergence-free.

$$\begin{aligned} \phi(\xi, \eta, \zeta) = & 1\varepsilon_{00} + \xi\varepsilon_{01} + \eta\varepsilon_{02} + \zeta\varepsilon_{03} + \\ & + \xi^2\varepsilon_{11} + \eta^2\varepsilon_{12} + \zeta^2\varepsilon_{13} + \xi\eta\varepsilon_{14} + \eta\zeta\varepsilon_{15} + \xi\zeta\varepsilon_{16} + \\ & + \left( \begin{array}{c} \xi^3\varepsilon_{17} + \eta^3\varepsilon_{18} + \zeta^3\varepsilon_{19} + \\ + \xi^2\eta\varepsilon_{10} + \xi^2\zeta\varepsilon_{11} + \eta^2\xi\varepsilon_{12} + \eta^2\zeta\varepsilon_{13} + \zeta^2\xi\varepsilon_{14} + \zeta^2\eta\varepsilon_{15} + \xi\eta\zeta\varepsilon_{16} \end{array} \right) + \\ & + \left( \begin{array}{c} \xi^4\varepsilon_{17} + \eta^4\varepsilon_{18} + \zeta^4\varepsilon_{19} + \\ + \xi^3\eta\varepsilon_{20} + \xi^3\zeta\varepsilon_{21} + \eta^3\xi\varepsilon_{22} + \eta^3\zeta\varepsilon_{23} + \zeta^3\xi\varepsilon_{24} + \zeta^3\eta\varepsilon_{25} + \\ + \xi^2\eta^2\varepsilon_{26} + \eta^2\zeta^2\varepsilon_{27} + \zeta^2\xi^2\varepsilon_{28} + \xi^2\eta\zeta\varepsilon_{29} + \xi\eta^2\zeta\varepsilon_{30} + \xi\eta\zeta^2\varepsilon_{31} \end{array} \right) \end{aligned} \quad (5.27)$$

$$\boldsymbol{\varepsilon}(\xi, \eta, \zeta) = \begin{Bmatrix} \varepsilon_x(\xi, \eta, \zeta) \\ \varepsilon_y(\xi, \eta, \zeta) \\ \varepsilon_z(\xi, \eta, \zeta) \\ \gamma_{xy}(\xi, \eta, \zeta) \\ \gamma_{yz}(\xi, \eta, \zeta) \\ \gamma_{xz}(\xi, \eta, \zeta) \end{Bmatrix} = \begin{Bmatrix} \phi_{,\eta\eta} + \phi_{,\zeta\zeta} \\ \phi_{,\xi\xi} + \phi_{,\zeta\zeta} \\ \phi_{,\xi\xi} + \phi_{,\eta\eta} \\ -\phi_{,\xi\eta} \\ -\phi_{,\eta\zeta} \\ -\phi_{,\xi\zeta} \end{Bmatrix} \quad (5.28)$$

The constant and linear part of the Airy function  $\phi$  leads to no contribution in the strain model. The quadratic, cubic and 4<sup>th</sup>-power terms generate constant, linear and quadratic strain parameters respectively. Applying definition (5.28) to (5.27) and simplifying the common terms over each  $\hat{\varepsilon}_i$  parameter, one obtains

$$\boldsymbol{\varepsilon}(\xi, \eta, \zeta) = \mathbf{N}_\varepsilon(\xi, \eta, \zeta) \hat{\boldsymbol{\varepsilon}}$$

where the  $[6 \times 31]$  matrix  $\mathbf{N}_\varepsilon$  modelling the strain field is

$$\mathbf{N}_\varepsilon = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & \eta & \zeta & 0 & 0 & \xi & \zeta & \xi & \eta & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \xi & 0 & \zeta & \eta & \zeta & 0 & 0 & \xi & \eta & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \xi & \eta & 0 & \eta & \zeta & \xi & \zeta & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -\xi & 0 & -\eta & 0 & 0 & 0 & -\zeta \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\eta & 0 & -\zeta & -\xi \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -\xi & 0 & 0 & -\zeta & 0 & -\eta \end{bmatrix}$$

$$\begin{bmatrix} 0 & \eta^2 & \zeta^2 & 0 & 0 & 2\xi\eta & 2\eta\zeta & 2\xi\zeta & 2\eta\zeta & \xi^2 & \eta^2 + \zeta^2 & \xi^2 & 0 & 2\xi\zeta & 2\xi\eta \\ \xi^2 & 0 & \zeta^2 & 2\xi\eta & 2\xi\zeta & 0 & 0 & 2\xi\zeta & 2\eta\zeta & \eta^2 & \eta^2 & \xi^2 + \zeta^2 & 2\eta\zeta & 0 & 2\xi\eta \\ \xi^2 & \eta^2 & 0 & 2\xi\eta & 2\xi\zeta & 2\xi\eta & 2\eta\zeta & 0 & 0 & \xi^2 + \eta^2 & \zeta^2 & \zeta^2 & 2\eta\zeta & 2\xi\zeta & 0 \\ 0 & 0 & 0 & -\xi^2 & 0 & -\eta^2 & 0 & 0 & 0 & -2\xi\eta & 0 & 0 & -2\xi\zeta & -2\eta\zeta & -\zeta^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\eta^2 & 0 & -\zeta^2 & 0 & -2\eta\zeta & 0 & -\xi^2 & -2\xi\eta & -2\xi\zeta \\ 0 & 0 & 0 & 0 & -\xi^2 & 0 & 0 & -\zeta^2 & 0 & 0 & 0 & -2\xi\zeta & -2\xi\eta & -\eta^2 & -2\eta\zeta \end{bmatrix} \quad (5.29)$$

The enriched strain field above leads to

- $[31 \times 24]$  matrix  $\mathbf{A} = \mathbf{A}_1$
- $[31 \times 31]$  matrix  $\mathbf{G}$
- $[31 \times 24]$  matrix  $\mathbf{C}$
- $[31 \times 31]$  matrix  $\mathbf{E}$
- $[24 \times 24]$  matrix  $\mathbf{K}_e$

- $[24 \times 1]$  vector  $\mathbf{F}_e$

which are constructed following the same procedure presented for deltahedra in 5.1.2. The generated element will be denoted  $\Delta\text{VEM8SS31-24DOFs}$ , referring to an 8-nodes self-stabilized virtual element with 31 strain parameters and 24 degrees of freedom. Proving that the stiffness matrix  $\mathbf{K}_e$  generated by the above strain model shows the correct rank deficiency, and hence the element require no stabilization, is an extremely difficult task. Indeed, to state that such element can be safely adopted, its stiffness matrix  $\mathbf{K}_e$  must have, up to machine precision, at most 6 null eigenvalues.

A numerical optimization has been performed to find the configuration of the element which leads to the closest-to-zero 7<sup>th</sup> smallest eigenvalue. This search is obviously non-trivial, as the convexity of the corresponding objective function is not guaranteed and enforcing constraints to allow the widest space of configurations while preserving the element's topology (i.e. its faces should not degenerate and its vertices should not penetrate other faces) is challenging. The following optimization problem has been implemented using the MATLAB Optimization Toolbox ([40]).

$$\begin{aligned} \text{Find } \min_{\mathbf{x} \in \mathbb{R}^{24}} f(\mathbf{x}) & \quad (5.30) \\ \text{s.t. } g(\mathbf{x}) \leq 0 & \end{aligned}$$

where  $f(\mathbf{x})$  is the objective function taking as input the variation  $\mathbf{x}$  of coordinates of the 8 vertices with respect to the initial configuration of the cubic element, of dimensions  $[1 \times 1 \times 1]$ , and returning the square of the 7<sup>th</sup> smallest eigenvalue of the generated stiffness matrix  $\mathbf{K}_e$ . The material properties are  $E = 1$ ,  $\nu = 1$ . The enforced non-linear constraints  $g(\mathbf{x})$  have been selected so that two random points cannot coincide, hence preventing triangles degeneration and penetration of faces, namely

$$g(\mathbf{x}) = \mathbf{x}^T \mathbf{x} - 0.5 \quad (5.31)$$

so that in the worst case scenario where only two adjacent vertices move towards each other, they are allowed to move only until they coincide. The configuration (Figure 5.4) leading to the closest-to-zero 7<sup>th</sup> smallest eigenvalue yields to

$$\lambda_7^{\min} = 0.03 \quad (5.32)$$

For comparison, the largest eigenvalue for such configuration was found to be  $\lambda_{\max} = 4.57$ , yielding to a partial validation of the element, as this optimization process would require

further analyses to be considered as a proper verification.

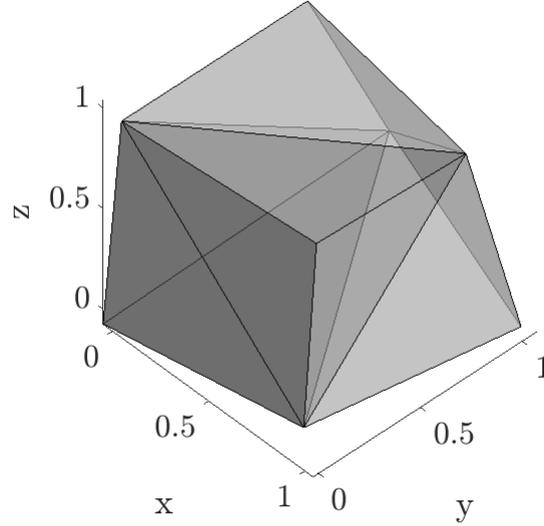


Figure 5.4: Configuration found leading to the closest-to-zero 7<sup>th</sup> smallest eigenvalue of the stiffness matrix  $\mathbf{K}_e$  generated by the 31-parameters strain field, solution of the optimization (5.30). The starting element is the  $1 \times 1 \times 1$  cube.

### 5.3.3. 25-strain-parameters 24-DOFs self-stabilized VE

Even though no stabilization is required for the element defined in 5.3.2, having a polynomial strain field made by 31 parameters leads to extensive computations, both for the entries of matrix  $\mathbf{A}_1$ , where monomials up to degree 3 have to be integrated (2 for the strain field and 1 for the polynomial projection of the displacement field), and for matrix  $\mathbf{G}$  and  $\mathbf{E}$ , where the integrands are monomials up to order 4. Therefore, a reduction of the strain field parameters is desirable. The following choice derives from the most general 2<sup>nd</sup>-degree polynomial strain field which is divergence-free and is obtained from the symmetric gradient of a 3<sup>rd</sup>-degree polynomial displacement field. The  $[6 \times 25]$  matrix  $\mathbf{N}_\epsilon$  modelling the strain field reads

$$\mathbf{N}_\epsilon = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \xi & 0 & \xi & \eta & \zeta & \eta & \zeta \\ 1 & 0 & 0 & 0 & 0 & 1 & \xi & \eta & \zeta & 0 & 0 & \xi & 0 & 0 & \eta & \zeta \\ 1 & 0 & 0 & 1 & 0 & 0 & \xi & \eta & \zeta & \xi & 0 & 0 & \eta & \zeta & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & -\xi & 0 & -\eta & -\zeta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -\xi & 0 & 0 & -\eta & -\zeta & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -\xi & 0 & -\eta & -\zeta & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.33)$$

$$\left[ \begin{array}{cccccccccc} -2\xi^2 + \eta^2 + \zeta^2 & -2\xi\eta & -2\xi\zeta & 0 & 0 & \eta^2 - \zeta^2 & 2\eta\zeta & 2\eta\zeta & \zeta^2 - \eta^2 & \\ \xi^2 - \zeta^2 & 0 & 2\xi\zeta & -2\xi\eta & 2\xi\zeta & \xi^2 - 2\eta^2 + \zeta^2 & -2\eta\zeta & 0 & \xi^2 + \zeta^2 & \\ \xi^2 - \eta^2 & 2\xi\eta & 0 & 2\xi\eta & -2\xi\zeta & \eta^2 - \xi^2 & 0 & -2\eta\zeta & \xi^2 + \eta^2 - 2\zeta^2 & \\ 2\xi\eta & \zeta^2 - \xi^2 & 2\eta\zeta & \zeta^2 - \eta^2 & 2\eta\zeta & 2\xi\eta & 2\xi\zeta & 2\xi\zeta & -2\xi\eta & \\ -2\eta\zeta & 2\xi\zeta & 2\xi\eta & 2\xi\zeta & 2\xi\eta & 2\eta\zeta & \xi^2 - \eta^2 & \xi^2 - \zeta^2 & 2\eta\zeta & \\ 2\xi\zeta & 2\eta\zeta & \eta^2 - \xi^2 & 2\eta\zeta & \eta^2 - \zeta^2 & -2\xi\zeta & 2\xi\eta & 2\xi\eta & 2\xi\zeta & \end{array} \right]$$

The elemental matrices and vectors generated by the above formulation become

- [25 × 24] matrix  $\mathbf{A} = \mathbf{A}_1$
- [25 × 25] matrix  $\mathbf{G}$
- [25 × 24] matrix  $\mathbf{C}$
- [25 × 25] matrix  $\mathbf{E}$
- [24 × 24] matrix  $\mathbf{K}_e$
- [24 × 1] vector  $\mathbf{F}_e$

The generated element will be denoted  $\Delta\text{VEM8SS25-24DOFs}$ , referring to an 8-nodes self-stabilized virtual element with 25 strain parameters and 24 degrees of freedom. The numerical optimization presented in 5.3.2 has been conducted also for this element, in order to attempt an evaluation of its robustness with respect to different nodal configurations. In this case the closest-to-zero 7<sup>th</sup> smallest eigenvalue was found to be

$$\lambda_7^{min} = 0.0093 \quad (5.34)$$

and the largest eigenvalue for the corresponding configuration was  $\lambda_{max} = 2.62$ .

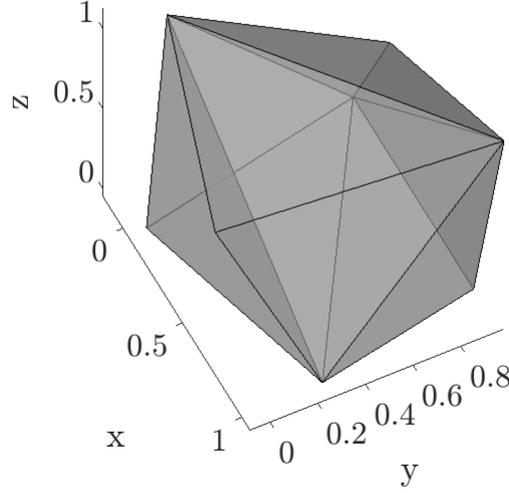


Figure 5.5: Configuration found leading to the closest-to-zero 7<sup>th</sup> smallest eigenvalue of the stiffness matrix  $\mathbf{K}_e$  generated by the 25-parameters strain field, solution of the optimization (5.30). The starting element is the  $1 \times 1 \times 1$  cube.

#### 5.3.4. 18-strain-parameters 24-DOFs self-stabilized VE

A third strain field made by 18 parameters - the minimum to suppress the rank deficiency caused by the shortage of strain parameters with respect to displacement parameters - is investigated in this section. The strain field is obviously still divergence-free to avoid the introduction of internal moments. It is derived by adding to the divergence-free 15-parameters linear field of (5.26) three other parameters obtained as a combination of the 31-parameters field of (5.28). More precisely, the 16<sup>th</sup> parameter is obtained by summing the 4<sup>th</sup> and 6<sup>th</sup> column of the 2<sup>nd</sup>-degree part of (5.28), the 17<sup>th</sup> parameter is obtained by summing the 7<sup>th</sup> and 9<sup>th</sup> column and the 18<sup>th</sup> parameter is obtained by summing the 5<sup>th</sup> and 8<sup>th</sup> column. The  $[6 \times 18]$  matrix  $\mathbf{N}_\varepsilon$  hence reads

$$\mathbf{N}_\varepsilon = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \eta & 0 & 0 & \zeta & 0 & 0 & 2\xi\eta & 4\eta\zeta & 2\xi\zeta \\ 0 & 1 & 0 & 0 & 0 & 0 & \xi & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \zeta & 0 & 2\xi\eta & 2\eta\zeta & 4\xi\zeta \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \xi & 0 & 0 & \eta & 0 & 0 & 0 & 0 & 0 & 4\xi\eta & 2\eta\zeta & 2\xi\zeta \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \zeta & -\xi^2 - \eta^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \xi & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\eta^2 - \zeta^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \eta & 0 & 0 & 0 & 0 & 0 & 0 & -\xi^2 - \zeta^2 \end{bmatrix} \quad (5.35)$$

The elemental matrices and vectors in this case reduces to

- $[18 \times 24]$  matrix  $\mathbf{A} = \mathbf{A}_1$
- $[18 \times 18]$  matrix  $\mathbf{G}$
- $[18 \times 24]$  matrix  $\mathbf{C}$
- $[18 \times 18]$  matrix  $\mathbf{E}$
- $[24 \times 24]$  matrix  $\mathbf{K}_e$
- $[24 \times 1]$  vector  $\mathbf{F}_e$

The element will be denoted  $\Delta\text{VEM8SS18-24DOFs}$ , referring to an 8-nodes self-stabilized virtual element with 18 strain parameters and 24 degrees of freedom. The solution of the optimization problem (5.30) in this case yields to a closest-to-zero 7<sup>th</sup> smallest eigenvalue

$$\lambda_7^{min} = 0.000326 \quad (5.36)$$

and the largest eigenvalue corresponding to the found configuration is  $\lambda_{max} = 3.76$ .

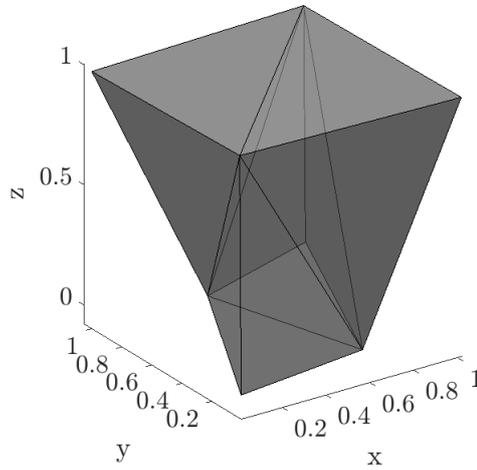


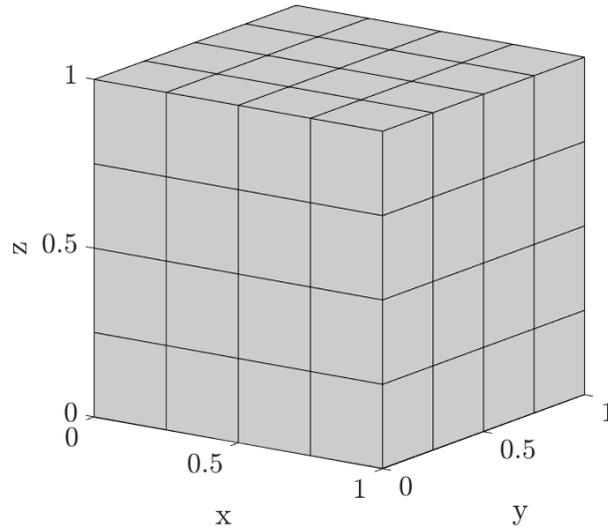
Figure 5.6: Configuration found leading to the closest-to-zero 7<sup>th</sup> smallest eigenvalue of the stiffness matrix  $\mathbf{K}_e$  generated by the 18-parameters strain field, solution of the optimization (5.30). The starting element is the  $1 \times 1 \times 1$  cube.

# 6 | Numerical tests of the $\Delta$ VEM and SSVEM for 3D elastostatics

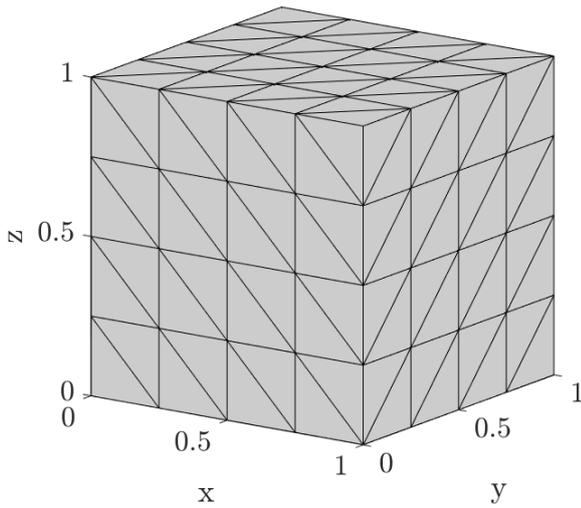
This chapter presents numerical results obtained by a MATLAB ([40]) implementation of the improvements of the standard virtual element method for three-dimensional linear elastostatics described in Chapter 5. Similarly to what has been done in Chapter 4, the units of measurement are not specified and can be arbitrarily taken provided they are consistent (e.g.,  $N/mm^3$  for body forces;  $N/mm^2$  for stresses, surface tractions, Young's modulus and Lamé parameters;  $mm$  for lengths).

## 6.1. h-convergence with brick-type meshes

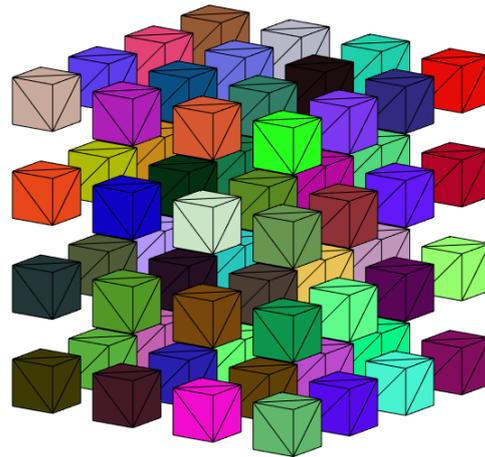
In this section, problem (4.1) is numerically solved for different data, to validate the effectiveness of the  $\Delta$ VEM and its three self-stabilized versions presented in Chapter 5, and compare the results with the standard  $k = 1$  and  $k = 2$  VEM. The meshes adopted here are made by regular cubic hexahedra (*brick* elements) and corresponding deltahedra (Figure 6.1).



(a) Hexahedral brick-type mesh



(b) Deltahedral brick-type mesh



(c) Single brick-type deltahedra

Figure 6.1: Deltahedral mesh (b) obtained by triangulating the faces of a hexahedral brick-type mesh (a) of the  $[1 \times 1 \times 1]$  domain.

### 6.1.1. Trigonometric displacement field

The first  $h$ -convergence analysis is performed with the same data already presented in Section 4.1, here recalled.

$$\begin{cases} -\mathbf{S}^T [\mathbf{D}(\mathbf{S}\mathbf{u})] = \mathbf{b} & \text{in } \Omega = [0, 1]^3 \\ \mathbf{u} = \mathbf{0} & \text{on } \partial\Omega \end{cases} \quad (6.1)$$

where

- Lamé parameters:  $\lambda = 1$  and  $\mu = 1$  (Young's modulus  $E = 2.5$ , Poisson ratio  $\nu = 0.25$ )
- trigonometric body forces  $\mathbf{b}(\mathbf{x})$  in  $\Omega$

$$\mathbf{b}(x, y, z) = C \begin{Bmatrix} -\pi^2 [(\lambda + \mu) \cos(\pi x) \sin(\pi y + \pi z) - (\lambda + 4\mu) \sin(\pi x) \sin(\pi y) \sin(\pi z)] \\ -\pi^2 [(\lambda + \mu) \cos(\pi y) \sin(\pi x + \pi z) - (\lambda + 4\mu) \sin(\pi x) \sin(\pi y) \sin(\pi z)] \\ -\pi^2 [(\lambda + \mu) \cos(\pi z) \sin(\pi x + \pi y) - (\lambda + 4\mu) \sin(\pi x) \sin(\pi y) \sin(\pi z)] \end{Bmatrix}$$

where  $C$  is a constant which is taken to be 0.1.

The exact solution corresponding to problem (6.1) is

$$\mathbf{u}(x, y, z) = C \sin(\pi x) \sin(\pi y) \sin(\pi z) \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix}$$

Figure 6.2 illustrates the convergence rates exhibited under  $h$ -refinement by the implemented  $\Delta$ VEM and self-stabilized VEM program, comparing them with those of the standard virtual elements of order  $k = 1$  and  $k = 2$ .

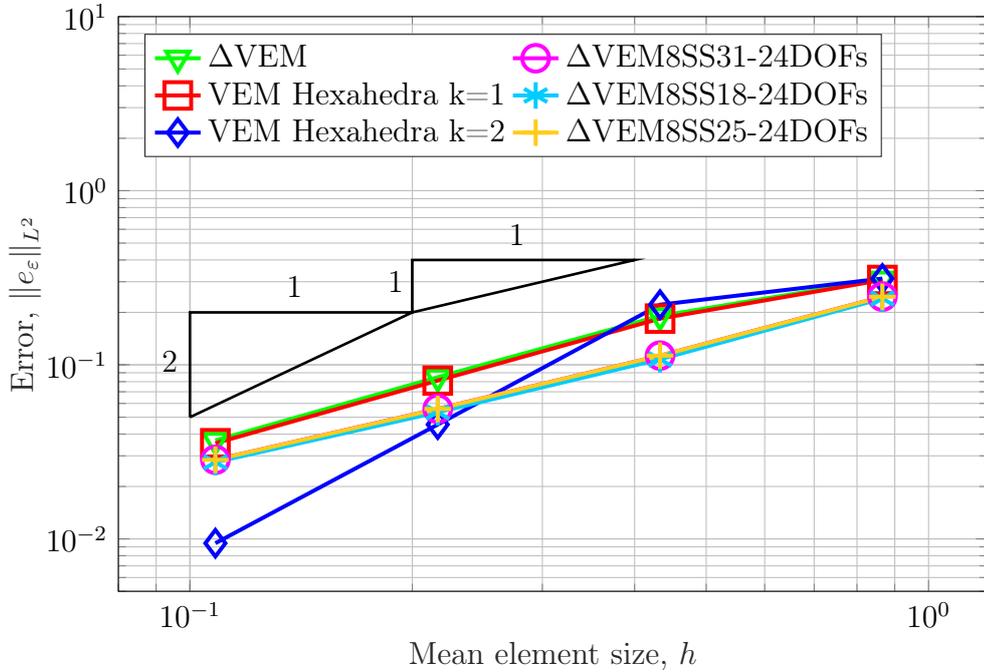


Figure 6.2:  $h$ -refinement convergence test for  $\Delta$ VEM and self-stabilized VEM numerically solving the elastostatic problem whose exact solution is trigonometric.

It is noted that all the convergence lines correctly align with the expected slope and a slight increase of accuracy is exhibited for the self-stabilized VEMs.

### 6.1.2. Cubic polynomial displacement field

The VE programs are tested here with in the unit cube  $\Omega = [0, 1]^3$  where non-homogeneous Dirichlet boundary conditions are enforced on the boundary of the domain and the exact solution is a cubic polynomial. The problem then reads

$$\begin{cases} -\mathbf{S}^T [\mathbf{D}(\mathbf{S}\mathbf{u})] = \mathbf{b} & \text{in } \Omega \\ \mathbf{u} = \bar{\mathbf{u}} & \text{on } \partial\Omega \end{cases} \quad (6.2)$$

where

- Lamé parameters:  $\lambda = 1$  and  $\mu = 1$  (Young's modulus  $E = 2.5$ , Poisson ratio  $\nu = 0.25$ )
- null body forces  $\mathbf{b}$  in  $\Omega$

$$\mathbf{b}(x, y, z) = \mathbf{0}$$

- non-homogeneous Dirichlet boundary conditions  $\bar{\mathbf{u}}(\mathbf{x})$  given by the exact cubic polynomial solution

$$\mathbf{u}(x, y, z) = C \begin{cases} 2x^3 - 3xy^2 - 3xz^2 \\ 2y^3 - 3yx^2 - 3yz^2 \\ 2z^3 - 3zy^2 - 3zx^2 \end{cases} \quad (6.3)$$

where  $C$  is a constant set to 0.1.

The corresponding exact strains to 6.3 are

$$\boldsymbol{\varepsilon}(x, y, z) = C \begin{cases} 6x^2 - 3y^2 - 3z^2 \\ -3x^2 + 6y^2 - 3z^2 \\ -3x^2 - 3y^2 + 6z^2 \\ -12xy \\ -12yz \\ -12xz \end{cases}$$

Figure 6.3 shows a perfect match between the convergence lines of every adopted element and the expected slope. Additionally, the self-stabilized versions of 31 and 25 strain parameters exhibit a slightly better performance in terms of accuracy compared to the  $k = 1$  standard VEM, the  $\Delta$ VEM and the 18-parameters self-stabilized VEM.

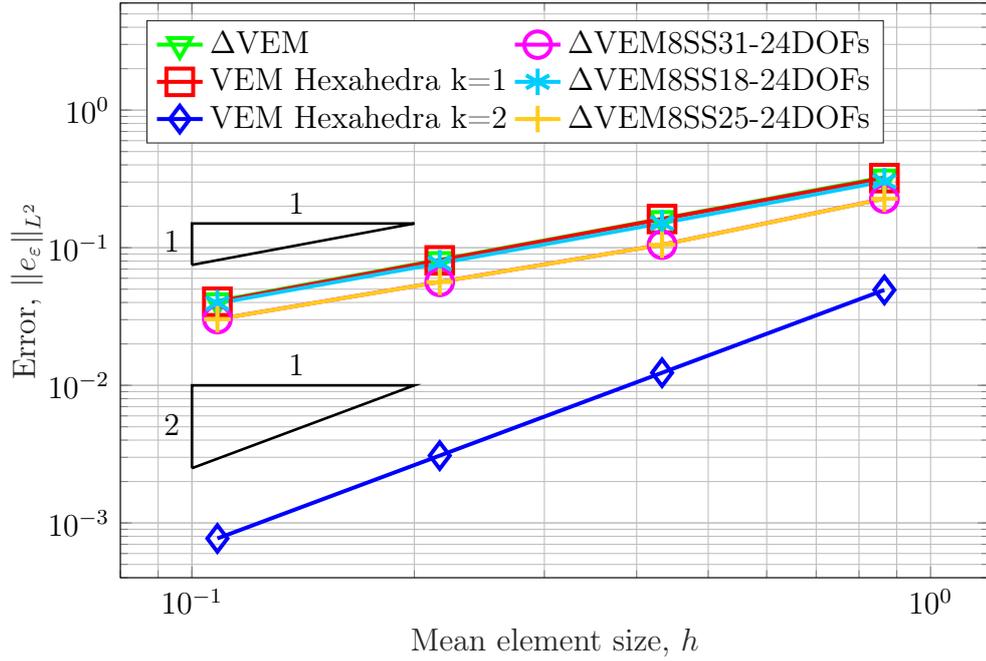


Figure 6.3: h-refinement convergence test for  $\Delta$ VEM and self-stabilized VEM numerically solving the elastostatic problem whose exact solution is a cubic polynomial field.

### 6.1.3. Quadratic polynomial displacement field

A similar study of 6.1.2 is conducted here, where the data is chosen so that the exact solution is a quadratic polynomial instead:

$$\begin{cases} -\mathbf{S}^T [\mathbf{D}(\mathbf{S}\mathbf{u})] = \mathbf{b} & \text{in } \Omega \\ \mathbf{u} = \bar{\mathbf{u}} & \text{on } \partial\Omega \end{cases} \quad (6.4)$$

where

- Lamé parameters:  $\lambda = 1$  and  $\mu = 1$  (Young's modulus  $E = 2.5$ , Poisson ratio  $\nu = 0.25$ )
- constant body forces  $\mathbf{b}$  in  $\Omega$

$$\mathbf{b}(x, y, z) = C \begin{Bmatrix} 2\lambda \\ 2\lambda \\ 2\lambda \end{Bmatrix}$$

- non-homogeneous Dirichlet boundary conditions  $\bar{\mathbf{u}}(\mathbf{x})$  given by the exact quadratic

polynomial solution

$$\mathbf{u}(x, y, z) = C \begin{pmatrix} x^2 - 2xy - 2xz \\ y^2 - 2yx - 2yz \\ z^2 - 2zy - 2zx \end{pmatrix} \quad (6.5)$$

where  $C$  is a constant set to 0.1.

The corresponding exact linear strains to 6.5 are

$$\boldsymbol{\varepsilon}(x, y, z) = C \begin{pmatrix} 2x - 2y - 2z \\ 2y - 2z - 2x \\ 2z - 2x - 2y \\ -2x - 2y \\ -2y - 2z \\ -2x - 2z \end{pmatrix} \quad (6.6)$$

Figure 6.4 shows a good agreement between the convergence lines and the expected slope. The  $L^2$ -norm strain errors for the  $k = 2$  standard VEM are not shown in the figure as their order of magnitude is of machine precision, meaning the method is able to capture in an *exact* way the analytical strain. Indeed, the strain model for second-order standard VEM is made by complete linear polynomials, and the exact strain in (6.6) is a subspace of such model.

$$\boldsymbol{\varepsilon}_{ex} \subset \mathbf{N}_{\boldsymbol{\varepsilon}}^{(VEM, k=2)} \hat{\boldsymbol{\varepsilon}}$$

The reason why self-stabilized VEM are not capturing the exact strain solution is due to the fact that the linear part of their strain models, despite being them quadratic, is not complete, after having required the divergence-free condition. The missing linear parameters imply that the exact strain field (6.6) is not a subspace of any of the models given by (5.29), (5.33) or (5.35), namely

$$\begin{aligned} \boldsymbol{\varepsilon}_{ex} &\not\subset \mathbf{N}_{\boldsymbol{\varepsilon}}^{(VEM8SS31-24DOFs)} \hat{\boldsymbol{\varepsilon}} \\ \boldsymbol{\varepsilon}_{ex} &\not\subset \mathbf{N}_{\boldsymbol{\varepsilon}}^{(VEM8SS25-24DOFs)} \hat{\boldsymbol{\varepsilon}} \\ \boldsymbol{\varepsilon}_{ex} &\not\subset \mathbf{N}_{\boldsymbol{\varepsilon}}^{(VEM8SS18-24DOFs)} \hat{\boldsymbol{\varepsilon}} \end{aligned}$$

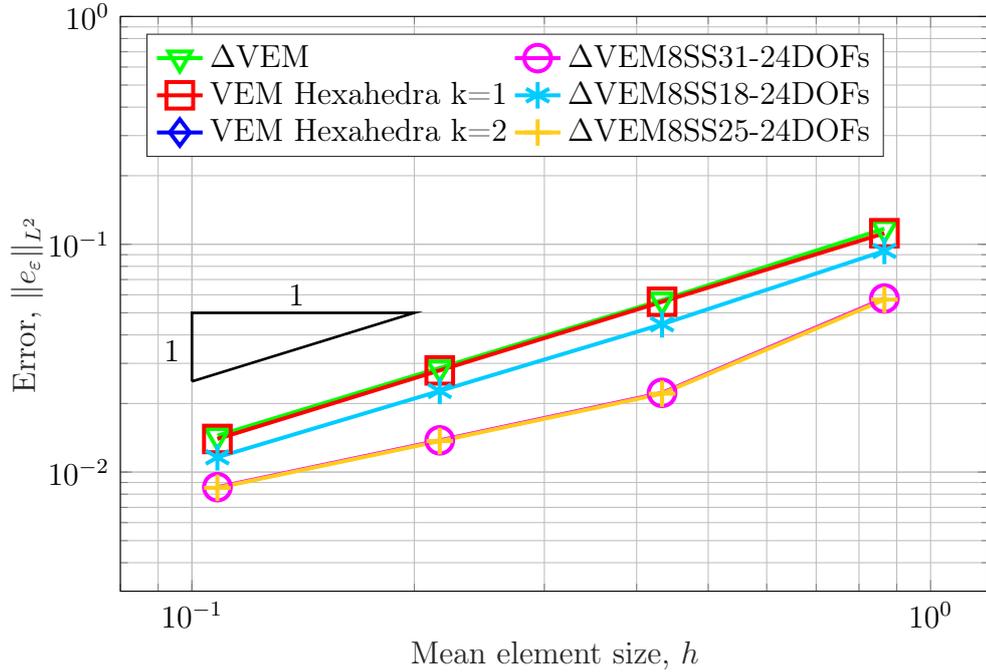


Figure 6.4: h-refinement convergence test for  $\Delta$ VEM and self-stabilized VEM numerically solving the elastostatic problem whose exact solution is a quadratic polynomial field.

#### 6.1.4. Incompressible limit

Some results are shown here for the VEM applied to elastostatics with nearly incompressible materials, i.e. when the Poisson ratio  $\nu \rightarrow 0$ . We premise that even though a special formulation should be appropriately implemented when dealing with incompressible media (see, e.g. [25]), promising results were obtained in [43] for the two-dimensional case, where it was shown that the self-stabilized version with additional 0<sup>th</sup>-order internal moments yielded to nearly-perfect convergence rates already with coarse meshes, while the standard VEM was badly behaving. However, it was also shown that the self-stabilized VE without additional 0<sup>th</sup>-order internal moments was exhibiting locking phenomena in a similar way to the standard VEM. In this section, some results on the nearly-incompressible limit are presented for the three-dimensional elastostatic problem, where no 0<sup>th</sup>-order internal moments seem to be allowed in the formulations, as already explained in Section 5.3.1.

The following problem is numerically solved

$$\begin{cases} -\mathbf{S}^T [\mathbf{D}(\mathbf{S}\mathbf{u})] = \mathbf{b} & \text{in } \Omega \\ \mathbf{u} = \mathbf{0} & \text{on } \partial\Omega \end{cases} \quad (6.7)$$

where trigonometric body forces  $\mathbf{b}(\mathbf{x})$  in  $\Omega$

$$\mathbf{b}(x, y, z) = \begin{pmatrix} -2\mu\pi^2 \cos(\pi y) \cos(\pi z) \sin(\pi y) \sin(\pi z) (3 \cos(2\pi x) - 2) \\ -2\mu\pi^2 \cos(\pi x) \cos(\pi z) \sin(\pi x) \sin(\pi z) (3 \cos(2\pi y) - 2) \\ 4\mu\pi^2 \cos(\pi x) \cos(\pi y) \sin(\pi x) \sin(\pi y) (3 \cos(2\pi z) - 2) \end{pmatrix}$$

The exact solution  $\mathbf{u}(\mathbf{x})$  of problem (6.7) is

$$\mathbf{u}(x, y, z) = \begin{pmatrix} \sin^2(\pi x) \sin(\pi y) \cos(\pi y) \sin(\pi z) \cos(\pi z) \\ \sin^2(\pi y) \sin(\pi x) \cos(\pi x) \sin(\pi z) \cos(\pi z) \\ -2 \sin^2(\pi z) \sin(\pi x) \cos(\pi x) \sin(\pi y) \cos(\pi y) \end{pmatrix}$$

to which the corresponding exact strains  $\boldsymbol{\varepsilon}(\mathbf{x})$  read

$$\boldsymbol{\varepsilon}(x, y, z) = \begin{pmatrix} 2\pi \cos(\pi x) \cos(\pi y) \cos(\pi z) \sin(\pi x) \sin(\pi y) \sin(\pi z) \\ 2\pi \cos(\pi x) \cos(\pi y) \cos(\pi z) \sin(\pi x) \sin(\pi y) \sin(\pi z) \\ -4\pi \cos(\pi x) \cos(\pi y) \cos(\pi z) \sin(\pi x) \sin(\pi y) \sin(\pi z) \\ -\pi \cos(\pi z) \sin(\pi z) (4 \cos(\pi x)^2 \cos(\pi y)^2 - 3 \cos(\pi x)^2 - 3 \cos(\pi y)^2 + 2) \\ \pi \cos(\pi x) \sin(\pi x) (2 \cos(\pi y)^2 \cos(\pi z)^2 - 3 \cos(\pi y)^2 + 1) \\ \pi \cos(\pi y) \sin(\pi y) (2 \cos(\pi x)^2 \cos(\pi z)^2 - 3 \cos(\pi x)^2 + 1) \end{pmatrix}$$

Lamé parameters are adjusted so that Poisson ratio is  $\nu = 0.45$  and  $\nu = 0.495$ :

- $\lambda = 9$  and  $\mu = 1$  ( $E = 2.9$ ,  $\nu = 0.45$ ), Figure 6.5
- $\lambda = 99$  and  $\mu = 1$  ( $E = 2.99$ ,  $\nu = 0.495$ ), Figure 6.6

While the locking phenomenon occurs for all the implemented virtual element versions, as their curves tend to flatten, the self-stabilized VEMs show better convergence rates, exhibiting a superior accuracy even with respect to  $k = 2$  standard VEM.

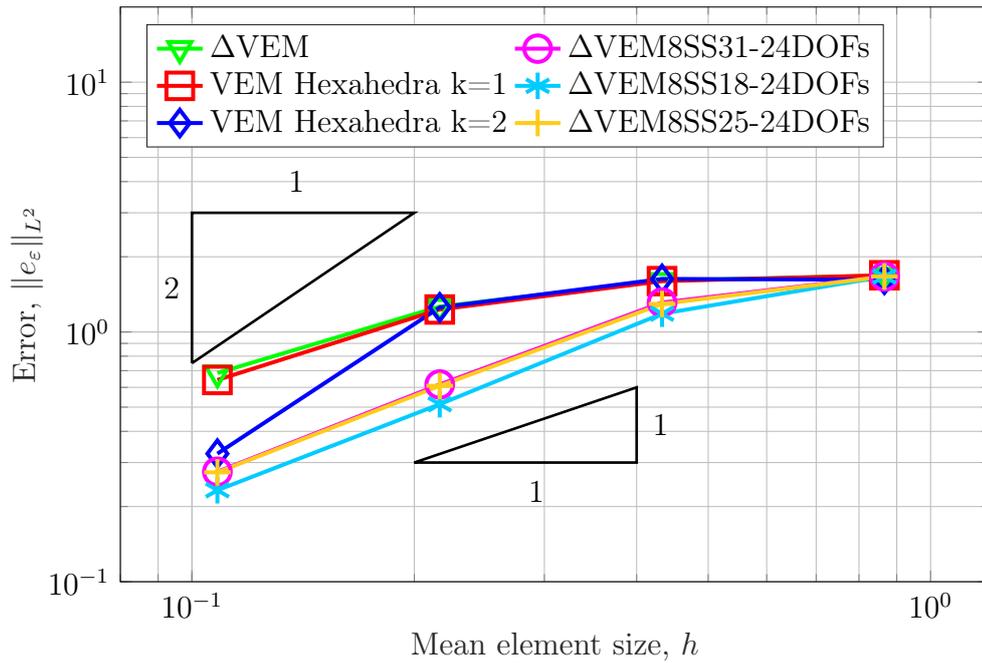


Figure 6.5: h-refinement convergence test for  $\Delta$ VEM and self-stabilized VEM numerically solving the elastostatic problem whose exact solution is trigonometric and Poisson ratio  $\nu = 0.45$ .

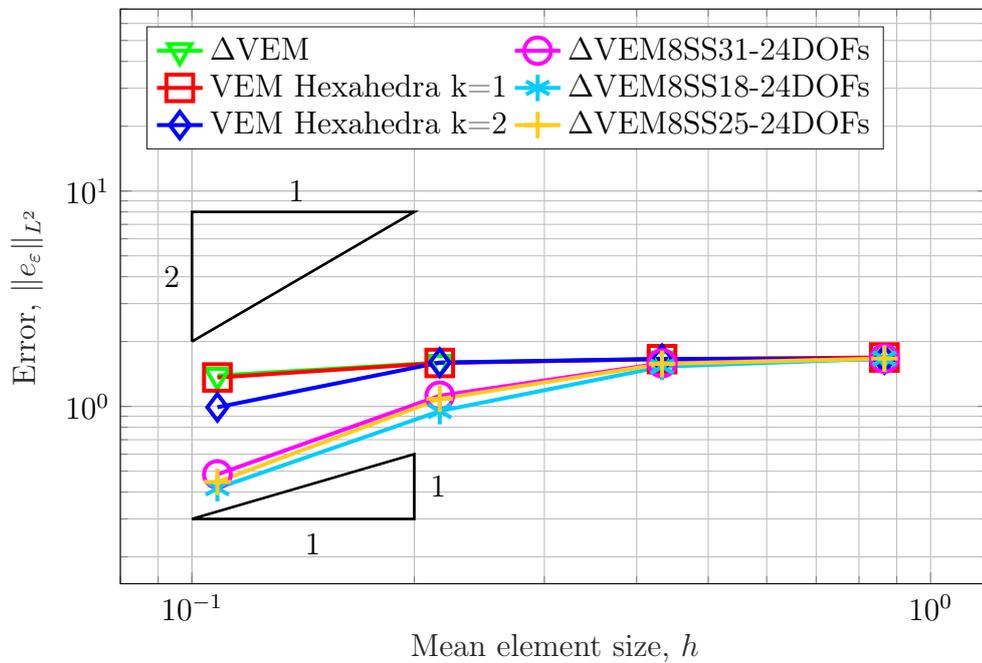


Figure 6.6: h-refinement convergence test for  $\Delta$ VEM and self-stabilized VEM numerically solving the elastostatic problem whose exact solution is trigonometric and Poisson ratio  $\nu = 0.495$ .

## 6.2. Cantilever beam

In this section, the self-stabilized VEM programs are employed to numerically solve the elastic problem of a cantilever beam under self weight, modelled as uniformly distributed load, as shown in Figure 6.7.

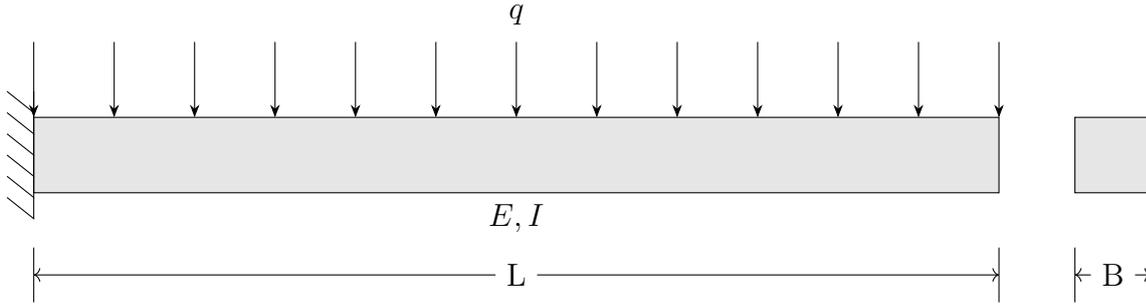


Figure 6.7: Cantilever beam with square cross section under uniformly distributed load.

The material properties and the selected data of the beam are:

- length of the beam  $L = 12$  m
- square cross section with edge  $B = 1$  m, so that the moment of inertia of area is  $I = \frac{B^4}{12} \approx 0.083333$  m<sup>4</sup>
- Young's modulus  $E = 32$  GPa
- Poisson ratio  $\nu = 0.2$
- material density  $\rho = 2400 \frac{\text{kg}}{\text{m}^3}$ , so that the distributed load  $q = \rho g B^2 = 23520 \frac{\text{N}}{\text{m}}$ , where  $g = 9.8 \frac{\text{m}}{\text{s}^2}$  is the gravitational acceleration.

Given the slenderness of the beam ( $\frac{L}{B} > 10$ ), Euler-Bernoulli analytical model ([60]) can be adopted to compare the maximum vertical displacement shown by the numerical VEM solution. According to Euler-Bernoulli theory for the structural problem in Figure 6.7 the maximum vertical displacement is given by

$$\delta_{\max} = \frac{qL^4}{8EI} \approx 0.02286 \text{ m} \quad (6.8)$$

The maximum vertical displacement  $\delta_{\max}$  found through the three self-stabilized VEM programs with the mesh shown in Figure 6.8 is reported in Table 6.1

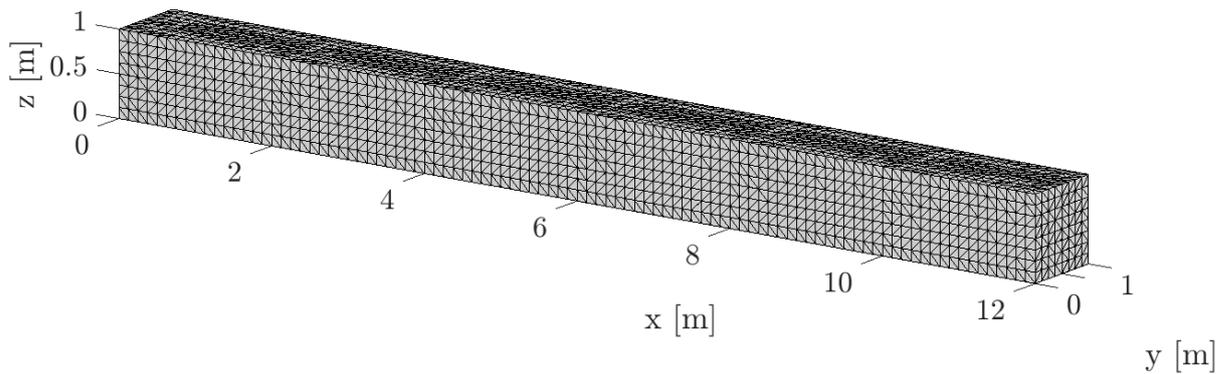


Figure 6.8: Deltahedral mesh adopted for a cantilever beam, with 23571 DOFs and 6144 elements. The clamped end is on the  $yz$  plane at  $x = 0$ .

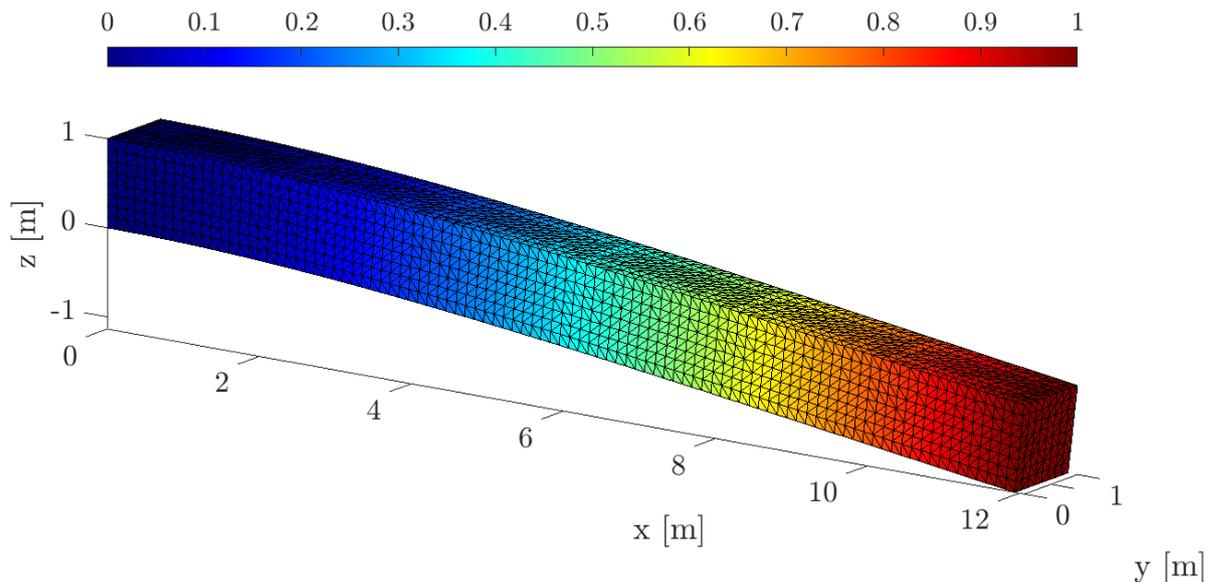


Figure 6.9: Numerical VEM solution obtained with  $\Delta$ VEM8SS31-24DOFs for the cantilever beam under self-weight. The color represents the vertical displacement and it is normalized with respect to the maximum value  $\delta_{\max}$ . The deformed shape is scaled by a factor of 50.

Method	$\delta_{\max}$ [m]	Relative error [%]
$\Delta$ VEM8SS31-24DOFs	0.02273	0.56
$\Delta$ VEM8SS25-24DOFs	0.02274	0.53
$\Delta$ VEM8SS18-24DOFs	0.02289	0.12

Table 6.1: Maximum vertical displacement found with self-stabilized VEM for a cantilever beam under self-weight and corresponding relative error with respect to Euler-Bernoulli beam theory prescription.

### 6.3. Simply supported plate

In the following section, the self-stabilized VEM programs are adopted to numerically solve the elastic problem of a simply supported square plate under self weight, modelled as uniformly distributed load, as shown in Figure 6.10.

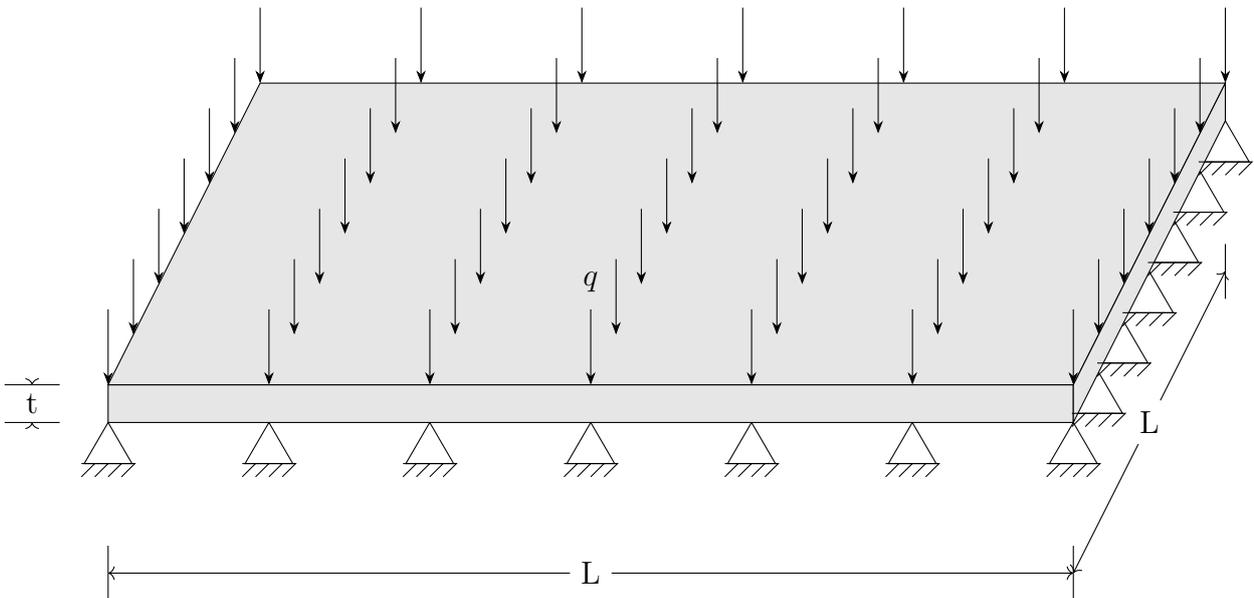


Figure 6.10: Simply supported square plate with constant thickness under uniformly distributed load.

The material properties and the selected data of the plate are:

- Young's modulus  $E = 32$  GPa

- Poisson ratio  $\nu = 0.2$
- length of the side of the plate  $L = 20$  m
- thickness of the plate  $t = 1$  m, so that the flexural stiffness is  $D = \frac{Et^3}{12(1-\nu^2)} \approx 2.77777$  GPa  $\cdot$  m<sup>3</sup>
- material density  $\rho = 2400 \frac{\text{kg}}{\text{m}^3}$ , so that the distributed load  $q = \rho g t = 23520 \frac{\text{N}}{\text{m}^2}$ , where  $g = 9.8 \frac{\text{m}}{\text{s}^2}$  is the gravitational acceleration.

Since the plate is thin, shear deformation can be neglected and Kirchhoff-Love plate theory ([61]) can be applied to obtain an analytical solution to compare the numerical displacements found through the VEM. Following the Navier approach, the following maximum displacement for a simply supported square plate is found

$$\delta_{\max} = \frac{16q}{\pi^6 D} \sum_{m=1,3,5,\dots}^{\infty} \sum_{n=1,3,5,\dots}^{\infty} \frac{L^4}{mn(m^2+n^2)^2} \sin\left(\frac{m\pi}{2}\right) \sin\left(\frac{n\pi}{2}\right) \approx 0.01344 \text{ m} \quad (6.9)$$

Table 6.2 shows the maximum vertical displacement  $\delta_{\max}$  found through the three self-stabilized VEM programs with the mesh of Figure 6.11.

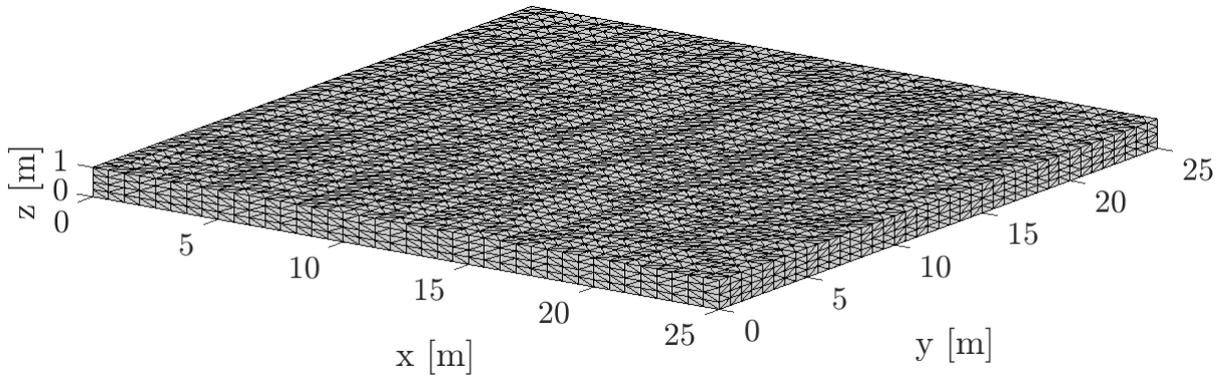


Figure 6.11: Deltahedral mesh adopted for a simply supported square plate, with 25215 DOFs and 6400 elements.

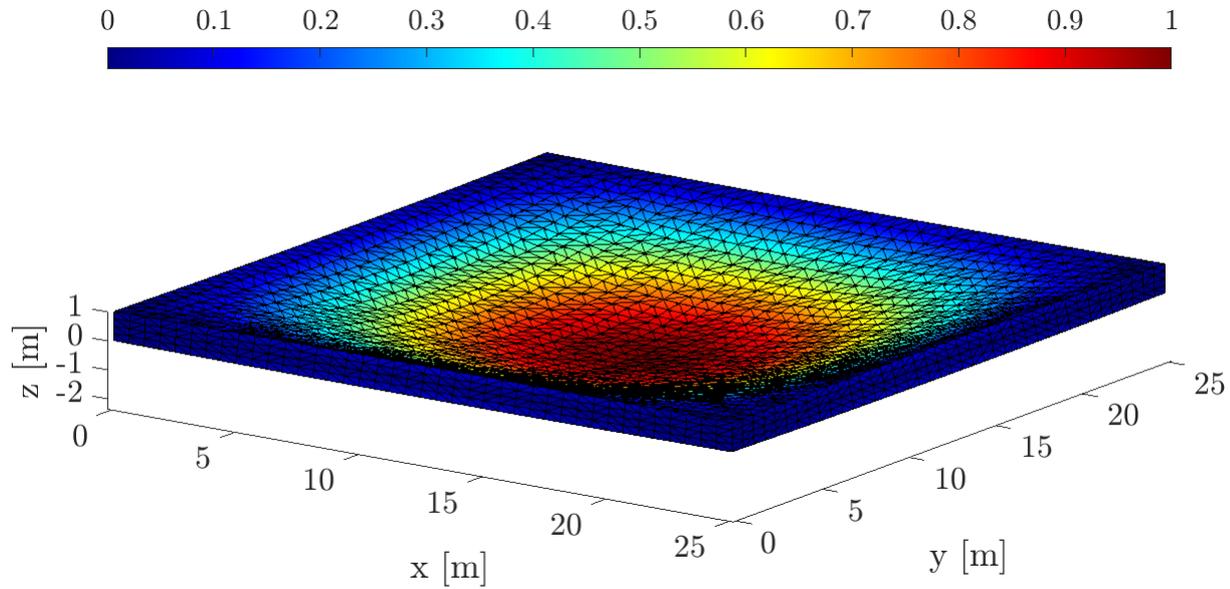


Figure 6.12: Numerical VEM solution obtained with  $\Delta$ VEM8SS31-24DOFs for the simply supported square plate under self-weight. The color represents the vertical displacement and it is normalized with respect to the maximum value  $\delta_{\max}$ . The deformed shape is scaled by a factor of 200.

Method	$\delta_{\max}$ [m]	Relative error [%]
$\Delta$ VEM8SS31-24DOFs	0.01183	11.96
$\Delta$ VEM8SS25-24DOFs	0.01199	10.74
$\Delta$ VEM8SS18-24DOFs	0.01338	0.41

Table 6.2: Maximum vertical displacement found with self-stabilized VEM for a simply supported square plate under self-weight and corresponding relative error with respect to Navier solution of the Kirchhoff-Love plate equation.

# 7 | Conclusions and future developments

## 7.1. Summary of results

In this Master thesis, the virtual element method (VEM) for three-dimensional elastostatics has been comprehensively presented, starting from a mixed variational formulation based on the three-field Hu-Washizu functional. The robustness of the solver was assessed through a MATLAB [40] implementation developed from scratch, emphasizing the strengths of the VEM with respect to standard finite elements. Specifically, virtual elements proved to be insensible to distortion, allowing to include in the mesh elements of diverse geometries, even non-convex ones.

Numerical tests in the form of  $h$ -refinement confirmed the expected order of convergence of the  $L^2$ -norm strain error -  $\mathcal{O}(h)$  for  $k = 1$  and  $\mathcal{O}(h^2)$  for  $k = 2$  VEM - regardless of the adopted elements. A slight variation in the accuracy was exhibited by exploiting different elements.

Subsequently, polyhedra with only triangular faces (*deltahedra*, hence  $\Delta$ VEM) were introduced to ease the VE projection operation on their boundary and three enhanced strain self-stabilized VEM formulations have been proposed. Starting from a  $k = 1$  8-nodes brick-type  $\Delta$ VEM, it was shown how a linear strain field would still require to implement a stabilization, even with the inclusion of internal moments DOFs. Therefore, the strain field has been further enhanced to a non-complete quadratic polynomial. In particular, 31-, 25- and 18-strain-parameters 8-nodes self-stabilized VEM (VEM8SS31-24DOFs, VEM8SS25-24DOFs and VEM8SS18-24DOFs) were developed and tested. An optimization to find the configurations of the element's geometry closest to a more-than-six-times degree of singularity was implemented to partially validate these new elements. Moreover, various  $h$ -refinement tests were performed, showing convergence rates of  $\mathcal{O}(h)$ , and compared to standard  $k = 1$  VEM: an increase in terms of accuracy of the self-stabilized VEM was appreciated. Additionally all the three new elements proved to be less sensitive

to locking phenomena induced by nearly-incompressible materials. Finally, two practical problems of a cantilever beam and a simply supported plate were solved through the self-stabilized VEM.

## 7.2. Outlook and future work

Further results originating from this Master thesis can be obtained in the context of standard VEM or self-stabilized VEM.

- Extension to three-dimensional elastodynamics. Coupling of the standard or self-stabilizing VEM with FEM environment through element aggregation, an exceptionally promising technique in the context of explicit dynamics where the numerical approximation is only conditionally stable.
- Implementation of material nonlinearities described by other constitutive laws (e.g. elastoplasticity) or geometric nonlinearities (finite strains).
- Further investigation on the self-stabilizing properties of the enhanced strain VEM.
- Analysis of VEM in the thin limit and comparison with shell FE, with potential coupling.

## Bibliography

- [1] B. Ahmad, A. Alsaedi, F. Brezzi, L. D. Marini, and A. Russo. Equivalent projectors for virtual element methods. *Computers & Mathematics with Applications*, 66(3): 376–391, 2013.
- [2] S. N. Alvarez, V. Bokil, V. Gyrya, and G. Manzini. The virtual element method for resistive magnetohydrodynamics. *Computer Methods in Applied Mechanics and Engineering*, 381:113815, 2021.
- [3] O. Andersen, H. M. Nilsen, and X. Raynaud. Virtual element method for geomechanical simulations of reservoir models. *Computational Geosciences*, 21:877–893, 2017.
- [4] P. F. Antonietti, P. Houston, and G. Pennesi. Fast numerical integration on polytopic meshes with applications to discontinuous galerkin finite element methods. *Journal of Scientific Computing*, 77(3):1339–1370, 2018.
- [5] E. Artioli, L. Beirão da Veiga, C. Lovadina, and E. Sacco. Arbitrary order 2d virtual elements for polygonal meshes: part i, elastic problem. *Computational Mechanics*, 60:355–377, 2017.
- [6] E. Artioli, S. De Miranda, C. Lovadina, and L. Patruno. A stress/displacement virtual element method for plane elasticity problems. *Computer Methods in Applied Mechanics and Engineering*, 325:155–174, 2017.
- [7] F. Auricchio, L. B. da Veiga, F. Brezzi, and C. Lovadina. Mixed finite element methods. *Encyclopedia of Computational Mechanics Second Edition*, pages 1–53, 2017.
- [8] B. Bauman, A. Sommariva, and M. Vianello. Compressed algebraic cubature over polygons with applications to optical design. *Journal of Computational and Applied Mathematics*, 370:112658, 2020.
- [9] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, and A. Russo.

- Basic principles of virtual element methods. *Mathematical Models and Methods in Applied Sciences*, 23(01):199–214, 2013.
- [10] L. Beirão da Veiga, 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(08):1541–1573, 2014.
- [11] L. Beirão da Veiga, C. Lovadina, and A. Russo. Stability analysis for the virtual element method. *Mathematical Models and Methods in Applied Sciences*, 27(13):2557–2594, 2017.
- [12] L. Beirão da Veiga, D. Mora, G. Rivera, and R. Rodríguez. A virtual element method for the acoustic vibration problem. *Numerische Mathematik*, 136(3):725–763, 2017.
- [13] D. Boffi, F. Brezzi, M. Fortin, et al. *Mixed finite element methods and applications*, volume 44. Springer, 2013.
- [14] A. Cangiani, G. Manzini, A. Russo, and N. Sukumar. Hourglass stabilization and the virtual element method. *International Journal for Numerical Methods in Engineering*, 102(3-4):404–436, 2015.
- [15] A. Cangiani, V. Gyrya, and G. Manzini. The nonconforming virtual element method for the stokes equations. *SIAM Journal on Numerical Analysis*, 54(6):3411–3435, 2016.
- [16] A. Chen and N. Sukumar. Stabilization-free virtual element method for plane elasticity. *Computers & Mathematics with Applications*, 138:88–105, 2023.
- [17] H. Chi, L. B. 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.
- [18] H. Chi, A. Pereira, I. F. Menezes, and G. H. Paulino. Virtual element method (vem)-based topology optimization: an integrated framework. *Structural and Multidisciplinary Optimization*, 62:1089–1114, 2020.
- [19] M. Cihan, B. Hudobivnik, F. Aldakheel, and P. Wriggers. 3d mixed virtual element formulation for dynamic elasto-plastic analysis. *Computational Mechanics*, 68:1–18, 2021.
- [20] B. Cockburn, G. E. Karniadakis, and C.-W. Shu. *Discontinuous Galerkin methods: theory, computation and applications*, volume 11. Springer Science & Business Media, 2012.

- [21] C. Comi and U. Perego. A variationally consistent generalized variable formulation of the elastoplastic rate problem. *Atti della Accademia Nazionale dei Lincei. Classe di Scienze Fisiche, Matematiche e Naturali. Rendiconti Lincei. Matematica e Applicazioni*, 2(2):177–190, 1991.
- [22] R. D. Cook et al. *Concepts and applications of finite element analysis*. John Wiley & sons, 2007.
- [23] L. Corradi. A displacement formulation for the finite element elastic-plastic problem. *Meccanica*, 18(2):77–91, 1983.
- [24] L. B. da Veiga and G. Manzini. A virtual element method with arbitrary regularity. *IMA Journal of Numerical Analysis*, 34(2):759–781, 2014.
- [25] L. B. Da Veiga, F. Brezzi, and L. D. Marini. Virtual elements for linear elasticity problems. *SIAM Journal on Numerical Analysis*, 51(2):794–812, 2013.
- [26] L. B. da Veiga, K. Lipnikov, and G. Manzini. *The mimetic finite difference method for elliptic problems*, volume 11. Springer, 2014.
- [27] L. B. Da Veiga, C. Lovadina, and D. Mora. A virtual element method for elastic and inelastic problems on polytope meshes. *Computer methods in applied mechanics and engineering*, 295:327–346, 2015.
- [28] L. B. Da Veiga, F. Brezzi, F. Dassi, L. D. Marini, and A. Russo. Virtual element approximation of 2d magnetostatic problems. *Computer Methods in Applied Mechanics and Engineering*, 327:173–195, 2017.
- [29] L. B. da Veiga, C. Lovadina, and G. Vacca. Divergence free virtual elements for the stokes problem on polygonal meshes. *ESAIM: Mathematical Modelling and Numerical Analysis*, 51(2):509–535, 2017.
- [30] L. B. da Veiga, F. Brezzi, F. Dassi, L. D. Marini, and A. Russo. A family of three-dimensional virtual elements with applications to magnetostatics. *SIAM Journal on Numerical Analysis*, 56(5):2940–2962, 2018.
- [31] L. B. Da Veiga, A. Russo, and G. Vacca. The virtual element method with curved edges. *ESAIM: Mathematical Modelling and Numerical Analysis*, 53(2):375–404, 2019.
- [32] L. B. da Veiga, F. Dassi, G. Manzini, and L. Mascotto. Virtual elements for maxwell’s equations. *Computers & Mathematics with Applications*, 116:82–99, 2022.
- [33] F. Dassi, A. Fumagalli, D. Losapio, S. Scialò, A. Scotti, and G. Vacca. The mixed

- virtual element method on curved edges in two dimensions. *Computer Methods in Applied Mechanics and Engineering*, 386:114098, 2021.
- [34] B. A. de Dios, K. Lipnikov, and G. Manzini. The nonconforming virtual element method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 50(3):879–904, 2016.
- [35] J. Djoko, B. Lamichhane, B. Reddy, and B. Wohlmuth. Conditions for equivalence between the hu–washizu and related formulations, and computational behavior in the incompressible limit. *Computer methods in applied mechanics and engineering*, 195(33-36):4161–4178, 2006.
- [36] A. L. Gain, C. Talischi, and G. H. Paulino. 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, 2014.
- [37] M. E. Gurtin. *An introduction to continuum mechanics*. Academic press, 1982.
- [38] R. J. Guyan. Reduction of stiffness and mass matrices. *AIAA journal*, 3(2):380–380, 1965.
- [39] P. Helnwein. Some remarks on the compressed matrix representation of symmetric second-order and fourth-order tensors. *Computer methods in applied mechanics and engineering*, 190(22-23):2753–2770, 2001.
- [40] T. M. Inc. Matlab version: 9.13.0 (r2022b), 2022. URL <https://www.mathworks.com>.
- [41] J. Jaśkowiec and N. Sukumar. High-order cubature rules for tetrahedra. *International Journal for Numerical Methods in Engineering*, 121(11):2418–2436, 2020.
- [42] A. Lamperti. Virtual elements based on the hu-washizu variational principle and their application in linear elastostatics and elastodynamics. Master’s thesis, Politecnico di Milano, 12 2021.
- [43] A. Lamperti, M. Cremonesi, U. Perego, A. Russo, and C. Lovadina. A hu–washizu variational approach to self-stabilized virtual elements: 2d linear elastostatics. *Computational Mechanics*, 71(5):935–955, 2023.
- [44] K. Lipnikov, G. Manzini, and M. Shashkov. Mimetic finite difference method. *Journal of Computational Physics*, 257:1163–1227, 2014. ISSN 0021-9991. doi: <https://doi.org/10.1016/j.jcp.2013.07.031>. URL <https://www.sciencedirect.com/>

- science/article/pii/S0021999113005135. Physics-compatible numerical methods.
- [45] X. Liu and Z. Chen. The nonconforming virtual element method for the navier-stokes equations. *Advances in Computational Mathematics*, 45:51–74, 2019.
- [46] J. Matousek. *Lectures on discrete geometry*, volume 212. Springer Science & Business Media, 2013.
- [47] G. H. Meisters. Polygons have ears. *The American Mathematical Monthly*, 82(6):648–651, 1975.
- [48] D. Mora, G. Rivera, and I. Velásquez. A virtual element method for the vibration problem of kirchhoff plates. *ESAIM: Mathematical Modelling and Numerical Analysis*, 52(4):1437–1456, 2018.
- [49] S. Mousavi and N. Sukumar. Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons. *Computational Mechanics*, 47:535–554, 2011.
- [50] A. Narkhede and D. Manocha. Fast polygon triangulation based on seidel’s algorithm. In *Graphics Gems V*, pages 394–397. Elsevier, 1995.
- [51] V. M. Nguyen-Thanh, X. Zhuang, H. Nguyen-Xuan, T. Rabczuk, and P. Wriggers. A virtual element method for 2d linear elastic fracture analysis. *Computer Methods in Applied Mechanics and Engineering*, 340:366–395, 2018.
- [52] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. Spatial tessellations: concepts and applications of voronoi diagrams. 2009.
- [53] K. Park, H. Chi, and G. H. Paulino. Numerical recipes for elastodynamic virtual element methods with explicit time integration. *International Journal for Numerical Methods in Engineering*, 121(1):1–31, 2020.
- [54] G. H. Paulino and A. L. Gain. Bridging art and engineering using escher-based virtual elements. *Structural and Multidisciplinary Optimization*, 51:867–883, 2015.
- [55] I. Perugia, P. Pietra, and A. Russo. A plane wave virtual element method for the helmholtz problem. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 50(3):783–808, 2016.
- [56] A. Russo and N. Sukumar. Quantitative study of the stabilization parameter in the virtual element method. *arXiv preprint arXiv:2304.00063*, 2023.

- [57] W. S. Slaughter. *The linearized theory of elasticity*. Springer Science & Business Media, 2012.
- [58] A. Sommariva and M. Vianello. Tetrafreeq: tetrahedra-free quadrature on polyhedral elements. *Applied Numerical Mathematics*, 2023.
- [59] O. J. Sutton. The virtual element method in 50 lines of matlab. *Numerical Algorithms*, 75:1141–1159, 2017.
- [60] S. Timoshenko. *Strength of materials*. 1930.
- [61] S. Timoshenko, S. Woinowsky-Krieger, et al. *Theory of plates and shells*, volume 2. McGraw-hill New York, 1959.
- [62] M. J. Turner, R. W. Clough, H. C. Martin, and L. J. Topp. Stiffness and deflection analysis of complex structures. *Journal of the Aeronautical Sciences*, 1956.
- [63] F. Wang and H. Wei. Virtual element methods for the obstacle problem. *IMA Journal of Numerical Analysis*, 40(1):708–728, 2020.
- [64] F. Wang and J. Zhao. Conforming and nonconforming virtual element methods for a kirchhoff plate contact problem. *IMA Journal of Numerical Analysis*, 41(2): 1496–1521, 2021.
- [65] K. Washizu. *Variational Methods in Elasticity and Plasticity*. Division 1 : solid and structural mechanics. Elsevier Science & Technology, 1974. ISBN 9780080176536. URL <https://books.google.it/books?id=jpseAQAAIAAJ>.
- [66] P. Wriggers, W. T. Rust, and B. Reddy. A virtual element method for contact. *Computational Mechanics*, 58:1039–1050, 2016.
- [67] P. Wriggers, B. D. Reddy, W. Rust, and B. Hudobivnik. Efficient virtual element formulations for compressible and incompressible finite deformations. *Computational Mechanics*, 60:253–268, 2017.
- [68] J. Zhao, S. Chen, and B. Zhang. The nonconforming virtual element method for plate bending problems. *Mathematical Models and Methods in Applied Sciences*, 26(09):1671–1687, 2016.
- [69] J. Zhao, B. Zhang, S. Chen, and S. Mao. The morley-type virtual element for plate bending problems. *Journal of Scientific Computing*, 76(1):610–629, 2018.
- [70] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The finite element method: its basis and fundamentals*. Elsevier, 2005.

# A | Gauss-Lobatto quadrature rule

This appendix presents the *Gauss-Lobatto integration rule* for the evaluation of one-dimensional integrals. The virtual element method of order  $k$  in three dimensions requires the projection of the shape functions onto the space of polynomials of degree  $k$  over the faces of the mesh, as described in Section 2.3. The right hand side of the algebraic system to perform such operation requires the integration of polynomials up to degree  $2k - 1$  over the skeleton of the polyhedra, made up by the edges  $E$ . Gauss-Lobatto quadrature rule is chosen because it matches with the 2 vertex-type plus  $k - 1$  edge-type DOFs completely describing the polynomial of order  $k$  representing the unknown field on the boundary, and allow to integrate *exactly* (up to machine precision) a polynomial of degree  $(k - 1) + k = 2k - 1$ , resulting from the product of the polynomial restriction on the edges of the shape functions and the derivatives of a scaled monomial of degree  $k$ .

The  $n - 2$  internal points of  $n^{\text{th}}$ -point Gauss-Lobatto quadrature rule over the standard integration interval  $[-1, 1]$  can be found as the stationary points of the  $(n - 1)^{\text{th}}$  *Legendre polynomials* (Figure A.1). These functions can be defined recursively as

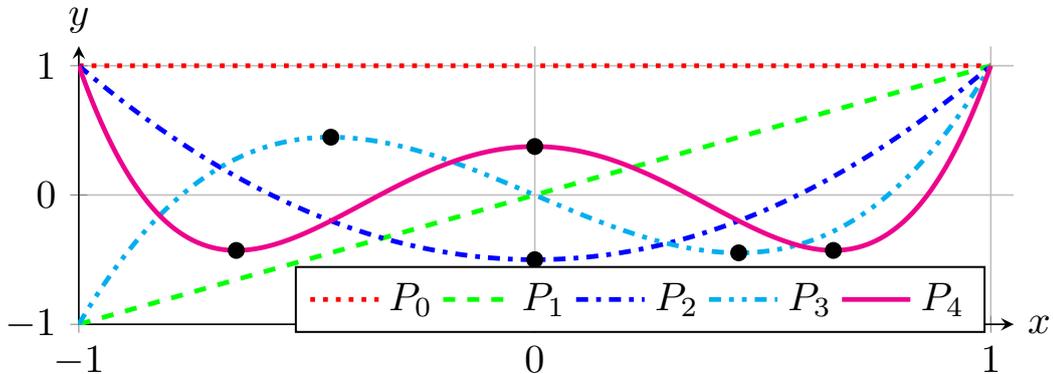


Figure A.1: Legendre polynomials up to degree 4.

$$\begin{aligned}
P_0(x) &= 1 \\
P_1(x) &= x \\
&\vdots \\
P_n(x) &= \frac{2n-1}{n}xP_{n-1}(x) - \frac{n-1}{n}P_{n-2}(x)
\end{aligned} \tag{A.1}$$

The derivatives of the Legendre polynomials can be computed recursively as

$$P'_n(x) = \frac{nxP_n(x) - nP_{n-1}(x)}{x^2 - 1} \tag{A.2}$$

To find the stationary points  $X_i$  of the Legendre polynomials in (A.1) one can easily set to zero (A.2). The weights  $W_i$  are then computed according to

$$W_i = \frac{2}{n(n-1)[P_{n-1}(X_i)]^2} \tag{A.3}$$

so that the Gauss-Lobatto rule for the integration of a function  $f(X)$  over  $[-1, 1]$  reads

$$\int_{-1}^1 f(X) dX \approx \sum_{i=1}^n f(X_i)W_i \tag{A.4}$$

The points  $X$  and weights  $W$  of Gauss-Lobatto quadrature rules up to 7 points over the standard interval  $[-1, 1]$  are reported in Table A.1. To find the Gauss-Lobatto integration points and weights for a general interval  $[a, b]$ , introducing the simple linear map

$$x_i = \frac{b-a}{2}X_i + \frac{a+b}{2} \tag{A.5}$$

one obtains

$$\int_a^b f(x) dx \approx \frac{b-a}{2} \sum_{i=1}^n \left[ f\left(\frac{b-a}{2}X_i + \frac{a+b}{2}\right) W_i \right] \tag{A.6}$$

where the weights are easily recognized as

$$w_i = \frac{b-a}{2}W_i \tag{A.7}$$

$n$	$X_i$	$W_i$
2	$\pm 1$	$\frac{1}{2}$
3	0	$\frac{4}{3}$
	$\pm 1$	$\frac{1}{3}$
4	$\pm \sqrt{\frac{1}{5}}$	$\frac{5}{6}$
	$\pm 1$	$\frac{1}{6}$
5	0	$\frac{32}{45}$
	$\pm \sqrt{\frac{3}{7}}$	$\frac{49}{90}$
	$\pm 1$	$\frac{1}{10}$
6	$\pm \sqrt{\frac{1}{3} - \frac{2\sqrt{7}}{21}}$	$\frac{14+\sqrt{7}}{30}$
	$\pm \sqrt{\frac{1}{3} + \frac{2\sqrt{7}}{21}}$	$\frac{14-\sqrt{7}}{30}$
	$\pm 1$	$\frac{1}{15}$
7	0	$\frac{256}{525}$
	$\pm \sqrt{\frac{5}{11} - \frac{2}{11}\sqrt{\frac{5}{3}}}$	$\frac{124+7\sqrt{15}}{350}$
	$\pm \sqrt{\frac{5}{11} + \frac{2}{11}\sqrt{\frac{5}{3}}}$	$\frac{124-7\sqrt{15}}{350}$
	$\pm 1$	$\frac{1}{21}$

Table A.1: Points and weights of the Gauss-Lobatto rule up to 7 points.



# B | Integration of monomials over polytopic domains

This appendix briefly presents the algorithm mentioned in Section 3.2.1 on the *quadrature-free* technique from [4]. Only aspects concerning the implementation will be discussed, leaving its theoretical background and derivation to [4]. The algorithm is a recursive call with a-priory exponential-time complexity with respect to the order of the monomial integrand, depending also on the number of lower order N-polytopes (faces, edges and points for polyhedra, edges and points for polygons). However, a clever choice of the local reference frame for each N-polytope allows to drastically reduce computation times. A scheme of the pseudo-algorithm is reported below. In the algorithm  $\mathbf{x}_0 = (x_{0,1}, \dots, x_{0,d})$

---

**Algorithm B.1** Integration of a monomial over a polytopic domain

---

$\mathcal{I}(N, \mathcal{E}, k_1, \dots, k_d) = \int_{\mathcal{E}} x_1^{k_1} \dots x_d^{k_d} d\sigma_N(x_1, \dots, x_d)$   
 1: **if**  $N = 0$  ( $\mathcal{E} = (v_1, \dots, v_d) \in \mathbb{R}^d$  is a point) **then**  
 2:     **return**  $\mathcal{I}(N, \mathcal{E}, k_1, \dots, k_d) = v_1^{k_1} \dots v_d^{k_d}$   
 3: **else if**  $1 \leq N \leq d - 1$  ( $\mathcal{E}$  is a point if  $d = 1$  or an edge if  $d = 2$  or a face if  $d = 3$ )  
    **then**  
 4:     **return**  $\mathcal{I}(N, \mathcal{E}, k_1, \dots, k_d) = \frac{1}{N + \sum_{n=1}^d k_n} (\sum_{i=1}^m d_i \mathcal{I}(N - 1, \mathcal{E}_i, k_1, \dots, k_d) +$   
         $+ x_{0,1} k_1 \mathcal{I}(N, \mathcal{E}, k_1 - 1, \dots, k_d) +$   
         $+ \dots +$   
         $+ x_{0,d} k_d \mathcal{I}(N, \mathcal{E}, k_1, \dots, k_d - 1))$   
 5: **else if**  $N = d$  ( $\mathcal{E}$  is an int'val if  $d = 1$  or a p.gon if  $d = 2$  or a p.hedron if  $d = 3$ )  
    **then**  
 6:     **return**  $\mathcal{I}(N, \mathcal{E}, k_1, \dots, k_d) = \frac{1}{N + \sum_{n=1}^d k_n} (\sum_{i=1}^m b_i \mathcal{I}(N - 1, \mathcal{E}_i, k_1, \dots, k_d))$   
 7: **end if**

---

is the origin of the reference system for the N-polytope  $\mathcal{E}$  called as input parameter by the algorithm, and  $d_i$  is the algebraic distance between the sub  $(N - 1)$ -polytope  $\mathcal{E}_i$  and  $\mathbf{x}_0$ . The choice of  $\mathbf{x}_0$  is critical as if selected such that one or more of its coordinates are 0, then the recursion calls for those components are not needed. In the implemented version this should be extensively exploited, considering also the degree of the variables making up the monomial being integrated: the coordinate of  $\mathbf{x}_0$  being selected to 0 is always looked

for when the corresponding exponent of the monomial is the highest. The situations in which faces or edges are parallel to the global axes are also taken into account, through a proper tolerance: in these cases there is no arbitrariness in the choice of  $\mathbf{x}_0$ .

## List of Figures

1.1	Elastic boundary value problem. . . . .	6
1.2	Example of element diameter and barycentric local coordinates for a polyhedron. . . . .	12
1.3	Some hourglass modes for a cubic element. . . . .	19
2.1	Hanging nodes. . . . .	27
2.2	Pascal's triangle truncated at $k = 6$ . . . . .	29
2.3	First three layers of Pascal's pyramid. . . . .	30
2.4	Layers corresponding to $k = 3$ and $k = 4$ of Pascal's pyramid. . . . .	30
2.5	Coordinates transformation from 2D face barycentric scaled reference system to 3D scaled polyhedral reference system. . . . .	31
2.6	Local degrees of freedom for three virtual element spaces in a pentagon $F$ . . . . .	37
2.7	Virtual shape functions in $\mathbb{R}^2$ for $k = 1$ . . . . .	37
2.8	Virtual shape functions in $\mathbb{R}^2$ for $k = 2$ . . . . .	38
2.9	Virtual shape functions in $\mathbb{R}^2$ for $k = 3$ . . . . .	39
2.10	Projection $\Pi_{F,1}^\nabla$ of a shape function for a linear virtual element. . . . .	43
2.11	Projection $\Pi_{F,2}^\nabla$ of shape functions for a quadratic virtual element. . . . .	46
2.12	Virtual and virtual enhanced shape functions in $\mathbb{R}^2$ for $k = 2$ . . . . .	48
2.13	Local degrees of freedom for two virtual element spaces in a pentagonal wedge $P$ . . . . .	51
2.14	Shape function for a linear virtual element in $\mathbb{R}^3$ . . . . .	52
4.1	Exact trigonometric solution in a unit cube. . . . .	68
4.2	Exact trigonometric $\varepsilon_x$ strain in a unit cube. . . . .	69
4.3	Exact trigonometric $\varepsilon_y$ strain in a unit cube. . . . .	70
4.4	Exact trigonometric $\varepsilon_z$ strain in a unit cube. . . . .	70
4.5	Exact trigonometric $\gamma_{xy}$ strain and $\tau_{xy}$ stress in a unit cube. . . . .	71
4.6	Exact trigonometric $\gamma_{yz}$ strain and $\tau_{yz}$ stress in a unit cube. . . . .	71
4.7	Exact trigonometric $\gamma_{xz}$ strain and $\tau_{xz}$ stress in a unit cube. . . . .	72
4.8	Exact trigonometric $\sigma_x$ stress in a unit cube. . . . .	73

4.9	Exact trigonometric $\sigma_y$ stress in a unit cube. . . . .	73
4.10	Exact trigonometric $\sigma_z$ stress in a unit cube. . . . .	74
4.11	Polyhedral elements for the considered meshes of problem (4.2). . . . .	75
4.12	Tetrahedral mesh. . . . .	75
4.13	Hexahedral mesh. . . . .	76
4.14	Honeycomb-type mesh. . . . .	76
4.15	Truncated-octahedral mesh. . . . .	77
4.16	Mesh obtained by Voronoi tessellation. . . . .	77
4.17	Non-convex mesh. . . . .	78
4.18	$h$ -refinement convergence test for 1 <sup>st</sup> order VEM. . . . .	79
4.19	Numerical $k = 1$ VEM solution in a unit cube, 504 and 7811 tetrahedra. . . . .	80
4.20	Numerical $k = 1$ VEM solution in a unit cube with 16, 512 and 4096 brick elements. . . . .	81
4.21	Numerical $k = 1$ VEM solution in a unit cube, 100 and 1000 Voronoi cells. . . . .	82
4.22	Numerical $k = 1$ VEM normal stresses in a unit cube with 1000 Voronoi cells. . . . .	83
4.23	Numerical $k = 1$ VEM shear stresses in a unit cube with 1000 Voronoi cells. . . . .	84
4.24	$h$ -refinement convergence test for 2 <sup>nd</sup> order VEM. . . . .	85
5.1	Deltahedral mesh obtained from Voronoi tessellation. . . . .	88
5.2	Triangulation of a simple polygon via the ear clipping method. . . . .	91
5.3	Fan triangulation of a convex polygon. . . . .	91
5.4	Configuration corresponding to the closest-to-zero eigenvalue of the stiffness matrix generated by 31-parameters strain field. . . . .	105
5.5	Configuration corresponding to the closest-to-zero eigenvalue of the stiffness matrix generated by 25-parameters strain field. . . . .	107
5.6	Configuration corresponding to the closest-to-zero eigenvalue of the stiffness matrix generated by 18-parameters strain field. . . . .	108
6.1	Deltahedral mesh obtained from hexahedral brick-type mesh. . . . .	110
6.2	$h$ -refinement convergence test for $\Delta$ VEM and SS-VEM, trigonometric solution. . . . .	111
6.3	$h$ -refinement convergence test for $\Delta$ VEM and SS-VEM, cubic polynomial solution. . . . .	113
6.4	$h$ -refinement convergence test for $\Delta$ VEM and SS-VEM, quadratic polynomial solution. . . . .	115
6.5	$h$ -refinement convergence test for $\Delta$ VEM and SS-VEM, Poisson ratio $\nu = 0.45$ . . . . .	117

6.6	$h$ -refinement convergence test for $\Delta$ VEM and SS-VEM, Poisson ratio $\nu = 0.495$ . . . . .	117
6.7	Cantilever beam with square cross section under uniformly distributed load. . . . .	118
6.8	Deltahedral mesh adopted for a cantilever beam. . . . .	119
6.9	Self-stabilized VEM solution for a cantilever beam. . . . .	119
6.10	Simply supported square plate with constant thickness under uniformly distributed load. . . . .	120
6.11	Deltahedral mesh adopted for a simply supported square plate. . . . .	121
6.12	Self-stabilized VEM solution for a simply supported square plate. . . . .	122
A.1	Legendre polynomials up to degree 4. . . . .	131



# List of Tables

3.1	Computational cost of the entries of matrix $\mathbf{G}$ through sub-tetrahedralization and Gauss quadrature. . . . .	56
6.1	Virtual element maximum vertical displacement for a cantilever beam under self-weight. . . . .	120
6.2	Virtual element maximum vertical displacement for a simply supported square plate under self-weight. . . . .	122
A.1	Points and weights of the Gauss-Lobatto rule up to 7 points. . . . .	133



# List of Symbols

## Symbol Description

---

$A$	standard FEM assembly operator
$\mathbf{A}$	matrix involved to compute $\mathbf{C}$
$\mathbf{A}_1$	part of $\mathbf{A}$ obtained by integration over the boundary
$\mathbf{A}_2$	part of $\mathbf{A}$ obtained by integration over the element interior
$\mathbf{B}$	standard FEM compatibility matrix
$\mathbf{b}$	body forces vector
$\mathbf{b}^h$	body forces vector projected onto $\mathcal{P}_{k-2}$
$\hat{\mathbf{b}}^h$	vector gathering the coefficients of the polynomial projection $\mathbf{b}^h$
$\mathbf{b}_{F,i}$	right hand side of linear algebraic system to find the projection $\Pi_{F,k}^\nabla \varphi_i$
$\mathbf{C}$	compatibility matrix
$DOF$	degree of freedom
$\mathbf{D}$	elastic material stiffness matrix
$\mathbf{D}_H$	fictitious hourglass stiffness matrix
$E$	Young's modulus or generic edge
$ E $	length of an edge $E$
$\mathbf{E}$	elastic matrix to compute $\mathbf{K}_e^c$
$\tilde{\mathbf{E}}$	modified elastic matrix to compute $\tilde{\mathbf{K}}_e$
$e_{ijk}$	Levi-Civita symbol in three dimensions
$\ e_\varepsilon\ _{L^2}$	$L^2$ -norm strain error
$F$	generic polygon
$ F $	area of a polygon $F$
$\mathbf{F}$	global equivalent nodal forces vector
$\mathbf{F}_e$	local equivalent nodal forces vector
$\mathbf{F}_e^b$	part of $\mathbf{F}_e$ from body forces
$\mathbf{F}_e^p$	part of $\mathbf{F}_e$ from surface tractions

$\mathbf{f}$	vector to compute $\mathbf{F}_e$
$\mathbf{G}$	matrix involved to compute $\mathbf{C}$
$\mathbf{G}_F$	matrix of the linear algebraic system to find the projection $\Pi_{F,k}^\nabla \varphi_i$
$\mathbf{H}$	hourglass matrix
$h$	characteristic mesh size
$h_e$	element diameter
$h_F$	diameter of a polygon $F$
$h_P$	diameter of a polyhedron $P$
$\mathbf{I}$	identity matrix
$\mathbf{K}$	global stiffness matrix
$\mathbf{K}_e$	local stiffness matrix
$\mathbf{K}_e^c$	part of $\mathbf{K}_e$ consistent with displacement and strain models
$\mathbf{K}_e^s$	part of $\mathbf{K}_e$ for hourglass stabilization
$\tilde{\mathbf{K}}$	global enhanced strain stiffness matrix
$\tilde{\mathbf{K}}_e$	local enhanced strain stiffness matrix
$k$	order of accuracy of VEM
$\mathbf{L}_e$	global-to-local matrix involved in the assembly operator $\mathbf{A}_e$
$m_\alpha$	scaled monomials in 2D
$\mathbf{m}_k$	vector of scaled monomials in 2D up to degree $k$
$\hat{\mathbf{M}}$	matrix involved to compute $\mathbf{A}_2$
$\hat{\mathbf{M}}_i$	matrix gathering the coefficients of $\mu_i$ involved to compute $\mathbf{A}_2$
$\mathbb{N}$	matrix of the direction cosines of $\mathbf{n}$
$\mathbb{N}_F$	matrix of the outward normal to a face $F$ of a polyhedron
$\mathbb{N}_P$	matrix of the outward normal to the boundary of a polyhedron $\partial P$
$N_{DOF}$	number of local degrees of freedom
$N_E$	number of edges of the element
$N_F$	number of faces of the element
$N_V$	number of vertices of the element
$\mathbf{N}_k$	matrix gathering the scaled monomials $\mu_k$
$\mathbf{N}_u$	matrix of displacement shape functions
$\mathbf{N}_\varepsilon$	matrix of strain field model
$\mathbf{N}_\varepsilon^{en}$	matrix of enhanced strain field model
$\mathbf{N}_\varepsilon^g$	matrix of generalized strain field model

$N_\sigma$	matrix of stress field model
$n$	number of integration points
$n_e$	number of elements in a mesh
$n_k$	number of scaled monomials in 2D up to degree $k$
$n_\varepsilon$	number of parameters describing the strain field
$n_\sigma$	number of parameters describing the stress field
$n_u$	number of parameters describing the displacement field
$\mathbf{n}$	outward normal unit vector
$\mathbf{n}_E$	outward normal unit vector of a polygon in correspondence of edge $E$
$\mathcal{P}_h$	polyhedral mesh
$\mathcal{P}_k$	space of polynomials up to degree $k$
$P$	generic polyhedron
$p_k$	element belonging to $\mathcal{P}_k$
$\mathbf{p}$	surface tractions vector
$\hat{\mathbf{p}}$	vector of natural parameters
$\hat{\mathbf{p}}_{D+R}$	part of $\hat{\mathbf{p}}$ related to deformation and rigid body modes
$\hat{\mathbf{p}}_H$	part of $\hat{\mathbf{p}}$ related to hourglass modes
$\mathbf{S}$	compatibility differential operator
$\mathbf{S}^T$	equilibrium differential operator
$\mathbf{s}_i$	vector gathering the coefficients of the scaled monomials of $\Pi_{F,k}^\nabla \varphi_i$
$\mathbf{T}$	transformation matrix mapping $\hat{\mathbf{p}}$ into $\hat{\mathbf{u}}$
$\mathbf{T}_{D+R}$	part of $\mathbf{T}$ related to deformation and rigid body modes
$\mathbf{T}_H$	part of $\mathbf{T}$ related to hourglass modes
$\mathbf{U}$	vector of global displacement parameters
$\mathbf{U}_c$	part of $\mathbf{U}$ containing constrained displacement parameters
$\mathbf{U}_f$	part of $\mathbf{U}$ containing free displacement parameters
$\mathbf{u}$	displacement vector
$\mathbf{u}^h$	local approximate displacement vector
$\bar{\mathbf{u}}$	vector of imposed displacement on $\partial_u \Omega$
$\hat{\mathbf{u}}$	vector of local displacement parameters
$\hat{\mathbf{u}}_{D+R}$	part of $\hat{\mathbf{u}}$ responsible of deformation and rigid body modes
$\hat{\mathbf{u}}_H$	part of $\hat{\mathbf{u}}$ responsible of hourglass modes
$\mathbf{u}$	displacement vector

$V$	generic vertex
$\mathbf{V}_e$	eigenvector matrix
$V_k(F)$	virtual space of order $k$ embedded in a polygonal domain $F$
$V_k(P)$	virtual space of order $k$ embedded in a polyhedral domain $P$
$v$	element belonging to $V_k$
$\mathbf{v}_e$	eigenvector
$W_k(F)$	enhanced virtual space of order $k$ embedded in a polygonal domain $F$
$w$	element belonging to $W_k(F)$
$\mathbf{x}_F$	centroid of a polygon
$\mathbf{x}_G$	centroid of an element
$\mathbf{x}_P$	centroid of a polyhedron
$\boldsymbol{\alpha}$	multiindex
$ \boldsymbol{\alpha} $	order of multiindex $\boldsymbol{\alpha}$
$\gamma_{xy}$	shear strain in $z$ plane
$\gamma_{yz}$	shear strain in $x$ plane
$\gamma_{xz}$	shear strain in $y$ plane
$\boldsymbol{\varepsilon}$	strain vector or strain tensor
$\boldsymbol{\varepsilon}^h$	local approximate strain vector
$\boldsymbol{\varepsilon}_H$	hourglass strain vector
$\hat{\boldsymbol{\varepsilon}}$	vector of local strain parameters
$\hat{\boldsymbol{\varepsilon}}^{en}$	vector of enhanced local strain parameters
$\hat{\boldsymbol{\varepsilon}}^g$	vector of generalized local strain parameters
$\varepsilon_x$	normal strain along $x$
$\varepsilon_y$	normal strain along $y$
$\varepsilon_z$	normal strain along $z$
$\boldsymbol{\Lambda}$	diagonal matrix for stiffness matrix stabilization
$\boldsymbol{\Lambda}_e$	diagonal matrix gathering the eigenvalues of $\mathbf{K}_e^c$
$\lambda$	first Lamé constant
$\lambda_e$	eigenvalue
$\mu$	second Lamé constant
$\mu_{\boldsymbol{\alpha}}$	scaled monomials in 3D
$\boldsymbol{\mu}_k$	vector of scaled monomials in 3D up to degree $k$
$\nu$	Poisson ratio

$\nu_k$	number of scaled monomials in 3D up to degree $k$
$\Xi$	DOF operator
$\xi$	vector of local non-dimensional coordinates
$\xi_f$	vector of local non-dimensional coordinates for a polygon
$\Pi$	Hu-Washizu functional
$\Pi^h$	discrete Hu-Washizu functional
$\Pi_e$	local Hu-Washizu functional
$\Pi_e^h$	local discrete Hu-Washizu functional
$\Pi_{F,k}^\nabla$	projection operator onto $\mathcal{P}_k$ over the polygon $F$
$\sigma$	stress vector
$\sigma^h$	local approximate stress vector
$\hat{\sigma}$	vector of local stress parameters
$\sigma_x$	normal stress along $x$
$\sigma_y$	normal stress along $y$
$\sigma_z$	normal stress along $z$
$\tau_{xy}$	shear stress in $z$ plane
$\tau_{yz}$	shear stress in $x$ plane
$\tau_{xz}$	shear stress in $y$ plane
$\varphi$	general shape function
$\Omega$	generic or three-dimensional domain
$ \Omega $	volume of $\Omega$
$\Omega_e$	generic or three-dimensional domain of the element
$\partial\Omega$	boundary of $\Omega$
$\partial_u\Omega$	constrained boundary of $\Omega$
$\partial_p\Omega$	boundary of $\Omega$ subjected to surface tractions
$\partial\Omega_e$	boundary of $\Omega_e$
$\partial_u\Omega_e$	constrained boundary of $\Omega_e$
$\partial_p\Omega_e$	boundary of $\Omega_e$ subjected to surface tractions
$\omega$	skew-symmetric part of $\nabla\mathbf{u}$



## Acknowledgements

I would like to sincerely thank Professor Massimiliano Cremonesi for the opportunity to work on this stimulating advanced numerical method and having supported me in filling the gap between the areas of Civil and Mathematical Engineering. I am especially grateful for his willingness to discuss, even on matters that very often were well beyond the purpose of the thesis. A special mention goes to Professor Franco Dassi, who spent some time right away to give me valuable hints on the implementation. And thanks to the whole virtual elements research group, in particular those I had the chance to interact with: Dexin Sun and Professors Umberto Perego, Paola Antonietti, Alessandro Russo and Carlo Lovadina.

Un grosso grazie alla mia famiglia per avermi supportato in questi tre anni di Doppia Laurea (ma anche sempre) e a mio fratello Ruben.

Et un grand merci à Juliette aussi.

