Finite volume for stress analysis upon polygonal, unstructured and non conforming meshes

Víctor E. Cardoso, Salvador Botello, Francisco Zárate, Eugenio Oñate

Center for Research in Mathematics, CIMAT A.C., Computer Science, Jalisco S/N, Valenciana, Guanajuato, Gto. México, 36250

Abstract

Within this paper we discuss a numerical strategy to solve the elasticity problem upon unstructured and non conforming meshes, allowing all kinds of flat-faced elements (polygons in 2D and polyhedra in 3D). The core of the formulation relies on two numerical procedures 1) the Control Volume Function Approximation (CVFA), and 2) the polynomial interpolation in the neighborhood of the control volumes, which is used to solve the surface integrals resulting from applying the divergence theorem. By comparing the estimated stress against the analytical stress field of the well known test of an infinite plate with a hole, we show that this conservative approach is robust and accurate.

Keywords: stress analysis, elasticity, solid mechanics, CVFA, Finite Volume, unstructured mesh, non conforming mesh

1. Introduction

One of the main aims in engineering is creating tools, structures and systems to enhance the quality of life in our society. In the course of the creation process, the design stage is critical for the final outcome. During this stage the engineer have to predict the prototype response when interacting with the physical world. Many of the observed phenomena in the physical world, such as solid mechanics, fluid dynamics, heat diffusion, and others, can be described with Partial Differential Equations (PDEs) by assuming time and space as a continuum.

Computational Continuum Mechanics (CCM) is the area dedicated to develop numerical methods and heuristics to solve these PDEs. Most of the methods can be classified into these two families: 1) weighted residual and 2) conservative methods. The Galerkin formulations are popular and widely used weighted residual methods, such as the Finite Element Method (FEM), which is a well established technique in Computational Solid Mechanics (CSM). Alternatively, the Finite Volume Method (FV) and the Control Volume Function Approximation (CVFA) are common approaches of conservative methods. The main difference between both families is that weighted residuals methods do not conserve quantities locally, but globally instead, resulting in linear systems with commendable numerical properties (symmetrical and well-conditioned matrices, for example). Nevertheless, due to its conservative nature, the second group is more attractive for fluid structure interaction ([1, 2]) and multiphysics simulations ([3, 4]), where several PDE-solvers must be coupled. For that reason, in recent years FV has been subject of interest for solving CSM problems.

Most of the CSM non-linear strategies depend on the accuracy of the estimated stress field for the elasticity problem, such as those for plasticity and damage (see [5, 6]). Hereafter we refer as elasticity-solver to the numerical computation that calculates the displacement and stress fields for a given domain and boundary conditions.

In his influential papers, Oñate et al [7, 8], propose a FV format for structural mechanics based on triangular meshes, discussing the *cell vertex scheme*, the *cell centred finite volume scheme* and its corresponding mixed formulations, showing that the *cell centred* strategy produces the same symmetrical global stiffness matrix that FEM using linear triangular elements. Analogously, Bailey et al [9, 10], develop a similar approach, but using quadrilateral elements to produce *cell-centred* volumes. Even though, the shapes of the volumes in both formulations are completely defined by the FEM-like mesh (triangular or quadrilateral) and it is not possible to handle arbitrary polygonal shapes, as we might expect when receiving the mesh as a read-only input.

Slone et al [11] extends the investigation of [7] by developing a dynamic solver based on an implicit Newmark scheme for the temporal discretization, with the motivation of coupling an elasticity-solver with his multi-physics modelling software framework, for later application to fluid structure interaction.

Another remarkable algorithm is the proposed by Demirdzic et al [12, 13, 14, 15, 16] The numerical procedure consists in decoupling the strain term into the displacement Jacobian and its transpose in a *cell-centred* scheme. The Jacobian is implicitly estimated by approximating the normal component of each face as the finite difference with respect to the adjacent nodes, while the Jacobian transpose is an explicit average of Taylor approximations around the same adjacent nodes. This decoupling produces a smaller memory footprint than FEM because the stiffness matrix is the same for all the components. The solution is found by solving one component each iteration in a coordinate descent minimization. This line of work has shaped most of the state of the art techniques in FV for coupling elasticity-solvers to Computational Fluid Dynamics (CFD) via finite volume practices (usually associated to CFD), such as the schemes proposed by [17, 18, 19] However, this segregated algorithm may lead to slow convergence rates when processing non-linear formulations, for example, when it is required to remove the positive principal components of the stress tensor in phase-field damage formulations [20]. In addition, if some non-linear strategy requires multiple iterations of the linear elasticity-solver, such as finite increments in damage models, the nested iterations will increase the processing requirements for simple problems. To circumvent this drawback Cardiff et al [21] presents a fully block coupled direct solution procedure, which does not require multiple iterations, at expense of decomposing the displacement Jacobian of any arbitrary face into a) the Jacobian of the displacement normal component, b) the Jacobian of the displacement tangential component, c) the tangential derivative of the displacement normal component and d) the tangential derivative of the displacement tangential component. This decomposition complicates the treatment of the stress tensor in the iterative non-linear solvers mentioned before for plasticity and damage.

A generalized finite volume framework for elasticity problems on rectangular domains is proposed by Cavalcante et al [22]. They use higher order displacement approximations at the expense of fixed axis-aligned grids for discretization.

Nordbotten [23] proposes a generalization of the multi-point flux approximation

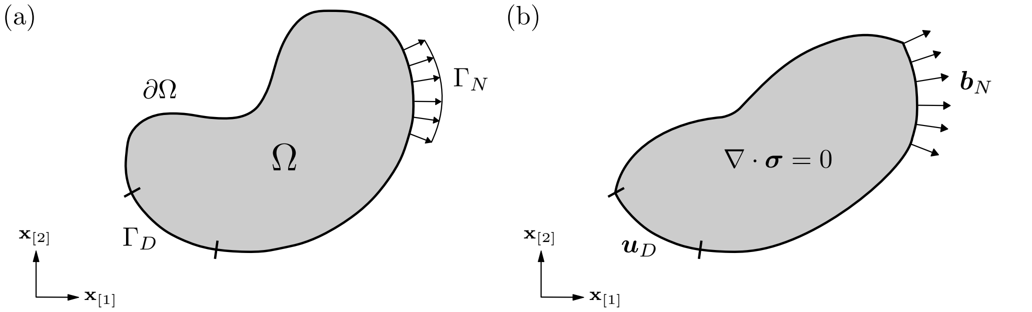

Figure 1: (a) The initial body, Ω, with its boundary conditions $\partial\Omega = \Gamma_N \cup \Gamma_D$. (b) The distorted body resulting from solving equilibrium in the elasticity equation, with the boundary conditions given by $\boldsymbol{b}_N$ and $\boldsymbol{u}_D$. The boundaries where there are not conditions indicated explicitly, correspond to Neumann conditions $\boldsymbol{b}_N = 0$.

(MPFA), which he names multi-point stress approximation (MPSA). The MPSA assembles unique linear expressions for the face average stress with more than two points in order to capture the tangential derivatives. The stress field calculated with this procedure is piece-wise constant.

In this work, we propose an elasticity-solver based on CVFA techniques (see [24, 25]), using piece-wise polynomial interpolators for solving the surface integrals on the volumes boundaries. The polynomials degree can be increased without incrementing the system degrees of freedom, which make this method more suitable for non-linear models and dynamic computations. Furthermore, this algorithm can handle polygonal/polyhedral, unstructured and non conforming meshes, and does not require the decomposition of the stress tensor.

2. Mathematical model

We consider an arbitrary body, $\Omega \in \mathbb{R}^{\mathrm{dim}}$, with boundary $\partial\Omega$. The displacement of a point $\mathbf{x} \in \Omega$ is denoted by $\boldsymbol{u}(\mathbf{x}) \in \mathbb{R}^{\mathrm{dim}}$. The subscript brackets $(\cdot)_{[]}$ indicates the component of the vector or matrix. By assuming small deformations and deformation gradients, the infinitesimal strain tensor, denoted $\boldsymbol{\varepsilon}(\mathbf{x}) \in \mathbb{R}^{\mathrm{dim}\times\mathrm{dim}}$, is given by

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla\boldsymbol{u} + [\nabla\boldsymbol{u}]^T\right). \tag{1}$$

Moreover, by considering isotropic elastic materials, the stress tensor, $\boldsymbol{\sigma}(\mathbf{x}) \in \mathbb{R}^{\mathrm{dim}\times\mathrm{dim}}$, is defined as

$$\boldsymbol{\sigma} = 2\mu\boldsymbol{\varepsilon} + \lambda tr(\boldsymbol{\varepsilon})\mathbf{I}, \tag{2}$$

where $\mathbf{I}$ is the identity matrix, $tr(\cdot)$ is the trace, and λ and μ are the Lamé parameters characterizing the material. These parameters are related with the Young's modulus, E, and the Poisson's ratio, ν, by the following equivalences

$$\mu = \frac{E}{2(1+\mu)}, \tag{3}$$

and

$$\lambda = \frac{\nu E}{(1+\nu)(1-\lambda_\nu)}, \tag{4}$$

where $\lambda_\nu = \nu$ for plane stress analysis, and $\lambda_\nu = 2\nu$ for plane strain and the 3D cases.

The strong form of the static linear elasticity equation is

$$\nabla \cdot \boldsymbol{\sigma} = 0, \tag{5}$$

The domain boundary $\partial\Omega = \Gamma_N \cup \Gamma_D$ is the union of the boundary with Neumann conditions, denoted Γ_N, and the boundary with Dirichlet conditions, denoted Γ_D. The equation (5) must be satisfied for the given forces $\boldsymbol{b}_N$ and displacements $\boldsymbol{u}_D$ along the boundary. The following equations remark these user defined boundary conditions,

$$\text{(S:BC)} \begin{cases} \boldsymbol{\sigma}\mathbf{n} = \boldsymbol{b}_N(\mathbf{x}), & \mathbf{x} \in \Gamma_N, \quad \text{(6a)} \\ \boldsymbol{u} = \boldsymbol{u}_D(\mathbf{x}), & \mathbf{x} \in \Gamma_D, \quad \text{(6b)} \end{cases}$$

The Figure 1 illustrates the initial body with the boundary conditions, and the distorted body after equilibrium is solved in the elasticity equation.

3. Numerical Method

On this section we go into the details of the numerical procedure by discussing 1) the discretization with CVFA, 2) the control volumes integration, 3) the subfaces integrals, 4) the simplex-wise polynomial approximation, 5) the pair-wise polynomial approximation, 6) the homeostatic splines used within the shape functions, 7) the linear system assembling, 8) how to impose boundary conditions, and 9) two special cases of the formulation.

For the sake of legibility, in some parts of the text we unfold the matrices only for the bidimensional case, but the very same procedures must be followed for the 3D case.

3.1. Discretization of domain into control volumes

The domain Ω is discretized into N control volumes, denoted V_i, using the Control Volume Function Approximation (CVFA) proposed by Li et al [24, 25]. The partition P_h of Ω is defined by

$$P_h = \bigcup_{i=1}^{N} V_i, \quad \text{with} \quad V_i \cap V_j = \emptyset, \quad i \neq j, \tag{7}$$

where the boundary of each control volume, ∂V_i, is composed by N_i flat faces, denoted e_{ij},

$$\partial V_i = \bigcup_{j=1}^{N_i} e_{ij}, \quad \text{with} \quad e_{ij} \cap e_{ik} = \emptyset, \quad j \neq k. \tag{8}$$

The Figure 2 illustrates the partition P_h of Ω into N control volumes defined in the equations (7) and (8). The Figure 3 shows a three dimensional control volume.

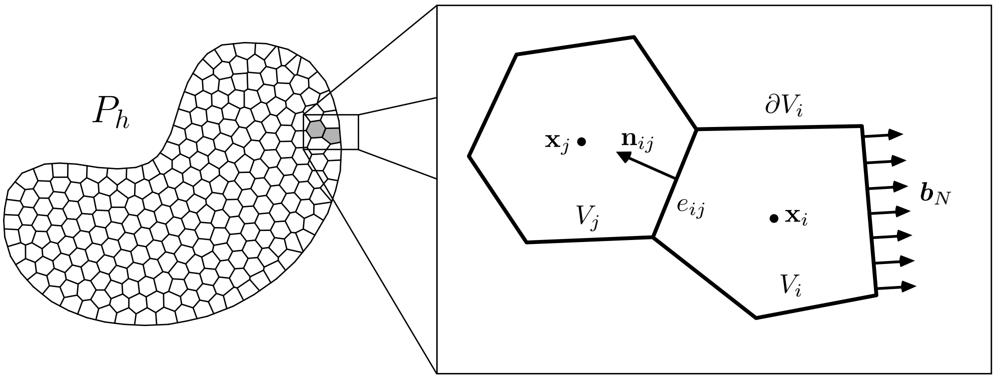

Figure 2: The partition P_h is the discretization of the domain Ω into N control volumes. The boundary of the control volumes, ∂V_i, is conformed by N_i flat faces, denoted e_{ij}. The unit vector $\mathbf{n}_{ij}$ is normal to the face e_{ij}. The faces of the volumes adjacent to the boundary Γ_N are integrated using the condition $\boldsymbol{b}_N$.

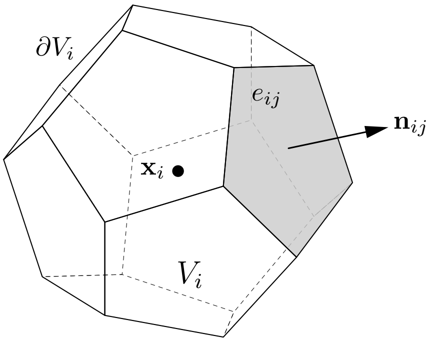

Figure 3: The boundary ∂V_i of the three dimensional control volume V_i is subdivided into N_i flat faces, denoted e_{ij}. The unit vector $\mathbf{n}_{ij}$ is normal to the face e_{ij}.

Every control volume V_i must have a calculation point

$$\mathbf{x}_i \in V_i \cup \partial V_i, \tag{9}$$

which is used to estimate the displacement field. Such a point is the base location to calculate the stiffness of the volume. In the volumes adjacent to the boundary Γ_D, it is convenient to establish the calculation point over the corresponding boundary face,

$$\mathbf{x}_i \in \partial V_i \cap \Gamma_D, \tag{10}$$

in order to set the Dirichlet condition directly on the point.

3.2. Control volumes integration

The elasticity equation (5) is now integrated over the control volume

$$\int_{V_i} \nabla \cdot \boldsymbol{\sigma} \; dV = 0, \tag{11}$$

using the divergence theorem we transform the volume integral into a surface integral over the volume boundary

$$\int_{\partial V_i} \boldsymbol{\sigma}\mathbf{n} \; dS = 0. \tag{12}$$

The evaluation of the surface integrals is based on the approximation of the displacement field inside the neighborhood of the volume, denoted $\mathcal{B}_i$,

$$\boldsymbol{u}_i(\mathbf{x}) = \sum_{q \in \mathcal{B}_i} \varphi_q \mathbf{x}_q, \tag{13}$$

making use of a group of piece-wise polynomial interpolators, denoted φ_q. We are going to discuss these interpolators later in this section.

For that reason, the displacement field is decoupled from the stress tensor by using the strain (1) and stress (2) definitions. Taking advantage of the stress tensor symmetry $\boldsymbol{\sigma}$, we rewrite the stress normal to the boundary as

$$\boldsymbol{\sigma}\mathbf{n} = \begin{bmatrix} \boldsymbol{\sigma}_{[11]} & \boldsymbol{\sigma}_{[12]} \\ \boldsymbol{\sigma}_{[12]} & \boldsymbol{\sigma}_{[22]} \end{bmatrix} \begin{bmatrix} \mathbf{n}_{[1]} \\ \mathbf{n}_{[2]} \end{bmatrix} = \begin{bmatrix} \mathbf{n}_{[1]} & & \mathbf{n}_{[2]} \\ & \mathbf{n}_{[2]} & \mathbf{n}_{[1]} \end{bmatrix} \begin{bmatrix} \boldsymbol{\sigma}_{[11]} \\ \boldsymbol{\sigma}_{[22]} \\ \boldsymbol{\sigma}_{[12]} \end{bmatrix} = \mathbf{T}\vec{\boldsymbol{\sigma}}, \tag{14}$$

where $\mathbf{T}$ is the face orientation matrix and $\vec{\boldsymbol{\sigma}}$ is the engineering stress vector. Developing the stress definition (2) component-wise we can decompose it into the constitutive matrix, denoted $\mathbf{D}$, and the engineering strain vector, denoted $\vec{\varepsilon}$, as follows

$$\begin{aligned} \vec{\boldsymbol{\sigma}} \quad &= \quad \begin{bmatrix} \boldsymbol{\sigma}_{[11]} \\ \boldsymbol{\sigma}_{[22]} \\ \boldsymbol{\sigma}_{[12]} \end{bmatrix} = \begin{bmatrix} 2\mu \; \varepsilon_{[11]} + \lambda \left(\varepsilon_{[11]} + \varepsilon_{[22]} \right) \\ 2\mu \; \varepsilon_{[22]} + \lambda \left(\varepsilon_{[11]} + \varepsilon_{[22]} \right) \\ 2\mu \; \varepsilon_{[12]} \end{bmatrix} & (15) \\ &= \quad \begin{bmatrix} (2\mu + \lambda) & \lambda & \\ \lambda & (2\mu + \lambda) & \\ & & \mu \end{bmatrix} \begin{bmatrix} \varepsilon_{[11]} \\ \varepsilon_{[22]} \\ 2\varepsilon_{[12]} \end{bmatrix} = \mathbf{D}\vec{\varepsilon}, & (16) \end{aligned}$$

then the components of the strain vector are retrieved from the equation (1), and it is decomposed into the matrix differential operator $\mathbf{S}$ and the displacement function $\boldsymbol{u}$.

$$\vec{\varepsilon} = \begin{bmatrix} \varepsilon_{[11]} \\ \varepsilon_{[22]} \\ 2\varepsilon_{[12]} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial \boldsymbol{u}_{[1]}}{\partial \mathbf{x}_{[1]}} \\ \dfrac{\partial \boldsymbol{u}_{[2]}}{\partial \mathbf{x}_{[2]}} \\ \dfrac{\partial \boldsymbol{u}_{[1]}}{\partial \mathbf{x}_{[2]}} + \dfrac{\partial \boldsymbol{u}_{[2]}}{\partial \mathbf{x}_{[1]}} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial}{\partial \mathbf{x}_{[1]}} & \\ & \dfrac{\partial}{\partial \mathbf{x}_{[2]}} \\ \dfrac{\partial}{\partial \mathbf{x}_{[2]}} & \dfrac{\partial}{\partial \mathbf{x}_{[1]}} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_{[1]} \\ \boldsymbol{u}_{[2]} \end{bmatrix} = \mathbf{S}\boldsymbol{u}, \tag{17}$$

Summarizing the equations (14), (16) and (17) we have

$$\boldsymbol{\sigma}\mathbf{n} = \mathbf{T}\vec{\boldsymbol{\sigma}} = \mathbf{TD}\vec{\varepsilon} = \mathbf{TDS}\boldsymbol{u}, \tag{18}$$

where $\mathbf{TDS}$ is the stiffness of the volume boundary.

Once the displacement field is decoupled, we rewrite the equation (12) as

$$\int_{\partial V_i} \mathbf{TDS}\boldsymbol{u} \; dS = 0. \tag{19}$$

Using the fact that the control volume boundary is divided into flat faces, as in equation (8), we split the integral (19) into the sum of the flat faces integrals

$$\sum_{j=1}^{N_i} \int_{e_{ij}} \mathbf{TDS}\boldsymbol{u} \; dS = 0. \tag{20}$$

Notice that the face orientation $\mathbf{T}$ along the flat face, denoted $\mathbf{T}_{ij}$, is constant. Furthermore, if the control volumes are considered to be made of a unique material and the flat faces to be formed by pairs of adjacent volumes, then the constitutive matrix $\mathbf{D}$ along the flat face, denoted $\mathbf{D}_{ij}$, is also considered constant. The matrix $\mathbf{D}_{ij}$ is estimated from the harmonic average of the Lamé parameters assigned to the adjacent volumes, where λ_i and μ_i are the properties of the volume V_i,

$$\mu_{ij} = \frac{2\mu_i\mu_j}{\mu_i + \mu_j} \quad \text{and} \quad \lambda_{ij} = \frac{2\lambda_i\lambda_j}{\lambda_i + \lambda_j}, \tag{21}$$

With $\mathbf{T}_{ij}$ and $\mathbf{D}_{ij}$ we simplify the equation (20) as

$$\sum_{j=1}^{N_i} \mathbf{T}_{ij}\mathbf{D}_{ij}\mathbf{H}_{ij} = 0, \tag{22}$$

where $\mathbf{H}_{ij}$ is the strain integral along the flat face e_{ij},

$$\mathbf{H}_{ij} = \int_{e_{ij}} \mathbf{S}\boldsymbol{u} \; dS. \tag{23}$$

The accuracy of the method depends on the correct evaluation of this integral.

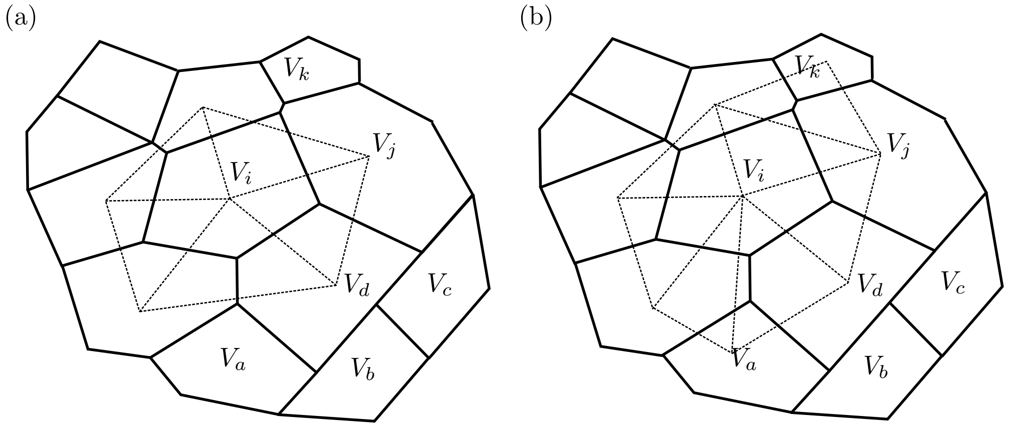

Figure 4: (a) The dotted line illustrates the triangulation of the calculation points of adjacent volumes to V_i, used by most of the FV methods. (b) The dotted line shows the simplices forming the piece-wise approximation used to solve the integrals $\mathbf{H}_{ij}$ of the control volume V_i.

3.3. Calculating face integrals

The surface integrals $\mathbf{H}_{ij}$ along the flat faces e_{ij} are calculated using an auxiliary piece-wise polynomial approximation of the displacement field. This approximation is based on the simplices (triangles in 2D or tetrahedra in 3D) resulting from the Delaunay triangulation of the calculation points $\mathbf{x}_i$ from the neighborhood of V_i. The Delaunay triangulation is the best triangulation for numerical interpolation, since it maximizes the minimum angle of the simplices, which means that its quality is maximized as well. We define the neighborhood $\mathcal{B}_i$ of volume V_i as the minimum set of calculation points $\mathbf{x}_j$ such that the simplices intersecting V_i does not change if we add another calculation point to the set. Observe that the neighborhood $\mathcal{B}_i$ does not always coincide with the set of calculation points in volumes adjacent to V_i, as in most of the FV formulations. Once the neighborhood $\mathcal{B}_i$ is triangulated, we ignore the simplices with angles smaller than 10 degrees, and the simplices formed outside the domain, which commonly appear in concavities of the boundary $\partial\Omega$. The local set of simplices resulting from the neighborhood of V_i is denoted P_α. The Figure 4 illustrates the difference between (a) the simplices resulting from the triangulation of the calculation points in adjacent volumes and (b) those resulting from the triangulation of the proposed neighborhood $\mathcal{B}_i$.

The face e_{ij} is subdivided into N_{ij} subfaces, denoted e_{ijk},

$$e_{ij} = \bigcup_{k=1}^{N_{ij}} e_{ijk}, \quad \text{with} \quad e_{ijk} \cap e_{ijl} = \emptyset, \quad l \neq k, \tag{24}$$

these subfaces result from the intersection between P_α and the control volume V_i. The Figure 4.b illustrates six key points of this approach, 1) the simplices are used to create a polynomial interpolation of $\boldsymbol{u}(\mathbf{x})$ over the boundary of the control volume, 2) most of the faces are intersected by several simplices, such faces must be divided into subfaces to be integrated, 3) some few faces are inside a single simplex, as illustrated in the face formed

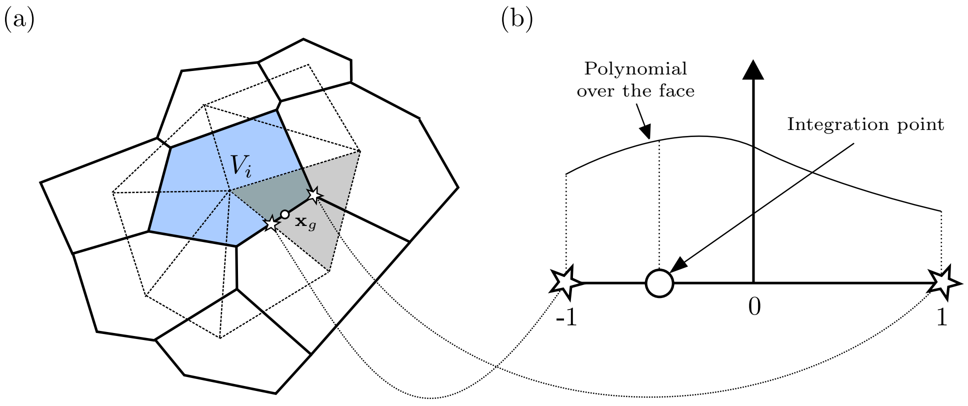

Figure 5: (a) The shaded volume V_i is being integrated. The integral over the subface e_{ijk} is calculated using the polynomial approximation of the shaded simplex. The integration point must be mapped to (b) the normalized space $[-1, 1]$ in order to use the Gauss- Legendre quadrature.

by V_i and V_k, 4) there are volumes that require information of non-adjacent volumes to calculate its face integrals, such as V_i requires V_k, 5) the dependency between volumes is not always symmetric, which means that if V_i requires V_k does not implies that V_k requires V_i, and 6) non conforming meshes are supported, as shown in the faces formed by V_a, V_b, V_c, V_d and V_j.

The integral (23) is now rewritten in terms of the subfaces

$$\mathbf{H}_{ij} = \sum_{k=1}^{N_{ij}} \int_{e_{ijk}} \mathbf{S}\boldsymbol{u} \; dS, \tag{25}$$

Each subface e_{ijk} is bounded by a simplex, where the displacement $\boldsymbol{u}_{ijk}$, and it derivatives, $(\mathbf{S}\boldsymbol{u})_{ijk}$, are a polynomial interpolation. Hence the integrals in equation (25) are solved exactly by using the Gauss-Legendre quadrature with the required number of integration points, denoted N_g, depending on the polynomial degree,

$$\int_{e_{ijk}} \mathbf{S}\boldsymbol{u} \; dS = \sum_{g=1}^{N_g} \; w_g (\mathbf{S}\boldsymbol{u})_{ijk}|_{\mathbf{x}_g}. \tag{26}$$

where w_g is the corresponding quadrature weight and $(\mathbf{S}\boldsymbol{u})_{ijk}|_{\mathbf{x}_g}$ is the strain evaluation of the Gauss point with the proper change of interval, denoted $\mathbf{x}_g$. The Figure 5 shows the change of interval required for a 2D face. A 3D face (a polygon) must be subdivided to be integrated with a triangular quadrature.

Most of the cases, the displacement $\boldsymbol{u}_{ijk}$ is interpolated inside the simplices, but in some geometrical locations these can not be created, in consequence, the displacement $\boldsymbol{u}_{ijk}$ is interpolated pair-wise using the volumes adjacent to the subface e_{ijk}. We discuss both strategies in the following subsections.

3.4. Simplex-wise polynomial approximation

In the general case, the simplices are formed by $(\dim{+}1)$ points. The points forming the simplex that is bounding the subface e_{ijk} are denoted $\mathbf{x}_q$, and its displacements $\boldsymbol{u}_q$.

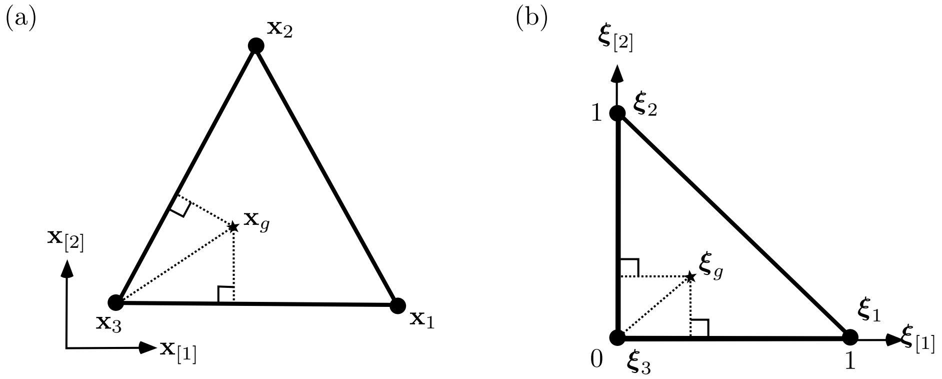

Figure 6: (a) The simplex formed by the points $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ in the original space contains an interior point $\mathbf{x}_g$ that is mapped to (b) $\boldsymbol{\xi}_g$ into the normalized 2D-simplex formed by the points $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_2$ and $\boldsymbol{\xi}_3$.

The shape functions used for the polynomial interpolation are defined into the normalized space. A point in such space is denoted $\boldsymbol{\xi}$, its d^{th} component is denoted $\boldsymbol{\xi}_{[d]}$, and the q^{th} point forming the simplex is expressed $\boldsymbol{\xi}_q$. The nodes of the normalized simplex are given by the origin, $\mathbf{0}$, and the standard basis vectors,

$$\boldsymbol{\xi}_q = \begin{cases} \mathbf{e}_q, & \text{for} \;\; q \in [1, \dim], \\ \mathbf{0}, & \text{if} \;\; q = \dim + 1 \end{cases} \tag{27}$$

where $\mathbf{e}_q$ is the q^{th} standard basis vector. The Figures 6 and 7 illustrates the original and the normalized simplices with the corresponding node numeration for 2D and 3D respectively.

The shape functions, denoted φ_q, are used to interpolate the displacement field inside the normalized simplex. Such functions are non-negative and are given by

$$\varphi_q(\boldsymbol{\xi}) = \begin{cases} \mathcal{P}_c\left(\boldsymbol{\xi}_{[q]}\right), & \text{if} \;\; q \in [1, \dim], \qquad \text{(28a)} \\ 1 - \sum_{d=1}^{\dim} \mathcal{P}_c\left(\boldsymbol{\xi}_{[d]}\right), & \text{for} \;\; q = \dim + 1, \qquad \text{(28b)} \end{cases}$$

where $\mathcal{P}_c(\cdot)$ is the *homeostatic spline*, which is the simplest polynomial defined in the interval $[0, 1]$ that have c derivatives equal to zero in the endpoints of the interval. We will discuss this spline later.

The set of shape functions is a partition of unity, which means that the sum of the functions in the set is equal to one into the interpolated domain

$$\sum_{q=1}^{\dim+1} \varphi_q(\boldsymbol{\xi}) = 1 \qquad \text{for any } \boldsymbol{\xi} \text{ inside the simplex,} \tag{29}$$

furthermore, these functions are equal to one in its corresponding node, which implies

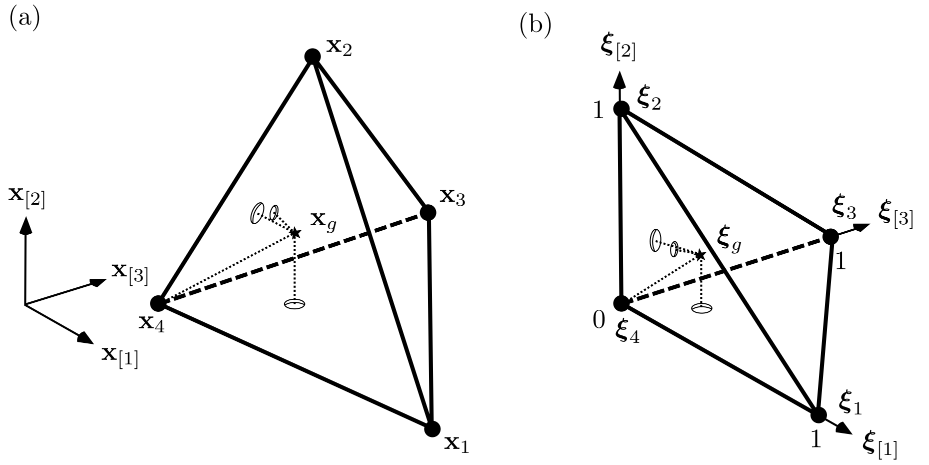

Figure 7: (a) The 3D-simplex formed by the points $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ and $\mathbf{x}_4$ in the original space contains an interior point $\mathbf{x}_g$ that is mapped to (b) $\boldsymbol{\xi}_g$ into the normalized 3D-simplex formed by the points $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_2$, $\boldsymbol{\xi}_3$ and $\boldsymbol{\xi}_4$.

that

$$\varphi_q(\boldsymbol{\xi}_q) = 1 \qquad \text{for any } \boldsymbol{\xi}_q \text{ forming the simplex,} \tag{30}$$

$$\varphi_q(\boldsymbol{\xi}_p) = 0 \qquad \text{for any } \boldsymbol{\xi}_p \neq \boldsymbol{\xi}_q \text{ forming the simplex,} \tag{31}$$

The gradients of the shape functions with respect to the normalized space are denoted $\nabla_{\boldsymbol{\xi}}\varphi_q$. The norm of the sum of such gradients is zero

$$\left\| \sum_{q=1}^{\dim+1} \nabla_{\boldsymbol{\xi}}\varphi_q\left(\boldsymbol{\xi}\right) \right\| = 0 \qquad \text{for any } \boldsymbol{\xi} \text{ inside the simplex,} \tag{32}$$

which means that there are not numerical artifacts into the strain field.

Any point inside the simplex can be formulated as a function of a point in the normalized space, $\boldsymbol{p}\left(\boldsymbol{\xi}\right)$, by using the shape functions and the points forming the simplex

$$\boldsymbol{p}\left(\boldsymbol{\xi}\right) = \sum_{q=1}^{\dim+1} \varphi_q\left(\boldsymbol{\xi}\right)\mathbf{x}_q, \tag{33}$$

In order to calculate the normalized point, denoted $\boldsymbol{\xi}_g$, associated to the integration point $\mathbf{x}_g = \boldsymbol{p}\left(\boldsymbol{\xi}_g\right)$, we use the shape functions definitions to rewrite the equation (33) in matrix form

$$\boldsymbol{p}\left(\boldsymbol{\xi}\right) = \underbrace{\begin{bmatrix} \mathbf{x}_{3[1]} \\ \mathbf{x}_{3[2]} \end{bmatrix} + \begin{bmatrix} \left(\mathbf{x}_{1[1]} - \mathbf{x}_{3[1]}\right) & \left(\mathbf{x}_{2[1]} - \mathbf{x}_{3[1]}\right) \\ \left(\mathbf{x}_{1[2]} - \mathbf{x}_{3[2]}\right) & \left(\mathbf{x}_{2[2]} - \mathbf{x}_{3[2]}\right) \end{bmatrix} \begin{bmatrix} \mathcal{P}_c\left(\boldsymbol{\xi}_{[1]}\right) \\ \mathcal{P}_c\left(\boldsymbol{\xi}_{[2]}\right) \end{bmatrix}}_{\text{2D case (triangle)}} \tag{34}$$

$$= \mathbf{x}_{(\dim+1)} + \mathbf{J}_{\Delta}\ \mathcal{P}_c\left(\boldsymbol{\xi}\right), \tag{35}$$

where $\mathcal{P}_c\left(\boldsymbol{\xi}\right)$ is the vector resulting from evaluating the spline for $\boldsymbol{\xi}$ component-wise, and $\mathbf{J}_{\triangle}$ is the distortion matrix given by the concatenation of the following column vector differences

$$\mathbf{J}_{\triangle} = \left[(\mathbf{x}_1 - \mathbf{x}_{(\mathrm{dim}+1)}), ..., (\mathbf{x}_{\mathrm{dim}} - \mathbf{x}_{(\mathrm{dim}+1)})\right] \tag{36}$$

Now, from equation (35) we retrieve the point $\mathbf{x}_g$ as

$$\mathbf{x}_g = \boldsymbol{p}\left(\boldsymbol{\xi}_g\right) = \mathbf{x}_{(\mathrm{dim}+1)} + \mathbf{J}_{\triangle}\ \mathcal{P}_c\left(\boldsymbol{\xi}_g\right), \tag{37}$$

and solving for $\boldsymbol{\xi}_g$ we obtain

$$\boldsymbol{\xi}_g = \mathcal{Q}_c\left(\left(\mathbf{J}_{\triangle}\right)^{-1}\left(\mathbf{x}_g - \mathbf{x}_{(\mathrm{dim}+1)}\right)\right), \tag{38}$$

where $\mathcal{Q}_c$ is the inverse function of $\mathcal{P}_c$ applied component-wise to the product of the matrix-vector operation.

Similar to the approximation in equation (33), within the simplex enclosing the subface e_{ijk}, the displacement field evaluated at $\mathbf{x}_g$ is defined as,

$$\boldsymbol{u}_{ijk}|_{\mathbf{x}_g} = \sum_{q=1}^{\mathrm{dim}+1} \varphi_q\left(\boldsymbol{\xi}_g\right)\ \boldsymbol{u}_q \tag{39}$$

Hence, when calculating the quadrature of equation (26), the strain evaluated at the integration point is given by

$$\begin{aligned}
(\mathbf{S}\boldsymbol{u})_{ijk}|_{\mathbf{x}_g} &= \sum_{q=1}^{\mathrm{dim}+1} \mathbf{S}\varphi_q(\boldsymbol{\xi}_g)\ \boldsymbol{u}_q, && (40)\\
&= \begin{bmatrix} \frac{\partial\varphi_1}{\partial\mathbf{x}_{[1]}} & & \frac{\partial\varphi_2}{\partial\mathbf{x}_{[1]}} & & \frac{\partial\varphi_3}{\partial\mathbf{x}_{[1]}} & \\ & \frac{\partial\varphi_1}{\partial\mathbf{x}_{[2]}} & & \frac{\partial\varphi_2}{\partial\mathbf{x}_{[2]}} & & \frac{\partial\varphi_3}{\partial\mathbf{x}_{[2]}} \\ \frac{\partial\varphi_1}{\partial\mathbf{x}_{[2]}} & \frac{\partial\varphi_1}{\partial\mathbf{x}_{[1]}} & \frac{\partial\varphi_2}{\partial\mathbf{x}_{[2]}} & \frac{\partial\varphi_2}{\partial\mathbf{x}_{[1]}} & \frac{\partial\varphi_3}{\partial\mathbf{x}_{[2]}} & \frac{\partial\varphi_3}{\partial\mathbf{x}_{[1]}} \end{bmatrix}_{|\mathbf{x}_g} \begin{bmatrix} \boldsymbol{u}_{1[1]} \\ \boldsymbol{u}_{1[2]} \\ \boldsymbol{u}_{2[1]} \\ \boldsymbol{u}_{2[2]} \\ \boldsymbol{u}_{3[1]} \\ \boldsymbol{u}_{3[2]} \end{bmatrix} && (41)\\
&= \mathbf{B}_{ijk}|_{\mathbf{x}_g}\quad \vec{\boldsymbol{u}}_{ijk}, && (42)
\end{aligned}$$

where $\mathbf{B}_{ijk}|_{\mathbf{x}_g}$ captures the deformation at $\mathbf{x}_g$, and $\vec{\boldsymbol{u}}_{ijk}$ is the vector with the concatenated displacement components of the points forming the simplex.

In order to calculate the deformation matrix $\mathbf{B}_{ijk}$, we require the derivatives of the shape functions with respect to $\mathbf{x}$, denoted $\nabla\varphi_q$. These derivatives are calculated by solving the linear systems resulting from the chain rule

$$\nabla_{\boldsymbol{\xi}}\varphi_q = \begin{bmatrix} \frac{\partial\varphi_q}{\partial\boldsymbol{\xi}_{[1]}} \\ \frac{\partial\varphi_q}{\partial\boldsymbol{\xi}_{[2]}} \end{bmatrix} = \begin{bmatrix} \frac{\partial\mathbf{x}_{[1]}}{\partial\boldsymbol{\xi}_{[1]}} & \frac{\partial\mathbf{x}_{[2]}}{\partial\boldsymbol{\xi}_{[1]}} \\ \frac{\partial\mathbf{x}_{[1]}}{\partial\boldsymbol{\xi}_{[2]}} & \frac{\partial\mathbf{x}_{[2]}}{\partial\boldsymbol{\xi}_{[2]}} \end{bmatrix} \begin{bmatrix} \frac{\partial\varphi_q}{\partial\mathbf{x}_{[1]}} \\ \frac{\partial\varphi_q}{\partial\mathbf{x}_{[2]}} \end{bmatrix} = \left(\nabla_{\boldsymbol{\xi}}\boldsymbol{p}\right)^T\ \nabla\varphi_q, \tag{43}$$

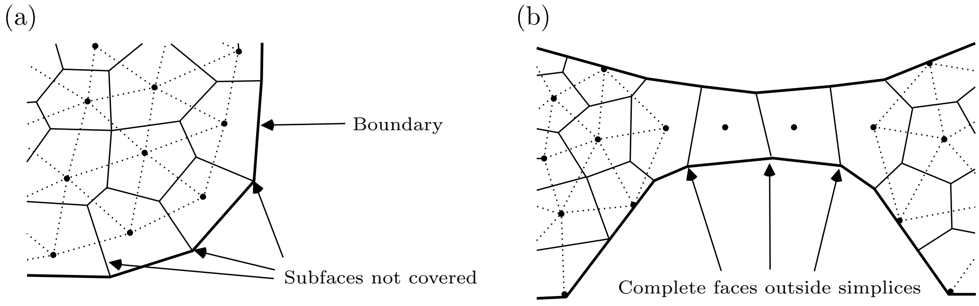

Figure 8: (a) When the calculation points of volumes contiguous to the boundary are in the interior of such volumes, there will arise subfaces next to the boundary that can not be covered by any simplex. (b) Portions of the mesh formed by a queue of aligned volumes do not allow the formation of simplices through that queue and there will be whole faces not covered by any simplex.

where $\nabla_{\boldsymbol{\xi}} \boldsymbol{p}$ is the geometric jacobian evaluated at $\boldsymbol{\xi}$. This jacobian relates both spaces, captures the distortion of the simplex, and is derivated from equation (33),

$$\nabla_{\boldsymbol{\xi}} \boldsymbol{p} = \sum_{q=1}^{\dim+1} \mathbf{x}_q \left(\nabla_{\boldsymbol{\xi}} \varphi_q\right)^T , \tag{44}$$

The gradients of the shape functions with respect to $\boldsymbol{\xi}$ inside the sum are obtained straightforward once we have the spline first derivative $\mathcal{P}'_c$. Notice that the geometric jacobian is equivalent to the distortion matrix $\mathbf{J}_\Delta$ if and only if the homeostatic spline is $\mathcal{P}_c(z) = z$.

3.5. Pair-wise polynomial approximation

Since we are not making any assumption about the volumes distribution through the mesh, neither about the internal location of its calculation points, then we have to deal with portions of the mesh that are no covered by any simplex. The Figure 8 illustrates the two most common cases. The first case takes place in meshes where the calculation points of volumes contiguous to the boundary are in the interior of such volumes, producing subfaces not intersected by any simplex, and the second case occurs when elongated sections of the domain are discretized with a queue of aligned volumes, where each volume has only two neighbors on opposite faces and no simplex can be formed.

In such cases, the displacement field within the subface e_{ijk} is a pair-wise polynomial approximation between the adjacent volumes, $\mathbf{x}_i$ and $\mathbf{x}_j$, regardless the dimension

$$\boldsymbol{u}_{ijk}(\mathbf{x}_g) = \underbrace{\left(1 - \mathcal{P}_c\left(z_g\right)\right)}_{\varphi_i} \, \boldsymbol{u}_i + \underbrace{\mathcal{P}_c\left(z_g\right)}_{\varphi_j} \, \boldsymbol{u}_j, \tag{45}$$

where φ_i and φ_j are the shape functions, and z_g is the normalized projection of the integration point $\mathbf{x}_g$ into the vector which goes from $\mathbf{x}_i$ to $\mathbf{x}_j$, denoted $\mathbf{x}_{\vec{ij}} = \left(\mathbf{x}_j - \mathbf{x}_i\right)$,

$$z_g = \frac{\left(\mathbf{x}_g - \mathbf{x}_i\right)^T \mathbf{x}_{\vec{ij}}}{||\mathbf{x}_{\vec{ij}}||^2}. \tag{46}$$

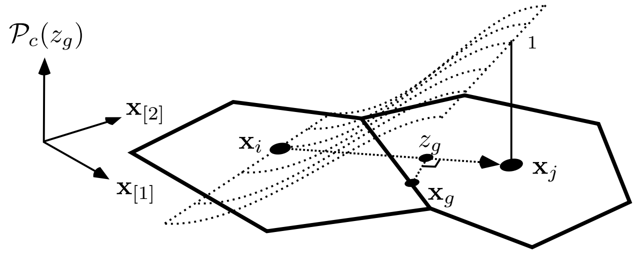

Figure 9: The gradient of the pairwise approximation is not constant along the face e_{ij}, since its normal is not necessary aligned with $\mathbf{x}_{\vec{ij}}$. The integration point is projected into $\mathbf{x}_{\vec{ij}}$ to evaluate the deformation matrix.

When calculating the quadrature of equation (26), the pairwise strain is given by

$$(\mathbf{S}\boldsymbol{u})_{ijk}\,|_{\mathbf{x}_g} = \mathbf{S}\varphi_i(\boldsymbol{\xi}_g)\,\boldsymbol{u}_i + \mathbf{S}\varphi_j(\boldsymbol{\xi}_g)\,\boldsymbol{u}_j, \tag{47}$$

$$= \begin{bmatrix} \dfrac{\partial\varphi_i}{\partial\mathbf{x}_{[1]}} & & \dfrac{\partial\varphi_j}{\partial\mathbf{x}_{[1]}} & \\ & \dfrac{\partial\varphi_i}{\partial\mathbf{x}_{[2]}} & & \dfrac{\partial\varphi_j}{\partial\mathbf{x}_{[2]}} \\ \dfrac{\partial\varphi_i}{\partial\mathbf{x}_{[2]}} & \dfrac{\partial\varphi_i}{\partial\mathbf{x}_{[1]}} & \dfrac{\partial\varphi_j}{\partial\mathbf{x}_{[2]}} & \dfrac{\partial\varphi_j}{\partial\mathbf{x}_{[1]}} \end{bmatrix}_{|\mathbf{x}_g} \begin{bmatrix} \boldsymbol{u}_{i[1]} \\ \boldsymbol{u}_{i[2]} \\ \boldsymbol{u}_{j[1]} \\ \boldsymbol{u}_{j[2]} \end{bmatrix} \tag{48}$$

$$= \mathbf{B}_{ijk}|_{\mathbf{x}_g} \quad \vec{\boldsymbol{u}}_{ijk}, \tag{49}$$

In the general case, the gradient is not constant along the face e_{ij}, since its normal is not necessary aligned with $\mathbf{x}_{\vec{ij}}$, as illustrated in Figure 9.

This pairwise approximation must be used only when necessary because it can not capture the deformation orthogonal to $\mathbf{x}_{\vec{ij}}$.

3.6. Homeostatic spline

The homeostatic spline is a function of a single variable defined from $z = 0$ to $z = 1$, denoted $\mathcal{P}_c(z)$, and curved by the parameter c, which indicates the level of smoothness. This spline is the simplest polynomial with c derivatives equal to zero at the endpoints $z = 0$ and $z = 1$. The polynomial degree is given by $2c + 1$, and such a polynomial requires $N_g = c + 1$ integration points to calculate the exact integral in equation (26) using the Gauss-Legendre quadrature.

When designing this spline, we wanted to gain accuracy by building a piece-wise bell-shaped interpolation function around the calculation points, inspired on the infinitely smooth kernels used in other numerical techniques. Therefore, we force the derivatives of the polynomial to be zero over such points in order to homogenize the function. For that reason, we use the term *homeostatic spline* when referring to this spline.

To fulfill the smoothness requisites commented before, we solved a linear system for calculating the $2c + 2$ coefficients of the polynomial. The equations of this system were obtained by forcing the c derivatives to be zero at the endpoints. Once we solved the coefficients for the first twenty polynomials, from $c = 0$ to $c = 19$, we found out that

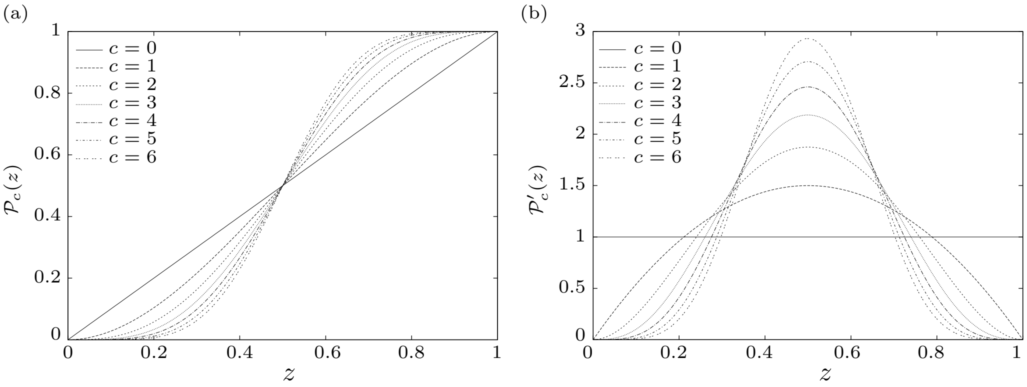

Figure 10: (a) The evolution of the homeostatic spline from $c = 0$ to $c = 6$ illustrates the smoothness requirements at the endpoints of each spline and its (b) first derivatives.

the first half of such coefficients are null, and the entire polynomial can be calculated directly as

$$\mathcal{P}_c(z) = \frac{1}{b_c} \sum_{k=1}^{1+c} (-1)^k \ b_k \ \ z^{(k+c)}, \tag{50}$$

where b_k is the k^{th} not null coefficient

$$b_k = \frac{1}{k+c} \left(\prod_{l=1}^{C_k} \frac{(1+c)}{l} - 1 \right), \tag{51}$$

C_k is the number of factors needed to calculate b_k

$$C_k = (c/2) - \big|1 + (c/2) - k\big|, \tag{52}$$

and b_c is accumulation of the coefficients for normalizing the spline

$$b_c = \sum_{k=1}^{1+c} (-1)^k b_k, \tag{53}$$

The first derivative is simply calculated as

$$\mathcal{P}'_c(z) = \frac{1}{b_c} \sum_{k=1}^{1+c} (-1)^k \ b_k \ \ (k+c) \ \ z^{(k+c-1)} \tag{54}$$

The Figure 10 shows (a) the evolution of the spline as we increase the smoothness parameter from $c = 0$ to $c = 6$, and (b) the evolution of it first derivative. Smoother splines produces higher order polynomials which increases the accuracy of the stress field approximation. This feature is specially important when solving non-linear problems sensibles to the stress field.

Since the derivatives of the homeostatic spline (50) are zero at the endpoints of the interval $[0, 1]$, the inverse function is not defined in that points. However, we estimate a

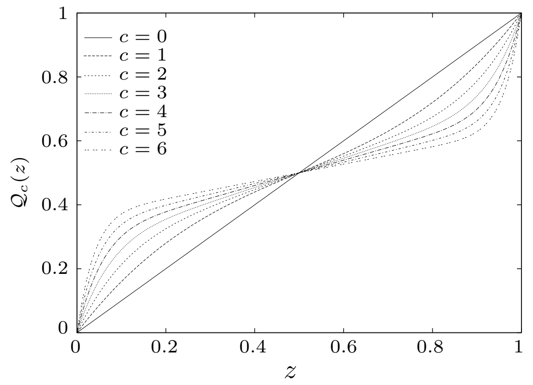

Figure 11: Curves of the pseudo-inverse $\mathcal{Q}_c$ for the first seven levels of smoothness. The slope at the midpoint exposes the null higher derivatives requirement when increasing the polynomial order.

pseudo-inverse within this interval, $\mathcal{Q}_c \approx \mathcal{P}_c^{-1}$, by finding the coefficients of a polynomial of the same degree, $2c+1$, such that the endpoints coincide with the spline and the first derivative at the midpoint is equivalent to the inverse of the spline first derivative, that is

$$\mathcal{Q}_c(0) = \mathcal{P}_c(0) = 0, \quad \mathcal{Q}_c(1) = \mathcal{P}_c(1) = 1, \quad \text{and} \quad \mathcal{Q}'_c(0.5) = \frac{1}{\mathcal{P}'_c(0.5)} \tag{55}$$

The higher derivatives in the midpoint are forced to be zero. Once we calculated the coefficients for the first twenty polynomials, from $c = 0$ to $c = 19$, we found out that the pseudo-inverse can be approximated directly from the following formulae

$$\mathcal{Q}(z) = \; a_1 \; z \quad + \quad (a_1 - 1)(2c+1) \; \sum_{k=1}^{2c} (-1)^k \; a_k \; z^{(k+1)} \tag{56}$$

where a_1 is the coefficient for z, and a_k is the factor that distinguish higher order coefficients. Such terms are calculated as

$$a_1 = \left(\frac{c}{2\sqrt{2}} + 1 \right)^2, \quad \text{and} \quad a_k = 2^{(k-1)} \; \prod_{l=1}^{k-1} \left(\frac{2c - l}{2 + l} \right), \tag{57}$$

respectively. The Figure 11 exhibits the curves for the first seven levels of smoothness. The null higher derivatives requirement is noticeable at the midpoint.

The Figure 12 shows the shape functions for the 2D case. The top displays the last node function and the bottom the first node function, the function of the second node is equivalent to that of the first one. The columns separate the first three levels of smoothness. Top and bottom functions coincides at the edges in order to create a continuous field, but only the bottom functions decay uniformly from the node to the opposite edge. The shape functions with $c = 0$ are the only case where all the shape functions are indistinguishable, these are planes.

The Figure 13 shows the magnitude of the gradient with respect to the normalized space. With the same tabular configuration of Figure 12, the columns separate the first

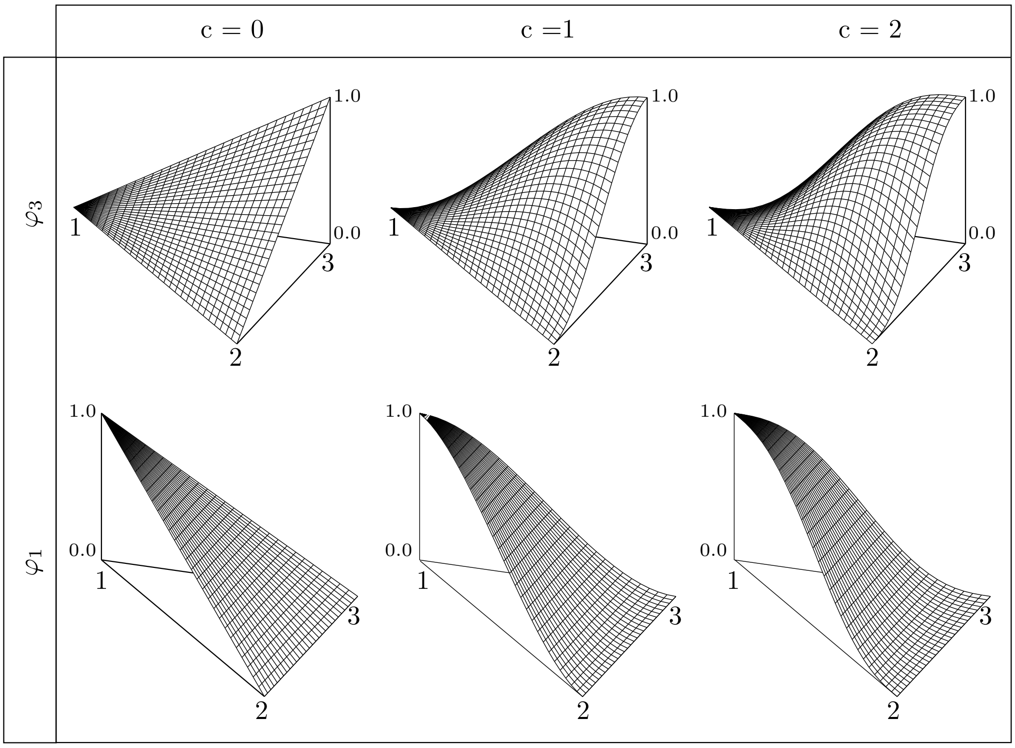

Figure 12: For the bidimensional case, the top displays the last node function and the bottom the first node function, the function of the second node is equivalent to that of the first one. The columns separate the first three levels of smoothness.

three levels of smoothness, the top displays the last node gradient and the bottom the first node gradient, the gradient of the second node is equivalent to that of the first one. Only the gradient magnitude at the bottom has a uniform variation from the node to the opposite face, and the value of the node does not contribute to the value of such a face. On the contrary, in the top can be observed that the value of the node contributes to the gradient at the opposite face, which means that using $c > 0$ the continuity on the stress field is only guaranteed at the calculation points, but not in the simplices edges.

3.7. Assembling volume's equation

By using the simplex-wise (42) or the pair-wise (49) approximation, the strain face integral (25) is reformulated as

$$\mathbf{H}_{ij} = \sum_{k=1}^{N_{ij}} \sum_{g=1}^{N_g} \; w_g \; \mathbf{B}_{ijk}|_{\mathbf{x}_g} \; \vec{\boldsymbol{u}}_{ijk}, \tag{58}$$

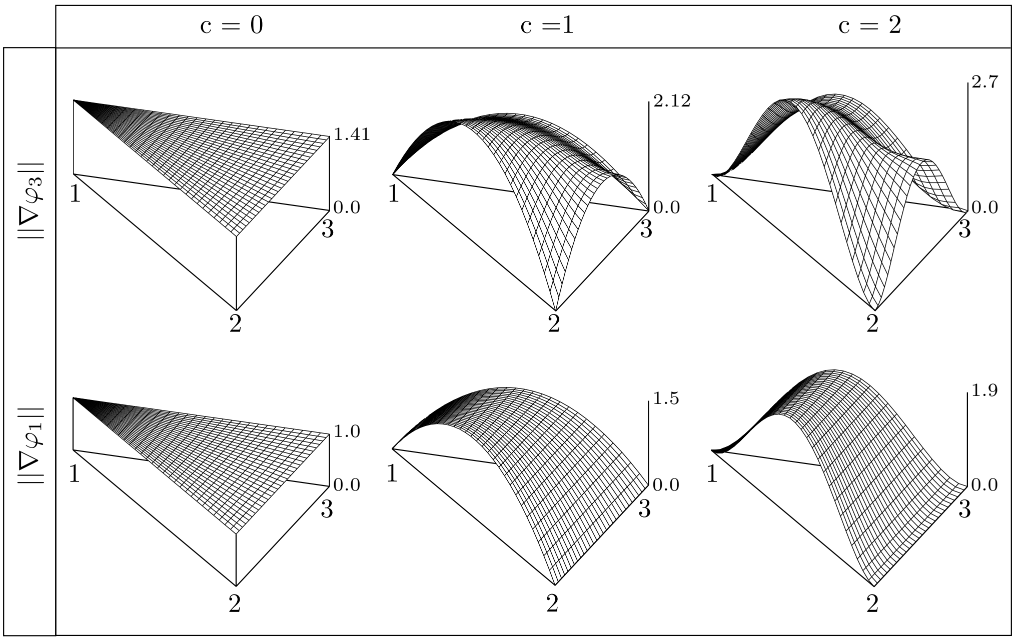

Figure 13: For the bidimensional case, the top displays the last node gradient magnitudes and the bottom the first node gradient magnitudes, the gradient magnitudes of the second node is equivalent to that of the first one. The columns separate the first three levels of smoothness.

then, the volume equilibrium equation (22) is

$$\sum_{j=1}^{N_i} \mathbf{T}_{ij}\mathbf{D}_{ij} \sum_{k=1}^{N_{ij}} \sum_{g=1}^{N_g} \; w_g \; \mathbf{B}_{ijk}|_{\mathbf{x}_g} \; \vec{\boldsymbol{u}}_{ijk} = 0, \tag{59}$$

reordering the terms we obtain

$$\sum_{j=1}^{N_i} \sum_{k=1}^{N_{ij}} \sum_{g=1}^{N_g} \; w_g \; \mathbf{K}_{ijk}|_{\mathbf{x}_g} \; \vec{\boldsymbol{u}}_{ijk} = 0, \tag{60}$$

where the matrix

$$\mathbf{K}_{ijk}|_{\mathbf{x}_g} = \mathbf{T}_{ij}\mathbf{D}_{ij}\mathbf{B}_{ijk}|_{\mathbf{x}_g}, \tag{61}$$

is the stiffness contribution at the integration point $\mathbf{x}_g$, within the subface e_{ijk} when integrating the i^{th} volume. Observe that the stiffness matrix $\mathbf{K}_{ijk}$ is rectangular and the resulting global stiffness matrix is asymmetric.

3.8. Boundary conditions

The Neumann boundary conditions are imposed over the volume faces e_{ij} intersecting the boundary, by replacing the corresponding term in the sum of equation (20) with the

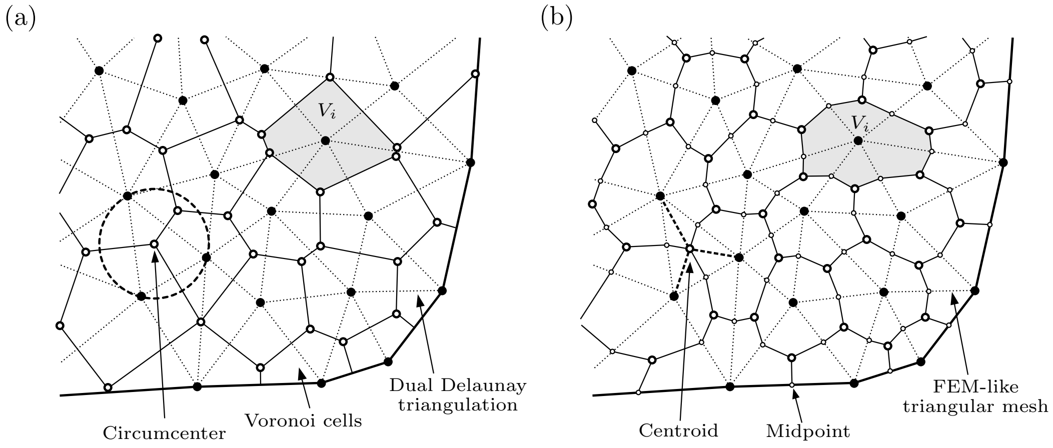

Figure 14: (a) The initial mesh is equivalent to the Voronoi diagram and the Voronoi centres correspond to the calculation points $\mathbf{x}_i$. (b) The initial mesh is generated from a FEM-like triangular mesh. The calculation points $\mathbf{x}_i$ are defined to be the nodes of the triangular mesh, and the volume faces are created by joining the centroids of the triangles with the midpoint of the segments.

integral of the function provided in (6a),

$$\int_{e_{ij}} \mathbf{TDS}\boldsymbol{u}\; dS = \int_{e_{ij}} \boldsymbol{b}_N(\mathbf{x})\; dS \tag{62}$$

The Dirichlet conditions are imposed over the volumes calculation points by fixing the displacement as it is evaluated in the function given in (6b),

$$\boldsymbol{u}_i = \boldsymbol{u}_D(\mathbf{x}_i), \tag{63}$$

Since the Dirichlet conditions are imposed directly on the calculation points, these points must be located along the face e_{ij} which intersects the boundary with the condition Γ_D.

3.9. Special cases

By making some considerations, we identify two special cases where the calculations can be simplified, in order to increase the performance of the total computation, at the expense of losing control over the volumes shape. These cases are 1) the Voronoi mesh assumption and 2) the FV-FEM correlation.

In the first case, we assume that the initial mesh is equivalent to the Voronoi diagram and that the Voronoi centres correspond to the calculation points $\mathbf{x}_i$. Hence, the subdivision of the neighborhood $\mathcal{B}_i$ is already given by the Delaunay triangulation which is dual to the Voronoi mesh, as illustrated in the Figure 14.a. Moreover, the integrals of subfaces e_{ijk} using pair-wise approximations can be exactly integrated with the midpoint rule, since the faces are orthogonal to the vector joining the calculation points $\mathbf{x}_{\vec{ij}}$, and the derivatives along the subface are constants.

In the second case, the initial mesh is generated from a FEM-like triangular mesh and the approximations are assumed to be linear. In such a case, the calculation points $\mathbf{x}_i$

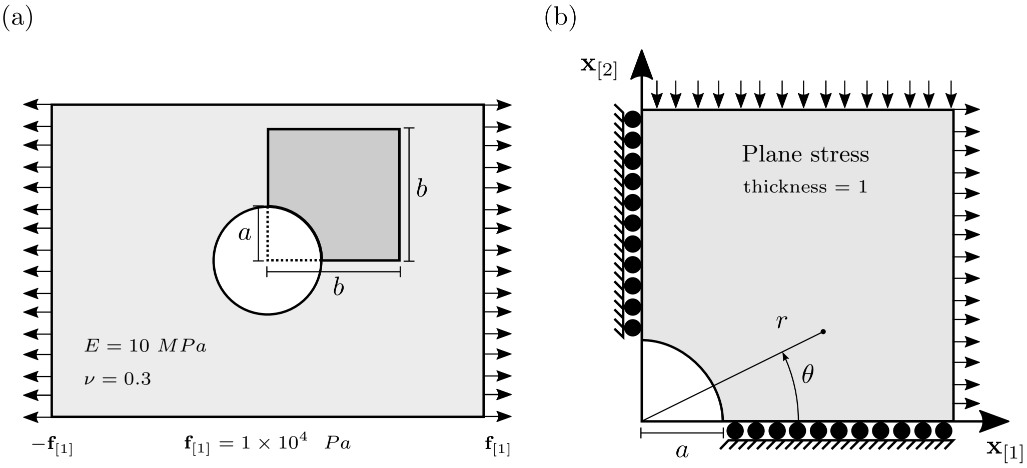

Figure 15: (a) Infinite plate with a hole being stretched along the horizontal axis with a force of $\mathbf{f}_{[1]} = 10\ \ kPa$ from each side. (b) Computational domain, $a = 0.5m$ and $b = 2m$, with axysymmetrical assumptions used to test the numerical method. The polar coordinates, r and θ, for calculating the analytical stress field.

are defined to be the nodes of the triangular mesh, and the volume faces are created by joining the centroids of the triangles with the midpoint of the segments, as presented in Figure 14.b. This particular version is equivalent to the cell-centred finite volume scheme introduced by Oñate et al [7], who proved that the global linear system produced by this FV scheme is identical to that produced by FEM if the same mesh is used.

4. Results

In order to test the numerical performance of the proposed method, we use the well known analytical experiment of an infinite plate with a circular hole in the origin (see [26]). In such a experiment, the plate is stretched along the horizontal axis with a uniform tension of $\mathbf{f}_{[1]} = 10\ \ kPa$ from each side, as is shown in Figure 15. The material is characterized by the Poisson ratio, $\nu = 0.3$, and Young modulus, $E = 10\ \ MPa$. Plane stress is assumed with thickness equivalent to the unity. The dimensions of the computational domain are $a = 0.5m$ and $b = 2m$. The analytical solution is given by the following formulae

$$\boldsymbol{\sigma}_{[11]} = \mathbf{f}_{[1]} \left[1 - \frac{a^2}{r^2} \left(\frac{3}{2} \cos(2\theta) + \cos(4\theta) \right) + \frac{3a^4}{2r^4} \cos(4\theta) \right], \tag{64}$$

$$\boldsymbol{\sigma}_{[22]} = \mathbf{f}_{[1]} \left[-\frac{a^2}{r^2} \left(\frac{1}{2} \cos(2\theta) - \cos(4\theta) \right) - \frac{3a^4}{2r^4} \cos(4\theta) \right], \tag{65}$$

$$\boldsymbol{\sigma}_{[12]} = \mathbf{f}_{[1]} \left[-\frac{a^2}{r^2} \left(\frac{1}{2} \sin(2\theta) + \sin(4\theta) \right) + \frac{3a^4}{2r^4} \sin(4\theta) \right], \tag{66}$$

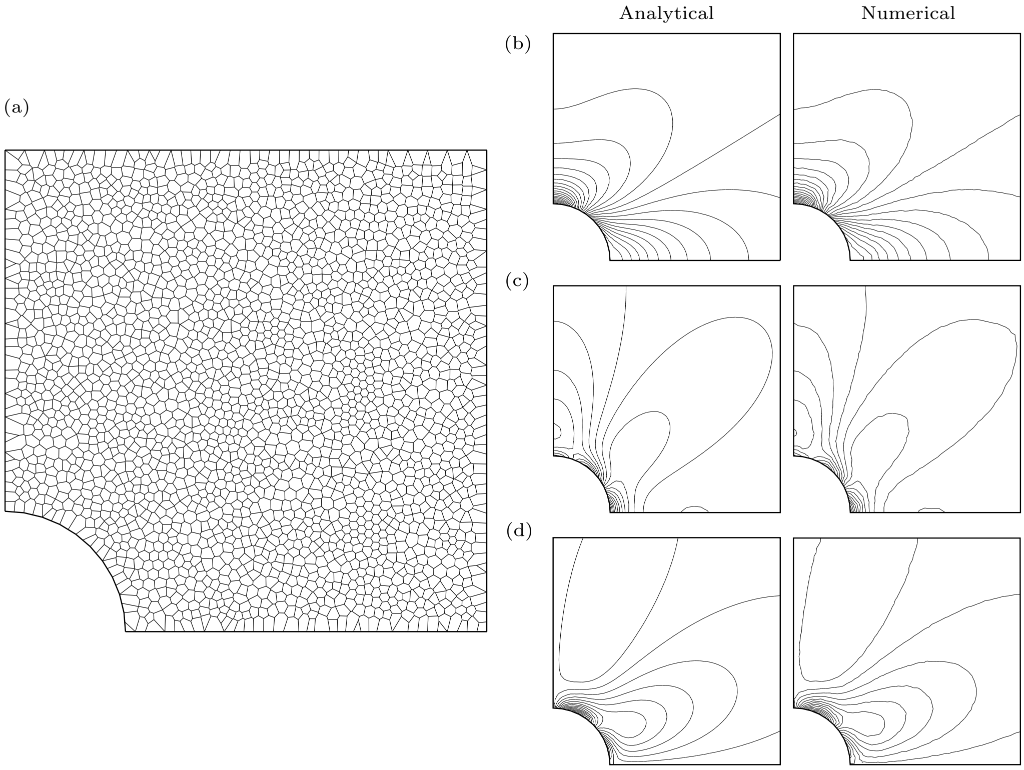

Figure 16: (a) Polygonal mesh used for comparison of numerical results. (b) Level sets of $\boldsymbol{\sigma}_{[11]}$ between 0 to 30 kPa. (c) Level sets of $\boldsymbol{\sigma}_{[22]}$ between -10 and 6 kPa. (d) Level sets of $\boldsymbol{\sigma}_{[12]}$ between -10 and 2 kPa.

where the polar coordinates,

$$r = \sqrt{\mathbf{x}_{[1]}^2 + \mathbf{x}_{[2]}^2}, \quad \text{and} \quad \theta = \tan^{-1}\left(\frac{\mathbf{x}_{[2]}}{\mathbf{x}_{[1]}}\right), \tag{67}$$

are used within the calculus. The Figure 16 exhibits (a) the discretization of the computational domain into 2411 polygonal volumes used to compare the numerical results against the analytical stress field. This mesh is not equivalent to the Voronoi diagram. (b) Level sets of $\boldsymbol{\sigma}_{[11]}$ between 0 to 30 kPa, with steps of 1 kPa. (c) Level sets of $\boldsymbol{\sigma}_{[22]}$ between -10 and 6 kPa, with steps of 0.8 kPa. (d) Level sets of $\boldsymbol{\sigma}_{[12]}$ between -10 and 2 kPa, with steps of 0.6 kPa.

The Dirichlet conditions are imposed on the bottom and left side of the computational domain as is shown in the Figure 16.b. Next in order, the analytic stress of equations (64), (65) and (66) is imposed as Neumann condition over the top and right side of the computational domain.

The Figure 17.a presents the averaged error as a function of the volumes face length mean, denoted Δx, as we might expect, the error is proportional to the mesh refinement. For a mesh of 478 volumes, the Figure 17.b shows the percentage of the error with

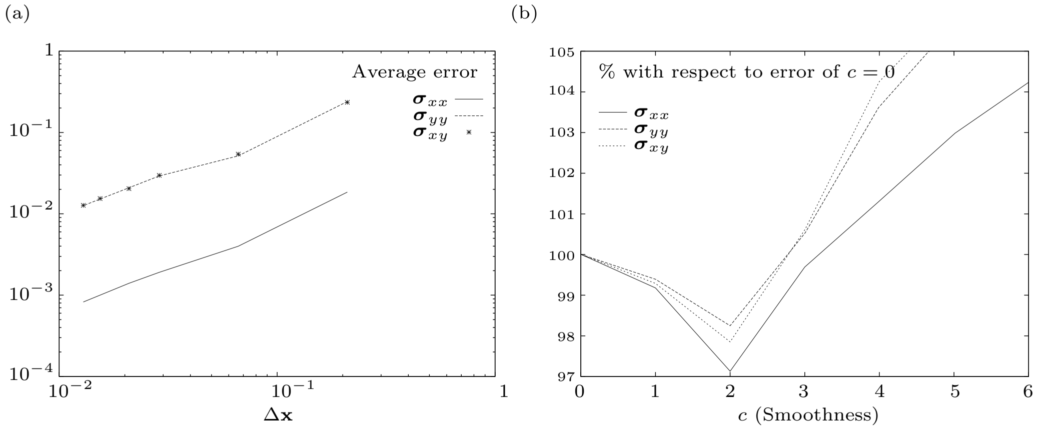

Figure 17: (a) The averaged error decreasing as a function of mesh size, denoted $\Delta\mathbf{x}$. (b) Using a mesh of 628 volumes, percentage of error for different smoothing levels with respect to the error of $c = 0$, which is the linear interpolator, error increases after $c = 2$.

respect to the error of $c = 0$, for different smoothing levels, $c = 0$ correspond the linear interpolator. Observe that the error of the stress field does not decreases significantly because we do not increase the degrees of freedom of the linear system, although we built a better field description, which can be useful when solving non-linear formulations.

5. Conclusions

In this work we discussed a numerical technique to solve the elasticity equation for unstructured and non conforming meshes formed by elements of any arbitrary polygonal/polyhedral shape. The elasticity-solver is based on a finite volume formulation that, using the divergence theorem, represent the volume integral of the stress divergence in terms of the surface integral of the stress over the volume boundary. For considering the unit vector normal to the boundary as a constant, the boundary is divided into flat faces. Conforming and non-conforming meshes are processed without distinction. The displacement field is a piece-wise polynomial approximation surrounding the volumes, built on the top of the simplices resulting from the Delaunay triangulation of the volume neighborhood. A pair-wise polynomial interpolation is used for neighborhoods where the simplices are not the best option, and for 1D problems.

We introduced the homeostatic splines and it pseudo-inverse for higher order polynomial interpolations without the necessity of increasing the discretization points.

Since the stress term is calculated directly on the boundary of the control volumes, this strategy can be used in fracture formulations where the volumes are treated as indivisible components and the rupture occurs across the volumes boundaries.

In future work, we will investigate the accuracy of this numerical procedure for solving non-linear and dynamic models when using higher order homeostatic splines.

Acknowledgement

The present investigation was sponsored by a CONACYT scolarship from the Mexican government and the TCAiNMaND project, an IRSES Marie Curie initiative under the European Union 7th Framework Programme. In addition, the authors want to express his gratitude to MSc. Ernesto Ortega, and to Drs. Rafel Herrera and Joaquin Peña for his insightful commments and discussions during the elaboration of this paper.

References

[1] A. Slone, K. Pericleous, C. Bailey, M. Cross, Dynamic fluid-structure interaction using finite volume unstructured mesh procedures, Computers and Structures 80 (2002) 371–390.

[2] A. Slone, K. Pericleous, C. Bailey, M. Cross, C. Bennett, A finite volume unstructured mesh approach to dynamic fluid-structure interaction: an assessment of the challenge of predicting the onset flutter, Applied Mathematical Modeling 28 (2004) 211–239. doi:10.1016/S0307-904X(03)00142-2.

[3] C. Bailey, G. A. Taylor, M. Cross, P. Chow, Discretisation procedures for multi-physics phenomena, Journal of Computational and Applied Mathematics 103 (1999) 3–17.

[4] M. Souli, A. Ouahsine, L. Lewin, Ale formulation for fluid-structure interaction problems, Computer Methods in Applied Mechanics and Engineering 190 (2000) 659–675.

[5] J. Lubliner, J. Oliver, S. Oller, E. Oñate, A plastic-damage model for concrete, International Journal of Solids and Structures 25 (1989) 299–326.

[6] F. Armero, S. Oller, A general framework for continuum damage models. i. infinitesimal plastic damage models in stress space, International Journal of Solids and Structures 37 (2000) 7409–7436.

[7] E. Oñate, M. Cervera, C. Zienkiewicz, A finite volume format for structural mechanics, International Journal for Numerical Methods in Engineering 37 (1994) 181–201.

[8] S. R. Idelsohn, E. Oñate, Finite volumes and finite elements: Two 'good friends', International Journal for Numerical Methods in Engineering 37 (1994) 3323–3341.

[9] Y. D. Fryer, C. Bailey, M. Cross, C. H. Lai, A control volume procedure for solving the elastic stress-strain equations on an unstructured mesh, Applied Mathematical Modeling 15 (1991) 639–645.

[10] C. Bailey, M. Cross, A finite volume procedure to solve elastic solid mechanics problems in three dimensions on an unstructured mesh, International Journal for Numerical Methods in Engineering 38 (1995) 1757–1776.

[11] A. K. Slone, C. Bailey, M. Cross, Dynamic solid mechanics using finite volume methods, Applied Mathematical Modeling (2003) 69–87.

[12] I. Demirdzic, D. Martinovic, Finite volume method for thermo-elasto-plastic stress analysis, Computer Methods in Applied Mechanics and Engineering 109 (1993) 331–349.

[13] I. Demirdzic, S. Muzaferija, Finite volume methods for stress analysis in complex domains, International Journal for Numerical Methods in Engineering 137 (1994) 3751–3766.

[14] I. Demirdzic, S. Muzaferija, M. Peric, Benchmark solutions of some structural analysis problems using finite-volume method and multigrid acceleration, International Journal for Numerical Methods in Engineering 40 (1997) 1893–1908.

[15] I. Bijelonja, I. Demirdzic, S. Muzaferija, A finite volume method for large strain analysis of incompressible hyperelastic materials, International Journal for Numerical Methods in Engineering 64 (2005) 1594–1609.

[16] I. Bijelonja, I. Demirdzic, S. Muzaferija, A finite volume method for incompressible linear elasticity, Computer Methods in Applied Mechanics and Engineering 195 (2006) 6378–6390.

[17] H. Jasak, H. G. Weller, Application of the finite volume method and unstructured meshes to linear elasticity, International Journal for Numerical Methods in Engineering 48 (2000) 267–287.

[18] Z. Tukovic, A. Ivankovic, A. Karac, Finite-volume stress analysis in multi-material linear elastic body, International Journal for Numerical Methods in Engineeringdoi:10.1002/nme.4390.

[19] T. Tang, O. Hededal, P. Cardiff, On finite volume method implementation of poro-elasto-plasticity soil model, International Journal for Numerical Methods in Engineeringdoi:10.1002/nag.2361.

[20] M. J. Borden, C. V. Verhoosel, M. Scott, T. J. R. Hughes, C. M. Landis, A phase-field description of dynamic brittle fracture, Computer Methods in Applied Mechanics and Engineering (2012) 217–220doi:10.1016/j.cma.2012.01.008.

[21] P. Cardiff, Z. Tukovic, H. Jasak, A. Ivankovic, A block-coupled finite volume methodology for linear elasticity and unstructured meshes, Computers and Structures (2016) 100–122doi:10.1016/j.compstruc.2016.07.004.

[22] M. A. A. Cavalcante, M.-J. Pindera, Generalized finite-volume theory for elastic stress analysis in solid mechanics - part i: Framework, Journal of Applied Mechanicsdoi:10.1115/1.4006805.

[23] J. M. Nordbotten, Cell-centered finite volume discretizations for deformable porous media, International Journal for Numerical Methods in Engineeringdoi:10.1002/nme.4734.

[24] B. Li, Z. Chen, G. Huan, Control volume function approximation methods and their applications to modeling porous media flow, Advances in water resources (2003) 435–444.

[25] B. Li, Z. Chen, G. Huan, Control volume function approximation methods and their applications to modeling porous media flow ii: the black oil model, Advances in water resources (2004) 99–120.

[26] S. P. Timoshenko, J. N. Goodier, Theory of Elasticity, 3rd Edition, McGraw-Hill, 1970.