Virtual Element Method for Geomechanical Simulations of Reservoir Models

Odd Andersen · Halvor M. Nilsen · Xavier Raynaud

Received: date / Accepted: date

Abstract In this paper we study the use of Virtual Element Method (VEM) for geomechanics. Our emphasis is on applications to reservoir simulations. The physical processes behind the formation of the reservoirs, such as sedimentation, erosion and faulting, lead to complex geometrical structures. A minimal representation, with respect to the physical parameters of the system, then naturally leads to general polyhedral grids. Numerical methods which can directly handle this representation will be highly favorable, in particular in the setting of advanced work-flows. The virtual element method is a promising candidate to solve the linear elasticity equations on such models. In this paper, we investigate some of the limits of the VEM method when used on reservoir models. First, we demonstrate that care must be taken to make the method robust for highly elongated cells, which is common in these applications, and show the importance of calculating forces in terms of traction on the boundary of the elements for elongated distorted cells. Second, we study the effect of triangulations on the surfaces of curved faces, which also naturally occur in subsurface models. We also demonstrate how a more stable stabilization term for reservoir application can be derived.

SINTEF ICT, Applied Mathematics, Oslo, Norway,
odd.andersen@sintef.no, halvorMoll.nilsen@sintef.no, xavier.raynaud@sintef.no

1 Introduction

Sedimentary formations are the result of long and complex geological processes. Sedimentation creates thin layers, faulting creates nontrivial connections between the layers and erosion creates degenerate layers. The formation retains an overall stratigraphic structure, in the sense that very different spatial correlations in the material properties can be observed between the horizontal and vertical directions, and long and thin cells are specific to reservoir simulations, see the section represent in Figure 1. The geometric modeling of sedimentary formations requires the parameterization of a very large number of complicated interfaces. Each interface then separates regions with material properties that may differ of several order of magnitude and must be captured with maximum accuracy. Because of these difficulties, computational considerations are often not prioritized in the design of geological grids, which will typically contain highly irregular cell shapes. The grid and material properties are strongly related, which cause severe limitations on remeshing. The industry standard for reservoir grids is the corner-point format [15]. In a corner-point grid, pillars which have a dominant vertical direction are first defined from a two-dimensional Cartesian partition. Then, for each set of four adjacent pillars, hexahedron cells are constructed by choosing 2 points on each pillars and connecting all these points (see the detailed in Section 6.2). Many geometrical grid formats have been proposed to improve on this format, for example Skua Grid [10], S-Grid, Faulted S-Grid and Cut-Cell [12]. By refining the mesh, it is of course always possible to improve the quality of the mesh from the point of view of numerical computation, but all compact representation of the underlying geology, that is a representation where the data (the material properties) is represented by the minimum number of cells, will lead to cells with high aspect ratio, distorted cells, faces or cells of very different sizes, cells or faces with different shapes. Methods which are robust for such grid will greatly simplify the modeling of subsurface physics.

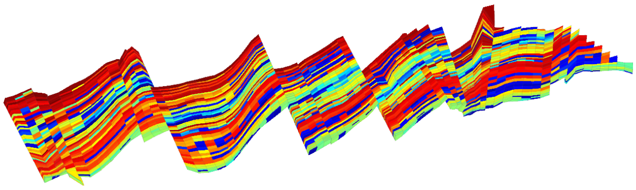


Fig. 1 Section of the Gullfaks reservoir model (Norway). Each color represents a different material property. We observe how the large aspect ratio in the cells.

In recent years, the coupling of geomechanical effects with subsurface flow has become more and more important in many areas including: oil and gas production from mature fields, oil and gas production from fractured tight reservoirs, fractured rock for geothermal application and risk assessment of CO_2 injection. Realistic modeling of these applications is hampered by the differences in the way geomechanics and flow models are build and discretized. Traditionally, the mechanic problems are solved using finite element methods but they are difficult to adapt to the standard geometrical representation of reservoir models, such as corner-point grids. In contrast, the Virtual Element Method (VEM) can operate on general polyhedral grids. As such, the ability of the method to handle irregular grids makes it very attractive for geomechanical applications. In this paper, we investigate if the method can effectively be applied on realistic reservoir grids. Our main result concerns the treatment of the load term. We observe that standard stabilization terms presented in the literature are not adapted to elongated cells with large aspect ratios, which are standard in geological models. We propose a modification of the stabilization constant which can be used in the 2D case and a discrete gradient approach to compute the load term which turns out to be little sensitive to the choice of stabilization and can be easily extended to 3D. In a first part of this paper, we present the VEM method following mostly [9] but we also try to clarify the connection with the construction of the projection operators, as introduced in the basic principles of [5]. In our numerical experiments, we will focus on the performance of the VEM method on geological grids. The emphasis will be on corner-point grids and complex small scale sedimentary models. To be able to demonstrate the VEM method in the reservoir setting we used the MRST framework [11] to simplify the grid handling. The numerical implementation of the VEM method used in this paper can be downloaded from [13], in particular the test case concerning the Norne reservoir model (see Section 6.2) is readily available from there.

2 The equations of linear elasticity

We consider the equation of linear elasticity for small deformations. The displacement is given by $\boldsymbol{u}(\boldsymbol{x}) \in \mathbb{R}^d$ for $d = 2, 3$ (2D or 3D case) and $\boldsymbol{x} \in \Omega \subset \mathbb{R}^d$. The equations are given by

$$\nabla \cdot \boldsymbol{\sigma} = \boldsymbol{f} \tag{1}$$

with

$$\boldsymbol{\sigma} = \boldsymbol{C}\boldsymbol{\varepsilon} \quad \text{and} \quad \boldsymbol{\varepsilon} = \frac{1}{2}(\nabla + \nabla^T)\boldsymbol{u} \tag{2}$$

Here, $\boldsymbol{\sigma}$, $\boldsymbol{\varepsilon}$ and $\boldsymbol{u}$ denote the Cauchy stress tensor, the infinitesimal strain tensors and the displacement field, respectively. The vector function $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^d$ is an external volumetric force that we will refer to as the *load term.* The linear operator $\boldsymbol{C}$ is a fourth order stiffness tensor which satisfies, for some constant $c > 0$, the ellipticity condition

$$c\boldsymbol{S} : \boldsymbol{S} \leq \boldsymbol{S} : \boldsymbol{C}\boldsymbol{S}, \tag{3}$$

for any symmetric matrix $\boldsymbol{S} \in \mathbb{R}^{d\times d}$. The symbol : denotes the scalar product in $\mathbb{R}^{d\times d}$ defined as

$$\boldsymbol{A} : \boldsymbol{B} = \mathrm{tr}(\boldsymbol{A}^T\boldsymbol{B}), \tag{4}$$

for any $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{d\times d}$.

3 Presentation of the VEM method for linear elasticity

The VEM method was first introduced in the framework of mimetic discretization methods but later rephrased in the language of finite element methods (see [4] for discussions). A general presentation of VEM is given in [5]. The same authors present convergence results for linear elasticity in [3]. Our implementation of the VEM follows the presentation done in [9]. A thorough account with all the practical details in the case of the Laplace operator can be found in [6]. We rewrite the equation of linear elasticity equation in the weak form,

$$\int_{\Omega} \boldsymbol{\varepsilon}(\boldsymbol{v}) : \boldsymbol{C}\boldsymbol{\varepsilon}(\boldsymbol{u}) \, \mathrm{d}\boldsymbol{x} = \int_{\Omega} \boldsymbol{f} \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x}, \tag{5}$$

which must hold for all displacement field $\boldsymbol{v} : \mathbb{R}^d \to \mathbb{R}^d$. Let N_c denote the number of cells and $\{E_i\}_{i=1}^{N_c}$ the grid cells. We define the bilinear form a_{E_i} as

$$a_{E_i}(\boldsymbol{u}, \boldsymbol{v}) = \int_{E_i} \boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{C}\boldsymbol{\varepsilon}(\boldsymbol{v}) \, \mathrm{d}\boldsymbol{x}$$

and decompose the global bilinear energy form $a(\cdot,\cdot)$ in cell contributions,

$$a(\boldsymbol{u}, \boldsymbol{v}) := \int_{\Omega} \boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{C}\boldsymbol{\varepsilon}(\boldsymbol{v}) \, \mathrm{d}\boldsymbol{x} = \sum_{i=1}^{N_c} a_{E_i}(\boldsymbol{u}, \boldsymbol{v}). \tag{6}$$

In the rest of this section, we consider a given cell E and will denote by $\mathcal{V}_E$ the finite dimensional approximating function space in E. In the VEM approach, the basis functions of $\mathcal{V}_E$ are not known explicitly but, for a first-order VEM method, the requirements on $\mathcal{V}_E$ are that it contains the space of polynomials of order 1, denoted $\mathbb{P}_1(E)$, and that the bilinear form $a_E(\boldsymbol{u}, \boldsymbol{v})$ can be computed exactly for any $\boldsymbol{u} \in \mathbb{P}_1(E)$ and any $\boldsymbol{v} \in \mathcal{V}_E$, using only the degrees of freedom of $\boldsymbol{v}$. As in the standard finite element method, the degrees of freedom are the nodal displacements, so that the continuity at the boundaries of each element is ensured by requiring linearity on the edges and a local reconstruction on the faces, which depends only on the values at the edges of the face where the reconstruction is done. The system matrix can be assembled element-wise. Let us denote

$$\mathcal{V}_E^{\text{scalar}} = \{\boldsymbol{v} \in H^1(E) \mid \boldsymbol{v}_{|e} \in \mathbb{P}_1(e) \text{ for all edges } e\}, \tag{7}$$

for $i = 1, \ldots, d$. For a given node η of E, we can construct a function ϕ_η in $\mathcal{V}_E^{\text{scalar}}$ such that $\phi_\eta(\bar{\eta}) = 1$ if $\bar{\eta} = \eta$ and zero if $\bar{\eta} \neq \eta$. The virtual basis functions of $\mathcal{V}_E$ are then given by

$$\boldsymbol{\phi}_\eta^k(\boldsymbol{x}) = \phi_\eta(\boldsymbol{x}) \boldsymbol{e}_k \tag{8}$$

for $\eta \in N(E)$ and $k = 1, \ldots, d$, where $N(E)$ denotes the set of nodes of the cell E and e_k is the unit vector in the direction given by the index k. After having introduced the projection operator, we will add some extra requirements for ϕ_η concerning its first and second order moment. But beside that, no more explicit properties for ϕ_η are needed and this is one of the important point of the method, which also makes it so flexible. The projection operator, which we denote $\boldsymbol{\pi}^\nabla$, is defined with respect to the energy norm a_E. We consider first order approximations and, for any displacement field $\boldsymbol{u}$ in the Hilbert space $[H^1(E)]^3$, the projection $\boldsymbol{\pi}^\nabla(\boldsymbol{u})$ of $\boldsymbol{u}$ is defined as the element $\boldsymbol{p} \in [\mathbb{P}_1(E)]^3$ such that

$$a_E(\boldsymbol{p}, \boldsymbol{q}) = a_E(\boldsymbol{u}, \boldsymbol{q}),$$

for all $\boldsymbol{q} \in [\mathbb{P}_1(E)]^3$. Since the bilinear form a_E is degenerate, additional conditions must be imposed to define completely $\boldsymbol{\pi}^\nabla$, see (22) in Section 3.2 for the rigorous definition. For any displacement field $\boldsymbol{u}$, the energy $a_E(\boldsymbol{u}, \boldsymbol{u})$ can be decomposed using Pythagoras' identity,

$$a_E(\boldsymbol{u}, \boldsymbol{u}) = a_E(\boldsymbol{\pi}^\nabla \boldsymbol{u}, \boldsymbol{\pi}^\nabla \boldsymbol{u}) + a_E((\boldsymbol{I} - \boldsymbol{\pi}^\nabla)\boldsymbol{u}, (\boldsymbol{I} - \boldsymbol{\pi}^\nabla)\boldsymbol{u}). \tag{9}$$

The first term on the right-hand side can be computed exactly from the degree of freedom, for any $\boldsymbol{u} \in \mathcal{V}_E$. The last term can not be handled generically and is therefore replace by a *stabilization* term which takes the form of a bilinear form s_E, whose role is to ensure that the ellipticity of a_E is retained. Hence, the energy is finally approximated by

$$a_{h,E}(\boldsymbol{u}, \boldsymbol{u}) = a_E(\boldsymbol{\pi}^\nabla \boldsymbol{u}, \boldsymbol{\pi}^\nabla \boldsymbol{u}) + s_E((\boldsymbol{I} - \boldsymbol{\pi}^\nabla)\boldsymbol{u}, (\boldsymbol{I} - \boldsymbol{\pi}^\nabla)\boldsymbol{u}). \tag{10}$$

In the general framework of VEM, as introduced in [5], the computation of the projection operator typically requires the computation of an inverse, locally for each cell. The formulation in [9] has the advantage of giving an explicit expression of the projection operator. In the presentation that follows, we will try to clarify the connection between the two approaches.

3.1 The kinematics of affine displacement

The physics of linear elasticity is associated with linear deformations, in particular the rigid body motions play a crucial role. Let us recall some simple facts on the kinematics of affine displacements. The linear space of affine displacements, which we denote by $\mathcal{P}$, corresponds to the sum of the translations and linear transformation so that any $\boldsymbol{l} \in \mathcal{P}$ can be written as $\boldsymbol{l}(\boldsymbol{x}) = \boldsymbol{u} + \boldsymbol{L}\boldsymbol{x}$, for $\boldsymbol{u} \in \mathbb{R}^d$ and $\boldsymbol{L} \in \mathbb{R}^{d \times d}$. The dimension of the space of $\mathcal{P}$ is $d^2 + d$. The subspace of rigid body motion, which we denote $\mathcal{P}_r$, contains the rotation and the translation. Any $\boldsymbol{l} \in \mathcal{P}_r$ can be written as

$$\boldsymbol{l}(\boldsymbol{x}) = \boldsymbol{u} + \boldsymbol{\Omega}(\boldsymbol{x} - \boldsymbol{x}_0), \tag{11}$$

for any $\boldsymbol{u}$, $\boldsymbol{x}_0 \in \mathbb{R}^d$ and $\boldsymbol{\Omega}$ that belongs to the space of skew-symmetric matrices, denoted $\mathrm{aSym}(\mathbb{R}^d)$. There is a redundancy in the choice of $\boldsymbol{x}_0$ and $\boldsymbol{u}$ so that a unique decomposition of $\boldsymbol{l} \in \mathcal{P}_r$ is given by $\boldsymbol{l}(\boldsymbol{x}) = \boldsymbol{u} + \boldsymbol{\Omega}\boldsymbol{x}$ for $\boldsymbol{u} \in \mathbb{R}^d$ and $\boldsymbol{\Omega}$ skew-symmetric. Hence, the space $\mathcal{P}_r$ is isomorphic to the sum of the linear space of translation and the linear space of skew-symmetric matrices, and its dimension is therefore $d(d+1)/2$. The space of non rigid body motion is the quotient of $\mathcal{P}$ with respect to $\mathcal{P}_r$, which we denote $\mathcal{P}/\mathcal{P}_r$. We introduce the projection operator $\boldsymbol{\pi}_c$ in $\mathcal{P}$ defined as

$$\boldsymbol{\pi}_c(\boldsymbol{l}) = \frac{1}{2}(\boldsymbol{L} + \boldsymbol{L}^T)(\boldsymbol{x} - \bar{\boldsymbol{x}}_E), \tag{12}$$

for any $\boldsymbol{l}(\boldsymbol{x}) = \boldsymbol{u} + \boldsymbol{L}\boldsymbol{x} \in \mathcal{P}$. Here, $\bar{\boldsymbol{x}}_E$ denotes the arithmetic average of the positions $\boldsymbol{x}_i$ of all the nodes of the cell E, that is

$$\bar{\boldsymbol{x}}_E = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i,$$

where n corresponds to the number of nodes in E. We can check that $\boldsymbol{\pi}_c$ is a projection and $\boldsymbol{\pi}_c(\boldsymbol{l}) = 0$ if and only if $\boldsymbol{l} \in \mathcal{P}_r$. Hence, the image of $\boldsymbol{\pi}_c$ is in bijection with the space of linear strain $\mathcal{P}/\mathcal{P}_r$ which we therefore identify to $\mathcal{P}_c = \boldsymbol{\pi}_c(\mathcal{P})$. Note that $\mathcal{P}_c$ can also be defined as

$$\mathcal{P}_c = \{\boldsymbol{l} \in \mathcal{P} \mid \nabla \boldsymbol{l} = \nabla \boldsymbol{l}^T \text{ and } \boldsymbol{l}(\bar{\boldsymbol{x}}_E) = 0\}. \tag{13}$$

Then, we introduce the projection $\boldsymbol{\pi}_r$ from $\mathcal{P}$ to $\mathcal{P}_r$ as

$$\boldsymbol{\pi}_r(\boldsymbol{l}) = \boldsymbol{l} - \boldsymbol{\pi}_c(\boldsymbol{l}) \tag{14}$$

so that

$$\boldsymbol{\pi}_r(\boldsymbol{l}) = \boldsymbol{l}(\bar{\boldsymbol{x}}_E) + \frac{1}{2}(\boldsymbol{L} - \boldsymbol{L}^T)(\boldsymbol{x} - \bar{\boldsymbol{x}}_E).$$

We can check that $\mathcal{P}_r = \boldsymbol{\pi}_r(\mathcal{P})$, $\boldsymbol{\pi}_c\boldsymbol{\pi}_r = \boldsymbol{\pi}_r\boldsymbol{\pi}_c = 0$, $\boldsymbol{\pi}_c + \boldsymbol{\pi}_r = \boldsymbol{I}_{\mathcal{P}}$. The space $\mathcal{P}_c$ is isomorphic to the space of symmetric matrices, denoted Sym. We consider the case $d = 3$ and use Kelvin's notation to represent Sym so that, for any $\boldsymbol{a} \in \text{Sym}$, its Kelvin representation in $\mathbb{R}^6$, which we denote $\hat{\boldsymbol{a}} \in \mathbb{R}^6$, is given by

$$\hat{\boldsymbol{a}}^T = [a_{11}, a_{22}, a_{33}, \sqrt{2}a_{23}, \sqrt{2}a_{13}, \sqrt{2}a_{12}]. \tag{15}$$

The square root in the definition above has the advantage to lead to the following correspondence between the scalar products in Sym and $\mathbb{R}^6$,

$$\boldsymbol{a} : \boldsymbol{b} = \hat{\boldsymbol{a}} \cdot \hat{\boldsymbol{b}}, \tag{16}$$

for any $\boldsymbol{a}, \boldsymbol{b} \in \text{Sym}$. Note that the authors in [9] use Voigt instead of Kelvin notations, which explains why the expressions given in the present paper differ up to a coefficient to those in [9]. We define the symmetric tensor $\hat{\boldsymbol{C}} \in \mathbb{R}^{6\times 6}$ by the identity

$$\boldsymbol{a} : \boldsymbol{C}\boldsymbol{b} = \hat{\boldsymbol{a}}^T \hat{\boldsymbol{C}} \hat{\boldsymbol{b}}, \tag{17}$$

for all $\boldsymbol{a}, \boldsymbol{b} \in \text{Sym}$. Then, we obtain that, for any $\boldsymbol{l}, \boldsymbol{m} \in \mathcal{P}$, we have

$$a_E(\boldsymbol{l}, \boldsymbol{m}) = \int_E \boldsymbol{\varepsilon}(\boldsymbol{l}) : \boldsymbol{C}\boldsymbol{\varepsilon}(\boldsymbol{m})\,\mathrm{d}\boldsymbol{x} = |E|\,\widehat{\boldsymbol{\pi}_c(\boldsymbol{l})}^T \hat{\boldsymbol{C}} \widehat{\boldsymbol{\pi}_c(\boldsymbol{m})} \tag{18}$$

The projection $\boldsymbol{\pi}_c(\boldsymbol{l})$ belongs to $\mathcal{P}_c$ and can therefore be written as $\boldsymbol{\pi}_c(\boldsymbol{l}) = \boldsymbol{Q}(\boldsymbol{x} - \hat{\boldsymbol{x}}_E)$ for $\boldsymbol{Q} \in \text{Sym}$. In (18), we wrote $\widehat{\boldsymbol{\pi}_c(\boldsymbol{l})}$ for the Kelvin representation

$\hat{\boldsymbol{Q}}$ of $\boldsymbol{Q}$, and similarly for $\widehat{\boldsymbol{\pi}_c(\boldsymbol{m})}$. For the space $\mathcal{P}_r$, we use the mapping between aSym and $\mathbb{R}^3$ given by the cross-product operation. For any $\boldsymbol{a} \in$ aSym, we can define a rotation vector $\hat{\boldsymbol{a}} \in \mathbb{R}^3$ as

$$\hat{\boldsymbol{a}} = \sqrt{2}[-a_{2,1}, a_{1,2}, -a_{1,2}]^T. \tag{19}$$

Then, we have

$$\boldsymbol{a}\boldsymbol{x} = \frac{1}{\sqrt{2}}\hat{\boldsymbol{a}} \times \boldsymbol{x} \tag{20}$$

and

$$\boldsymbol{a} : \boldsymbol{b} = \hat{\boldsymbol{a}} \cdot \hat{\boldsymbol{b}} \tag{21}$$

for any $\boldsymbol{a}, \boldsymbol{b} \in$ aSym. The normalization using $\frac{1}{\sqrt{2}}$ in (20) is used in order to get an exact correspondence between the scalar products in aSym and $\mathbb{R}^3$, as in (16) in the case of Sym. The basis of $\mathcal{P}_r$ is given by the canonical basis of $\mathbb{R}^6$, the first three components corresponding to the translation vector while the three last components correspond to a rotation vector.

3.2 The projection operator

The projection operator on the space of affine displacement, with respect to the bilinear form a, plays an essential role in the formulation of a first order VEM method. We follow the notation of [5] and denote this projection by $\boldsymbol{\pi}^\nabla$, even if the bilinear form we consider is not the H^1 semi-norm, which is used as example in [5]. For a given displacement function $\boldsymbol{\nu} \in \mathcal{V}_E$, the projection $\boldsymbol{p} = \boldsymbol{\pi}^\nabla(\boldsymbol{\nu})$ is defined as the unique element $\boldsymbol{p} \in \mathcal{P}$ which satisfies

$$a_E(\boldsymbol{p}, \boldsymbol{q}) = a_E(\boldsymbol{\nu}, \boldsymbol{q}), \tag{22a}$$

for all $\boldsymbol{q} \in \mathcal{P}$ and such that

$$\boldsymbol{\pi}_r \boldsymbol{p} = \boldsymbol{\pi}_r \boldsymbol{\nu}, \tag{22b}$$

which means that the projection of $\boldsymbol{p}$ and $\boldsymbol{\nu}$ on $\mathcal{P}_r$ coincide. The condition (22b) is necessary to determine a unique solution. Indeed, the bilinear form a is degenerate as it is invariant with respect to the space of rigid body motion and the condition (22b) eliminates this underteminancy by imposing a rigid body motion for $\boldsymbol{p}$. It is important to note that the definition of the projection operator is in fact independent of $\mathcal{V}_E$ and could be extended to $[H^1(E)]^3$. However, for $\mathcal{V}_E$, the projection can be computed exactly, using directly the degrees of freedom and without further integration. Up to now, we have only introduced $\boldsymbol{\pi}_r$ and $\boldsymbol{\pi}_c$ on $\mathcal{P}$ and we will now extend these definitions to $\mathcal{V}_E$ so that the definition in (22b) makes actually sense. We let the reader check that the new definitions of $\boldsymbol{\pi}_c$ and $\boldsymbol{\pi}_r$ on $\mathcal{V}_E$, when restricted to $\mathcal{P}$, coincide with those introduced previously. We define the projection $\boldsymbol{\pi}_r : \mathcal{V}_E \to \mathcal{P}_r$ as

$$\boldsymbol{\pi}_r(\boldsymbol{\nu}) = \bar{\boldsymbol{\nu}}_E + \frac{1}{2}\left\langle \nabla\boldsymbol{\nu} - \nabla\boldsymbol{\nu}^T \right\rangle (\boldsymbol{x} - \bar{\boldsymbol{x}}_E), \tag{23}$$

where the bracket denote the cell average, i.e.,

$$\langle \boldsymbol{w} \rangle = \frac{1}{|E|} \int_E \boldsymbol{w} \, \mathrm{d}\boldsymbol{x} \quad \text{and} \quad \bar{\boldsymbol{\nu}} = \frac{1}{n} \sum_E \boldsymbol{\nu}_i.$$

We define the projection $\boldsymbol{\pi}_c : \mathcal{V}_E \to \mathcal{P}_c$ as

$$\boldsymbol{\pi}_c(\boldsymbol{\nu}) = \frac{1}{2} \left\langle \nabla \boldsymbol{\nu} + \nabla \boldsymbol{\nu}^T \right\rangle (\boldsymbol{x} - \bar{\boldsymbol{x}}_E).$$

In a moment, we are going to check that both projections can be computed directly from the degree of freedoms. First, we use these definitions to compute $\boldsymbol{\pi}^{\nabla}$. We start by considering a solution $\boldsymbol{p} = \boldsymbol{\pi}^{\nabla}(\boldsymbol{\nu})$ to (22) and show that (22a) yields

$$a_E(\boldsymbol{p}_c, \boldsymbol{q}_c) = a_E(\boldsymbol{\nu}, \boldsymbol{q}_c), \tag{24}$$

for any $\boldsymbol{q} \in \mathbb{P}_1(E)$, where $\boldsymbol{p}_c = \boldsymbol{\pi}_c(\boldsymbol{p})$ and $\boldsymbol{q}_c = \boldsymbol{\pi}_c(\boldsymbol{q})$. The symmetric gradient is zero for any element in $\mathcal{P}_r$, that is $\varepsilon \circ \boldsymbol{\pi}_r = 0$. Hence, $a_E(\boldsymbol{\pi}_r(\boldsymbol{p}), \boldsymbol{\nu}) = 0$ for any $\boldsymbol{p} \in \mathbb{P}_1(E)$ and $\boldsymbol{\nu}$. It implies that $a_E(\boldsymbol{p}, \boldsymbol{q}) = a_E(\boldsymbol{p}_c, \boldsymbol{q}_c)$ and $a_E(\boldsymbol{\nu}, \boldsymbol{q}) = a_E(\boldsymbol{\nu}, \boldsymbol{q}_c)$ so that Equation (22a) indeed implies (24). Let us now determine the element $\boldsymbol{p}$ that satisfies (22) for a given $\boldsymbol{\nu} \in \mathcal{V}_E$. The coercivity of the the form a_E on $\mathcal{P}_c$ follows from the definition of $\mathcal{P}_c$ and the coercivity of the tensor $\boldsymbol{C}$, see (3). Therefore, there exists a unique solution $\boldsymbol{p}_c \in \mathcal{P}_c$ such that (24) holds for all $\boldsymbol{q}_c \in \mathcal{P}_c$. For any $\boldsymbol{q}_c \in \mathcal{P}_c$, we have

$$a_E(\boldsymbol{p}_c, \boldsymbol{q}_c) = \int_E \nabla \boldsymbol{p}_c : \boldsymbol{C} \nabla \boldsymbol{q}_c \, \mathrm{d}\boldsymbol{x} = |E| \, \nabla \boldsymbol{p}_c : \boldsymbol{C} \nabla \boldsymbol{q}_c \tag{25}$$

and

$$a_E(\boldsymbol{\nu}, \boldsymbol{q}_c) = \int_E \frac{1}{2} (\nabla \boldsymbol{\nu} + \nabla \boldsymbol{\nu}^T) : \boldsymbol{C} \nabla \boldsymbol{q}_c \, \mathrm{d}\boldsymbol{x} = \left(\frac{1}{2} \int_E (\nabla \boldsymbol{\nu} + \nabla \boldsymbol{\nu}^T) \, \mathrm{d}\boldsymbol{x} \right) : \boldsymbol{C} \nabla \boldsymbol{q}_c. \tag{26}$$

Hence, $\nabla \boldsymbol{p}_c = \frac{1}{2} \int_E (\nabla \boldsymbol{\nu} + \nabla \boldsymbol{\nu}^T) \, \mathrm{d}\boldsymbol{x}$ which implies that $\boldsymbol{p}_c$ is uniquely defined as $\boldsymbol{p}_c = \boldsymbol{\pi}_c(\boldsymbol{\nu})$. We can conclude that $\boldsymbol{p}$ defined as

$$\boldsymbol{p} = \boldsymbol{p}_c + \boldsymbol{\pi}_r(\boldsymbol{\nu}) \tag{27}$$

is the unique solution to (22). Indeed, $\boldsymbol{\pi}_c(\boldsymbol{p}) = \boldsymbol{p}_c$ and $\boldsymbol{\pi}_r(\boldsymbol{p}) = \boldsymbol{\pi}_r(\boldsymbol{\nu})$ are both uniquely defined by (24) and (22b).

Let us now give more details on the assembly. To do so, we consider a basis function $\boldsymbol{\nu}^i \in \mathcal{V}_E$ for which the only non-zero displacement can only occur at the node i, that is $\boldsymbol{\nu}^i(\boldsymbol{x}_j) = 0$ if $i \neq j$. Such function can be written as

$$\boldsymbol{\nu}^i(\boldsymbol{x}) = \textstyle\sum_{j=1}^d \nu_j \phi_i(\boldsymbol{x}) \boldsymbol{e}_j,$$

where $\{\boldsymbol{e}\}_{j=1}^d$ is the basis for Cartesian coordinates. We have

$$\left\langle \nabla \boldsymbol{\nu}^i \right\rangle = \textstyle\sum_{j=1}^d \nu_j^i \boldsymbol{e}_j \left\langle \nabla \phi_i \right\rangle^T,$$

and we have to compute $\boldsymbol{q}^i = \langle \nabla \phi_i \rangle$. The expression above simplifies to

$$\langle \nabla \boldsymbol{\nu}^i \rangle = \boldsymbol{\nu}^i \boldsymbol{q}^{iT}.$$

For $\boldsymbol{q}^i$, using Stokes' theorem, we have

$$\boldsymbol{q}^i = \int_E \nabla \phi_i \, \mathrm{d}\boldsymbol{x} = \int_{\partial E} \phi_i \boldsymbol{n} \, \mathrm{d}\boldsymbol{x} = \sum_{f \in F(E)} \Big(\int_f \phi_i \, \mathrm{d}\boldsymbol{x} \Big) \, \boldsymbol{n}_f, \tag{28}$$

where $F(E)$ denotes the set of faces that belong to E. The integral in (28) can be computed exactly. For the 3D, we use a virtual space such that the first two moments of the virtual basis elements coincide with those of their projection, see [1]. The integral is zero if the node i does not belong to the face f and, otherwise,

$$\int_f \phi_i \, \mathrm{d}\boldsymbol{x} = \begin{cases} \frac{|f|}{m} + \frac{1}{2}(\boldsymbol{n}_{e,i^-} + \boldsymbol{n}_{e,i^+}) \cdot (\boldsymbol{x}^f - \bar{\boldsymbol{x}}^f) & \text{in 3D,} \\ \frac{|f|}{2} & \text{in 2D,} \end{cases} \tag{29}$$

where $\boldsymbol{x}^f$ is the centroid of the face f and $\bar{\boldsymbol{x}}^f$ the arithmetic average of the node coordinates, i.e. $\bar{\boldsymbol{x}}^f = \frac{1}{m}\sum_{j=1}^m \boldsymbol{x}_j^f$. We denote by $\boldsymbol{W}_c^i \in \mathbb{R}^{6\times 3}$ the matrix representation of $\boldsymbol{\pi}_c$ written in the basis of displacement for the node i (that is $\mathbb{R}^3$) and the basis of $\mathcal{P}_c$ (that is $\mathbb{R}^6$, using the Kelvin notation). For $l, m = \{1, 2, 3\}$, we have

$$\frac{1}{2} \left\langle \nabla \boldsymbol{\nu}^i + \nabla \boldsymbol{\nu}^{iT} \right\rangle_{l,m} = \frac{1}{2}(\nu_l^i q_m^i + \nu_m^i q_l^i)$$

so that

$$(\boldsymbol{W}_c^i)^T = \begin{pmatrix} q_1^i & 0 & 0 & 0 & \frac{1}{\sqrt{2}} q_3^i & \frac{1}{\sqrt{2}} q_2^i \\ 0 & q_2^i & 0 & \frac{1}{\sqrt{2}} q_3^i & 0 & \frac{1}{\sqrt{2}} q_1^i \\ 0 & 0 & q_3^i & \frac{1}{\sqrt{2}} q_2^i & \frac{1}{\sqrt{2}} q_1^i & 0 \end{pmatrix}.$$

We have

$$\frac{1}{2} \left\langle \nabla \boldsymbol{\nu}^i - \nabla \boldsymbol{\nu}^{iT} \right\rangle = \frac{1}{2}(\boldsymbol{\nu}^i \boldsymbol{q}^{iT} - \boldsymbol{q}^i \boldsymbol{\nu}^{iT}).$$

Using the general identity $(\boldsymbol{q}^i \times \boldsymbol{\nu}^i) \times \boldsymbol{x} = (\boldsymbol{q}^i \cdot \boldsymbol{x})\boldsymbol{\nu}^i - (\boldsymbol{\nu}^i \cdot \boldsymbol{x})\boldsymbol{q}^i$, we get that the $\mathbb{R}^3$ representation of the matrix in aSym above is given by $\frac{1}{\sqrt{2}} \boldsymbol{q}^i \times \boldsymbol{\nu}^i$. Hence, the matrix $\boldsymbol{W}_r^i \in \mathbb{R}^{6\times 3}$ that represents $\boldsymbol{\pi}_r$ written in the basis of displacement for the node i (that is $\mathbb{R}^3$) and the basis of $\mathcal{P}_r$ (that is $\mathbb{R}^6$ for the translation and the rotation vector) is given by

$$(\boldsymbol{W}_r^i)^T = \begin{pmatrix} \frac{1}{n} & 0 & 0 & 0 & \frac{-1}{\sqrt{2}} q_3^i & \frac{1}{\sqrt{2}} q_2^i \\ 0 & \frac{1}{n} & 0 & \frac{1}{\sqrt{2}} q_3^i & 0 & \frac{-1}{\sqrt{2}} q_1^i \\ 0 & 0 & \frac{1}{n} & \frac{-1}{\sqrt{2}} q_2^i & \frac{1}{\sqrt{2}} q_1^i & 0 \end{pmatrix}.$$

The matrices $\boldsymbol{W}_c$ from the space of all the degrees of freedom (that is $\mathbb{R}^{3n}$) to $\mathcal{P}_c$ is obtained by concatenating $\boldsymbol{W}_c^i$ and similarly for $\boldsymbol{W}_r$. To obtain, from $\boldsymbol{W}_c$ and $\boldsymbol{W}_r$, the matrix representations of $\boldsymbol{\pi}_c$ and $\boldsymbol{\pi}_r$ in terms only of the degrees of freedom, we have to find the decomposition of $\mathcal{P}_c$ and $\mathcal{P}_r$ in terms of the degrees of freedom. To do so, we introduce the vectors $\boldsymbol{r}^i$, for $i = \{1, \dots, n\}$ as

$$\boldsymbol{r}^i = \boldsymbol{x}_i - \bar{\boldsymbol{x}}_E.$$

We define $\boldsymbol{N}_c^i, \boldsymbol{N}_c^i \in \mathbb{R}^{3\times 6}$ as

$$\boldsymbol{N}_c^i = \begin{pmatrix} r_1^i & 0 & 0 & 0 & \frac{1}{\sqrt{2}}r_3^i & \frac{1}{\sqrt{2}}r_2^i \\ 0 & r_2^i & 0 & \frac{1}{\sqrt{2}}r_3^i & 0 & \frac{1}{\sqrt{2}}r_1^i \\ 0 & 0 & r_3^i & \frac{1}{\sqrt{2}}r_2^i & \frac{1}{\sqrt{2}}r_1^i & 0 \end{pmatrix} \text{ and } \boldsymbol{N}_r^i = \begin{pmatrix} 1 & 0 & 0 & 0 & \frac{-1}{\sqrt{2}}r_3^i & \frac{1}{\sqrt{2}}r_2^i \\ 0 & 1 & 0 & \frac{1}{\sqrt{2}}r_3^i & 0 & \frac{-1}{\sqrt{2}}r_1^i \\ 0 & 0 & 1 & \frac{-1}{\sqrt{2}}r_2^i & \frac{1}{\sqrt{2}}r_1^i & 0 \end{pmatrix}. \tag{30}$$

Then, the matrices $\boldsymbol{N}_c, \boldsymbol{N}_r \in \mathbb{R}^{3n\times 6}$ are obtained by concatenating $\boldsymbol{N}_c^i, \boldsymbol{N}_r^i$, respectively.

The projections can be then written in terms as a mapping from degrees of freedom to degrees of freedom,

$$\boldsymbol{P}_r = \boldsymbol{N}_r\boldsymbol{W}_r \quad \text{ and } \quad \boldsymbol{P}_c = \boldsymbol{N}_c\boldsymbol{W}_c \tag{31}$$

and the projection on affine displacement is given by $\boldsymbol{P} = \boldsymbol{P}_c + \boldsymbol{P}_r$. For any $\boldsymbol{\nu}, \boldsymbol{\eta} \in \mathcal{V}_E$, we have that

$$a_E(\boldsymbol{\pi}^\nabla\boldsymbol{\nu}, \boldsymbol{\pi}^\nabla\boldsymbol{\eta}) = |E|\,\widehat{\boldsymbol{\pi}_c(\boldsymbol{\nu})}^T\,\hat{\boldsymbol{C}}\widehat{\boldsymbol{\pi}_c(\boldsymbol{\eta})} = |E|\,\boldsymbol{\nu}^T\boldsymbol{W_c}^T\hat{\boldsymbol{C}}\boldsymbol{W}_c\boldsymbol{\eta},$$

where in the last term, slightly abusing the notations, we denote by $\boldsymbol{\nu}, \boldsymbol{\eta} \in \mathbb{R}^{3n}$ the vector composed of the degrees of freedoms of $\boldsymbol{\nu}, \boldsymbol{\eta} \in \mathcal{V}_E$. Using the same convention, we can write the bilinear form as

$$a_{h,E}(\boldsymbol{\nu}, \boldsymbol{\eta}) = \boldsymbol{\nu}^T\left(|E|\,\boldsymbol{W_c}^T\hat{\boldsymbol{C}}\boldsymbol{W}_c + (\boldsymbol{I} - \boldsymbol{P})^T\boldsymbol{S}(\boldsymbol{I} - \boldsymbol{P})\right)\boldsymbol{\eta}, \tag{32}$$

where $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ is a *stabilization* term. For the VEM method to be well-defined, the matrix $\boldsymbol{S}$ must be chosen such that it is positive, symmetric and definite on the kernel of $\boldsymbol{P}$. We note that the decomposition of the energy in two orthogonal parts, the linear part which ensures consistency and the higher order part which are handled so that stability is preserved, is analog to the decomposition used in [7], even if it was introduced there to add some freedom in the choice of the basis functions.

4 Implementation of the load term

The load term can be calculated in several different ways which are equivalent up to the order of accuracy of the methods. We have investigated the three following alternatives,

1. Computation using the projection operator $\boldsymbol{\pi}^\nabla$,
2. Integration using nodal quadrature,
3. Computation based on a discrete gradient operator.

Alternative 1 is the choice that naturally follows from the VEM approach and which is proposed in [5]. Alternative 2 was argued to be simpler and with similar accuracy in [9]. Alternative 3 is possible when the force is equal to the gradient of a potential. This last alternative actually came to the mind of the authors when they considered the poro-elasticity equation, where the divergence operator naturally arises. As we will see below, the discrete gradient is in fact derived from the discrete divergence operator by duality. The two first alternatives give similar results. We show that the last one has significantly less errors than the others for elongated grid cells.

4.1 Standard assembly of the load term (alternative 1 and 2)

For a given force $\boldsymbol{f}$, we consider the work done by the force for a given displacement field $\boldsymbol{u}$,

$$\int_{\Omega} \boldsymbol{f} \cdot \boldsymbol{u} \,\mathrm{d}\boldsymbol{x}. \tag{33}$$

This expression defines a linear form on the space of displacement. We denote by $\mathcal{V}$ the global discrete function space of displacement, which is constructed by taking the product of the $\mathcal{V}_E$ for all the cells E of the grid and requiring continuity at the cell boundaries and correspond to the nodal displacement in terms of the degrees of freedom. We want to find a discrete linear form on $\mathcal{V}$ that approximates (33). We can equip $\mathcal{V}$ with the standard scalar product in $\mathbb{R}^{n_N}$, that is $\sum_{\eta} \boldsymbol{u}_\eta \cdot \boldsymbol{\nu}_\eta$. Here, n_N denotes the total number of nodes. Any linear form on $\mathcal{V}$ can be represented by an element in $\mathcal{V}$, using this scalar product. Hence, we end up looking for an element $\widehat{\boldsymbol{f}} \in \mathcal{V}$ such that

$$\int_{\Omega} \boldsymbol{f} \cdot \boldsymbol{u} \,\mathrm{d}\boldsymbol{x} \approx \sum_{\eta} \widehat{\boldsymbol{f}}_\eta \cdot \boldsymbol{u}_\eta. \tag{34}$$

The vector $\widehat{\boldsymbol{f}} \in \mathbb{R}^{N_n}$ can be interpreted as a vector of *nodal forces.* We present several expressions for $\widehat{\boldsymbol{f}}$ corresponding to the three alternatives presented previously. First, we can use weights which are obtained using a first-order quadrature. For a node η, let us denote by $E(\eta)$ the set of cells to which the node η belongs. Using quadrature rules to integrate f on each cell, we obtain

$$\widehat{\boldsymbol{f}}_\eta = \left(\sum_{i \in E(\eta)} w_i^\eta \right) \boldsymbol{f}(\eta), \tag{35}$$

see [9] for the definitions of the weights w_i^η. This corresponds to alternative 2. For alternative 1, let $\boldsymbol{u}^\eta \in \mathcal{V}$ be a displacement for which the only non-zero degrees of freedom are those corresponding to the node η. Then, we have

$$\begin{aligned}\int_{\Omega} \boldsymbol{f} \cdot \boldsymbol{u}^{\eta} \, \mathrm{d}\boldsymbol{x} &= \sum_{i \in E(\eta)} \int_{E_i} \boldsymbol{f} \cdot \boldsymbol{u}^{\eta} \, \mathrm{d}\boldsymbol{x} \\ &\approx \sum_{i \in E(\eta)} \int_{E_i} \boldsymbol{\pi}_i^0(\boldsymbol{f}) \cdot \boldsymbol{u}^{\eta} \, \mathrm{d}\boldsymbol{x} \\ &= \sum_{i \in E(\eta)} \boldsymbol{\pi}_i^0(\boldsymbol{f}) \cdot \int_{E_i} \boldsymbol{u}^{\eta} \, \mathrm{d}\boldsymbol{x} \\ &= \sum_{i \in E(\eta)} \boldsymbol{\pi}_i^0(\boldsymbol{f}) \cdot \int_{E_i} \boldsymbol{\pi}_i^{\nabla}(\boldsymbol{u}^{\eta}) \, \mathrm{d}\boldsymbol{x}. \end{aligned} \tag{36}$$

Here, $\boldsymbol{\pi}_i^0$ denotes the L^2 projection to the space of constant functions (polynomials of degree zero) in the element E_i. To obtain the last integral, we use that fact that the virtual basis functions can be chosen such that the zero and first moment of a function $\boldsymbol{\nu}$ in $\mathcal{V}_E$ coincide with those of its projection $\boldsymbol{\pi}_i^{\nabla}\boldsymbol{\nu}$, that is

$$\int_E \boldsymbol{p} \cdot \boldsymbol{\nu} \, \mathrm{d}\boldsymbol{x} = \int_E \boldsymbol{p} \cdot \boldsymbol{\pi}_i^{\nabla}(\boldsymbol{\nu}) \, \mathrm{d}\boldsymbol{x},$$

for any $\boldsymbol{p} \in \mathbb{P}_1$ and $\boldsymbol{\nu} \in \mathcal{V}_E$. See [1] for more details. The choice of such basis implies that, for an element E, the modes that belong to $\ker \boldsymbol{\pi}^{\nabla}$, typically higher nonlinear modes, will not be excited directly by the force. From (36), we infer that $\widehat{\boldsymbol{f}}_{\eta}$ is defined as a linear combination of cell averages of $\boldsymbol{f}$,

$$\widehat{\boldsymbol{f}}_{\eta} = \sum_{i \in E(\eta)} m_i^{\eta} \boldsymbol{\pi}_i^0(\boldsymbol{f}) \tag{37}$$

where m_i^{η} are the weights given

$$m_i^{\eta} = \boldsymbol{e}_k \cdot \int_{E_i} \boldsymbol{\pi}_i^{\nabla}(\boldsymbol{\phi}_k^{\eta}) \, \mathrm{d}\boldsymbol{x}.$$

Note that the expression on the right above do not depend on k, as the same basis function is used in all directions. An other way to use the gradient projection to assemble the load term is given by the approximation

$$\int_{E_i} \boldsymbol{f} \cdot \boldsymbol{u}^{\eta} \, \mathrm{d}\boldsymbol{x} \approx \int_{E_i} \boldsymbol{f} \cdot \boldsymbol{\pi}_i^{\nabla}(\boldsymbol{u}^{\eta}) \, \mathrm{d}\boldsymbol{x} \tag{38}$$

on each cell E_i, and then a higher order approximation of $\boldsymbol{f}$ can be used. In the case of the a constant load term, as the one we will consider in our experiments, both choices (36) and (38) lead to the same discrete load term.

4.2 The discrete gradient approach

Let us now turn to alternative 3. We assume that the force can be written as the gradient of a potential, $\boldsymbol{f} = \nabla\psi$. We have, for a node η and a dimension $k \in \{1,2,3\}$,

$$\int_\Omega \boldsymbol{f}\cdot\boldsymbol{u}\,\mathrm{d}\boldsymbol{x} = \int_\Omega \nabla\psi\cdot\boldsymbol{u}\,\mathrm{d}\boldsymbol{x} = -\int_\Omega \psi\nabla\cdot\boldsymbol{u}\,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega}\psi\boldsymbol{u}\,\mathrm{d}\boldsymbol{x} = -\int_\Omega \psi\nabla\cdot\boldsymbol{u}\,\mathrm{d}\boldsymbol{x}. \tag{39}$$

The boundary integral vanishes because we assume Dirichlet boundary condition, $\boldsymbol{u} = 0$ for $\boldsymbol{x} \in \partial\Omega$. In the VEM space, there exists a natural discretization of the divergence operator as an operator from $\mathcal{V}$ to cell-wise constant functions, denoted $\mathcal{T}$, which is isomorphic to $\mathbb{R}^{N_c}$, where N_c denotes the number of cells. Indeed, for any discretized potential $\hat{\psi} \in \mathcal{T}$ and $\boldsymbol{\nu} \in \mathcal{V}$, we have

$$\int_\Omega \hat{\psi}\nabla\cdot\boldsymbol{\nu} = \sum_E \int_E \hat{\psi}_E\nabla\cdot\boldsymbol{\nu}\,\mathrm{d}\boldsymbol{x} = \sum_E \sum_{f\in F(E)} \hat{\psi}_E \int_f \boldsymbol{\nu}\cdot\boldsymbol{n}\,\mathrm{d}\boldsymbol{x}, \tag{40}$$

where $F(E)$ as before denotes the set of faces that belong to E. The last integral can be computed exactly as shown in (29). Then, using partial integration, we get

$$\int_\Omega \hat{\psi}\nabla\cdot\boldsymbol{\phi}^k_\eta\,\mathrm{d}\boldsymbol{x} = \sum_{j\in E(\eta)} \sum_{f_{j,l}\in E_j\cap E_l} \hat{\psi}_{E_j}(\boldsymbol{e}_k\cdot\boldsymbol{n}_{j.l}) \int_{f_{j,l}} \phi_\eta\,\mathrm{d}\boldsymbol{x}, \tag{41}$$

with the convention that we only get contribution in the integral when the face $f_{j,l}$ exists, that is when E_j and E_l share a common face. Note that by definition of the exterior normal, we have $\boldsymbol{n}_{j.l} = -\boldsymbol{n}_{j.l}$. We use (29) to compute the integral and therefore the divergence operator $\texttt{div} : \mathcal{V} \to \mathcal{T}$ is defined and can be computed exactly in the sense that

$$\texttt{div}(\boldsymbol{\nu}) = \boldsymbol{\pi}^0\nabla\cdot\boldsymbol{\nu}$$

for any $\boldsymbol{\nu} \in \mathcal{V}$, where $\boldsymbol{\pi}^0$ denotes the L^2 projection to $\mathcal{T}$. The transpose of the discrete divergence operator will give us a discrete approximation of the gradient. We can obtain an expression of the discrete gradient by reverting the order of the sum in (41). Let us denote by $F(\eta)$ the set of faces to which the node η belongs and, for a face f_k, we denote the neighboring cells of f_k by E^+_k and E^-_k. From (41), we can rewrite

$$\int_\Omega \hat{\psi}\nabla\cdot\boldsymbol{\phi}^k_\eta\,\mathrm{d}\boldsymbol{x} = -\sum_{f\in F(\eta)} (\hat{\psi}_{E^+_f} - \hat{\psi}_{E^-_f})(\boldsymbol{e}_k\cdot\boldsymbol{n}_f)\int_f \phi_\eta\,\mathrm{d}\boldsymbol{x}.$$

where the normal $\boldsymbol{n}_f$ of the face f is directed from E^-_f to E^+_f. This convention implies that $\boldsymbol{n}_{jl} = -\boldsymbol{n}_{lj} = \boldsymbol{n}_f$ if $E_j = E^-_f$ and $E_l = E^+_f$. Hence, the discrete gradient operator `grad` is the mapping from scalar cell values to vector node value given by

$$[\mathtt{grad}(\hat{\psi})]_{\eta,k} = \sum_{f\in F(\eta)} (\hat{\psi}_{E_f^+} - \hat{\psi}_{E_f^-})(\boldsymbol{e}_k \cdot \boldsymbol{n}_f) \int_f \phi_\eta \,\mathrm{d}\boldsymbol{x}. \tag{42}$$

Hence, gathering (39), (40) and (42), in this formulation, we obtain the following expression for $\widehat{\boldsymbol{f}}$, as the discrete gradient of the discretized potential, that is

$$\widehat{\boldsymbol{f}} = \mathtt{grad}(\hat{\psi}).$$

In (42), the expression only depends on differences of the potential, which can be estimated locally without knowledge of the global potential, i.e.

$$\hat{\psi}_{E_f^+} - \hat{\psi}_{E_f^-} = \hat{\boldsymbol{f}}_f \cdot \boldsymbol{dr}_f, \tag{43}$$

where $\hat{\boldsymbol{f}}_f$ is an approximation of the force on the face f and $\boldsymbol{dr}_f$ is the vector joining the centroids of E_f^- and E_f^+. In practice, it means that the method can be applied *even* if the force is not derived from a potential, as we can see that the potential ψ does not have to be computed. Note that, in the numerical tests that follow, we have not tested this case.

4.3 Interpretation of the discrete gradient approach using singular load term functions

When we consider a cell-valued potential ψ, the corresponding force $\boldsymbol{f} = \nabla\psi$ can be defined as a singular function with support on the cell faces. Let us define this class of function, which we will refer to as *2D-Dirac* functions. Given an internal 2D surface S in Ω (or 1D line in 2D), we define the *constant 2D-Dirac* function $\delta_S(x)$ as the distribution given by

$$< \delta_S, \phi >= \int_S \phi(x)\, dx,$$

for all $\phi \in C^\infty(\Omega)$. The 2D-Dirac δ_S is a measure which coincides with the Hausdorff measure on the $d-1$ dimensional set S. Then, we can also define 2D-Dirac function $h(x)\delta_S(x)$, for any $h \in L^1(S)$ as $< h\delta_S, \phi >= \int_S h(x)\phi(x)\, dx,$, for any $\phi \in C^\infty(\Omega)$. If the surface S is Lipschitz, then a continuous trace operator from $H^1(\Omega)$ to $H^{\frac{1}{2}}(S)$ can be defined, see for example [8]. Therefore, at least if $h \in L^2(S)$, we have that $h(x)\delta_S(x) \in H^{-1}(\Omega)$. Indeed, we have, for any $\phi \in H^1(\Omega)$,

$$\begin{aligned} < h\delta_S, \phi >= \int_S h(x)\phi(x)\, dx &\leq \|h\|_{L^2(S)} \|\phi\|_{L^2(S)} \\ &\leq C_1 \|h\|_{L^2(S)} \|\phi\|_{H^{\frac{1}{2}}(S)} \\ &\leq C_1 C_2 \|h\|_{L^2(S)} \|\phi\|_{H^1(\Omega)}, \end{aligned}$$

for two constants C_1 and C_2. From this observation, we can infer that the original system of equation (1) is *well-posed* for 2D Dirac vector functions $\boldsymbol{f}$.

Let us now consider a surface S that splits the domain Ω in two subdomains, namely Ω_- and Ω_+, and a potential ψ which is piecewise constant and takes the values $\psi_\pm$ in $\Omega_\pm$. The gradient of ψ in the sense of distribution is defined as

$$< \nabla\psi, \phi >= -\int \psi \nabla\phi \, dx \tag{44}$$

Let us consider ϕ with compact support in Ω so that, after using integration by part we obtain,

$$< \nabla\psi, \phi >= -(\psi_- \int_{\Omega_-} \nabla\phi \, dx + \psi_+ \int_{\Omega_+} \nabla\phi \, dx) == \int_S ((\psi_+ - \psi_-)\boldsymbol{n})\phi \, dx, \tag{45}$$

where $\boldsymbol{n}(x)$ denotes the normal to S at $x \in S$ pointing from Ω_- to Ω_+. From the definition (44) and (45), we get that the gradient of $\hat{\psi}$ is a 2D Dirac vector function given by

$$\nabla\psi = [\psi_+ - \psi_-]\boldsymbol{n}(x)\delta_S. \tag{46}$$

Let us now consider again a cell-wise constant potential function $\hat{\psi}$ defined on a mesh. Using the same notation as in the previous section, we infer from (46) that the gradient $\mathring{\boldsymbol{f}}$ of $\hat{\psi}$ in the sense of distribution is given by

$$\mathring{\boldsymbol{f}} = \nabla\hat{\psi} = \sum_{f \in F_{\text{int}}} (\hat{\psi}_{E_f^+} - \hat{\psi}_{E_f^-})\boldsymbol{n}_f \delta_f, \tag{47}$$

where F_{int} denotes the set of internal faces. Note that for the basis function $\boldsymbol{\phi}_\eta^k$ as defined in (8), we get

$$\int_\Omega \mathring{\boldsymbol{f}} \cdot \boldsymbol{\phi}_\eta^k = \sum_{f \in F_i} (\hat{\psi}_{E_f^+} - \hat{\psi}_{E_f^-})(\boldsymbol{e}_k \cdot \boldsymbol{n}_f) \int_f \phi_\eta(x) \, dx \tag{48}$$

and we recover expression (42). Hence, the discrete gradient approach can be interpreted in the following way. First, we approximate the volumetric load term $\boldsymbol{f}$ by a 2D Dirac function $\mathring{\boldsymbol{f}}$ with support on the cell faces and which is constant on each face, that is $\mathring{\boldsymbol{f}}$ has the form

$$\mathring{\boldsymbol{f}}(x) = \sum_{f \in F_i} \mathring{\boldsymbol{f}}_f \delta_f(x) \tag{49}$$

where $\mathring{\boldsymbol{f}}_f$ is a constant vector, for each face f. In the case the force $\boldsymbol{f}$ is derived from a potential, we can use the expression (47) to carry on this approximation. Otherwise, we propose to use expression (43) and consider

$$\mathring{\boldsymbol{f}}_f = (\hat{\boldsymbol{f}}_f \cdot \boldsymbol{dr}_f)\, \boldsymbol{n}_f.$$

Once $\mathring{\boldsymbol{f}}$ is computed, we use the VEM method to solve the problem defined as

$$\nabla \cdot \boldsymbol{\sigma} = \mathring{\boldsymbol{f}}.$$

Then, the assembly of the load term can be done *exactly*, as we can see from (48) in the case of a potential and otherwise

$$\int_{\Omega} \mathring{\boldsymbol{f}} \cdot \boldsymbol{\phi}_{\eta}^{k} = \sum_{f \in F_i} \mathring{\boldsymbol{f}}_f \cdot \boldsymbol{e}_k \int_f \phi_\eta(x)\, dx. \tag{50}$$

in the case where (49) is used. Note that the integrals in (48) and (49) can be computed exactly we use the virtual basis proposed in [1].

5 Stability with respect to aspect ratio

Let us now discuss the choice of the stabilization matrix $\boldsymbol{S}$ in (32). In [3], the authors propose

$$\boldsymbol{S} = \boldsymbol{I},$$

which is the simplest choice. In [9], the authors look at several cell shapes and recommend the stabilization term given by

$$\boldsymbol{S} = \alpha \boldsymbol{I} \tag{51}$$

where the constant α is chosen as

$$\alpha_G = \frac{|E|\,\mathrm{tr}(\hat{\boldsymbol{C}})}{\mathrm{tr}(\boldsymbol{N}_c^T \boldsymbol{N}_c)}, \tag{52}$$

as it gives an overall satisfactory approximation of the higher order nonlinear modes. This constant is stable with respect to isotropic scaling but it is not stable with respect to the aspect ratio.

5.1 Instability of α_G with respect to aspect ratio

To demonstrate that, we consider a rectangular element in 2D given by $[-h_1, h_1] \times [-h_2, h_2]$. In this case, an explicit definition of the virtual element space is available, as it is spanned by the four following functions

$$\varphi_1^l(\boldsymbol{x}) = 1, \quad \varphi_2^l(\boldsymbol{x}) = \frac{x_1}{h_1}, \quad \varphi_3^l(\boldsymbol{x}) = \frac{x_2}{h_2}, \quad \varphi(\boldsymbol{x}) = \frac{x_1 x_2}{h_1 h_2}, \tag{53}$$

in each Cartesian direction. They coincide in this case to the standard finite elements for quadrilaterals. The functions $\varphi_j^l(\boldsymbol{x})\boldsymbol{e}_i$ for $j = 1, 2, 3$ and $i = 1, 2$ provides a basis for the affine space $\mathcal{P}$. Let $\boldsymbol{\varphi}_i(\boldsymbol{x}) = \varphi(\boldsymbol{x})\boldsymbol{e}_i$. We check directly, using the symmetry of the domain, that

$$\boldsymbol{\pi}^{\nabla}(\boldsymbol{\varphi}_i) = 0.$$

Hence, for each basis functions in (53), we have that the zero and first order moments correspond to those of their projection so that, indeed, they form a basis of $\mathcal{V}$. Moreover $\{\boldsymbol{\varphi}_i\}_{i=1,2}$ constitutes a basis for $\ker \boldsymbol{\pi}^{\nabla}$. In this two-dimensional case, the matrix $\boldsymbol{N}_c$ is given by

$$\boldsymbol{N}_c^i = \begin{pmatrix} h_1 & 0 & \frac{1}{\sqrt{2}}h_2 \\ 0 & h_2 & \frac{1}{\sqrt{2}}h_1 \end{pmatrix}$$

We collect the contributions of the four nodes of the cell and obtain the matrix $\boldsymbol{N}_c$ given by

$$\boldsymbol{N}_c^T = \begin{pmatrix} h_1 & 0 & -h_1 & 0 & -h_1 & 0 & h_1 & 0 \\ 0 & h_2 & 0 & h_2 & 0 & -h_2 & 0 & -h_2 \\ \frac{1}{\sqrt{2}}h_2 & \frac{1}{\sqrt{2}}h_1 & \frac{1}{\sqrt{2}}h_2 & -\frac{1}{\sqrt{2}}h_1 & -\frac{1}{\sqrt{2}}h_2 & -\frac{1}{\sqrt{2}}h_1 & -\frac{1}{\sqrt{2}}h_2 & \frac{1}{\sqrt{2}}h_1 \end{pmatrix} \tag{54}$$

which yields

$$\boldsymbol{N}_c^T\boldsymbol{N}_c = \begin{pmatrix} 4h_1^2 & 0 & 0 \\ 0 & 4h_2^2 & 0 \\ 0 & 0 & 2(h_1^2+h_2^2) \end{pmatrix} \tag{55}$$

so that $\operatorname{tr}(\boldsymbol{N}_c^T\boldsymbol{N}_c) = 6(h_1^2+h_2^2)$. Hence, the scaling ratio α_G is given by

$$\alpha_G = \frac{4h_1h_2\operatorname{tr}(\hat{\boldsymbol{C}})}{6(h_1^2+h_2^2)} = \frac{2}{3}\frac{\operatorname{tr}(\hat{\boldsymbol{C}})}{(\varepsilon+\varepsilon^{-1})}, \tag{56}$$

where $\varepsilon = \frac{h_1}{h_2}$ denotes the aspect ratio. Let us now compute how this weight in the stabilization term compares with the actual energy for the functions that belong to $\ker \boldsymbol{\pi}^{\nabla}$. To do so, we consider an isotropic material where the stress is given as

$$\boldsymbol{\sigma} = \lambda\operatorname{tr}(\boldsymbol{\varepsilon}) + 2\mu\boldsymbol{\varepsilon}, \tag{57}$$

which implies

$$a(\boldsymbol{u},\bar{\boldsymbol{u}}) = \int_{\Omega}(\lambda\operatorname{tr}(\boldsymbol{\varepsilon})\operatorname{tr}(\bar{\boldsymbol{\varepsilon}}) + 2\mu\boldsymbol{\varepsilon} : \bar{\boldsymbol{\varepsilon}})\,\mathrm{d}\boldsymbol{x} \tag{58}$$

For $\boldsymbol{\varphi}_i$ we denote by $\boldsymbol{\varepsilon}_i$, the corresponding strain, which is given by

$$\boldsymbol{\varepsilon}_i = \frac{1}{2}(\boldsymbol{e}_i\nabla\phi^T + \nabla\phi\boldsymbol{e}_i^T). \tag{59}$$

We get

$$\boldsymbol{\varepsilon}_i : \boldsymbol{\varepsilon}_j = \frac{1}{2}\left(\delta_{i,j}\left|\nabla\varphi\right|^2 + \frac{\partial\varphi}{\partial x_i}\frac{\partial\varphi}{\partial x_j}\right),$$

where $\delta_{i,j} = 1$ if $i = j$ and zero otherwise. Hence, using the symmetry of the domain, we get

$$\int_E \boldsymbol{\varepsilon}_i : \boldsymbol{\varepsilon}_j\,\mathrm{d}\boldsymbol{x} = \delta_{i,j}\int_E\left|\nabla\varphi\right|^2\,\mathrm{d}\boldsymbol{x}.$$

We have $\operatorname{tr}(\boldsymbol{\varepsilon}_i) = \frac{\partial\varphi}{\partial x_i}$. Hence, using the symmetry of the domain we get

$$\int_E \operatorname{tr}(\boldsymbol{\varepsilon}_i)\operatorname{tr}(\boldsymbol{\varepsilon}_j)\,\mathrm{d}\boldsymbol{x} = \delta_{i,j}\int_E\left|\frac{\partial\varphi}{\partial x_i}\right|^2\,\mathrm{d}\boldsymbol{x}.$$

Finally, the restriction of the bilinear form a to $\ker \boldsymbol{\pi}^{\nabla}$ takes the form

$$a(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j) = \begin{pmatrix} \int_E (\lambda \left|\frac{\partial \varphi}{\partial x_1}\right|^2 + 2\mu \left|\nabla \varphi\right|^2) \,\mathrm{d}\boldsymbol{x} & 0 \\ 0 & \int_E (\lambda \left|\frac{\partial \varphi}{\partial x_2}\right|^2 + 2\mu \left|\nabla \varphi\right|^2) \,\mathrm{d}\boldsymbol{x} \end{pmatrix}$$

The integrals above can be computed exactly and we have

$$\int_E \left|\frac{\partial \varphi}{\partial x_1}\right|^2 \,\mathrm{d}\boldsymbol{x} = \frac{4}{3}\varepsilon^{-1}, \quad \int_E \left|\frac{\partial \varphi}{\partial x_2}\right|^2 \,\mathrm{d}\boldsymbol{x} = \frac{4}{3}\varepsilon,$$

Hence,

$$a\left(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j\right) = \begin{pmatrix} \frac{4}{3}\lambda\varepsilon^{-1} + \frac{8}{3}\mu(\varepsilon + \varepsilon^{-1}) & 0 \\ 0 & \frac{4}{3}\lambda\varepsilon + \frac{8}{3}\mu(\varepsilon + \varepsilon^{-1}) \end{pmatrix}.$$

We denote by α_1 and α_2 the two eigenvalues of the matrix above. We obtain

$$\lim_{\varepsilon \to 0,\infty} \frac{\alpha_1}{\alpha_G} = \lim_{\varepsilon \to 0,\infty} \frac{\alpha_2}{\alpha_G} = \infty$$

This enables us to conclude that, when the aspect ratio ε tends either to zero or infinity, the ratios above tends to infinity so that we cannot find a constant $c > 0$, independent of the aspect ratio ε, such that

$$c a_E(\boldsymbol{u}, \boldsymbol{u}) \leq s_E(\boldsymbol{u}, \boldsymbol{u}),$$

for all $\boldsymbol{u} \in \ker \boldsymbol{\pi}^{\nabla}$. It implies that the stabilization term is not stable with respect to the aspect ratio.

5.2 An alternative choice of the stabilization scaling

Instead of using α_G, let us use

$$\alpha_N = \frac{1}{9}|E| \operatorname{tr}(\hat{\boldsymbol{C}}) \operatorname{tr}(\operatorname{inv}(\boldsymbol{N}_c^T \boldsymbol{N}_c)). \tag{60}$$

Both α_N and α_G are invariant with respect to rotation. Because of the coefficient $\frac{1}{9}$, we have that if $\boldsymbol{N}_c^T \boldsymbol{N}_c$ *were* diagonal with constant coefficient, then α_N and α_G would be equal. But in general they differ and we have

$$\alpha_N = \frac{2\lambda + 6\mu}{4} \left(\varepsilon + \varepsilon^{-1} + \frac{2}{\varepsilon + \varepsilon^{-1}}\right) \tag{61}$$

It implies that

$$\lim_{\varepsilon \to 0} \frac{\alpha_1}{\alpha_N} = \lim_{\varepsilon \to \infty} \frac{\alpha_2}{\alpha_N} = \frac{16}{3} \frac{\lambda + 2\mu}{\lambda + 3\mu}$$

and

$$\lim_{\varepsilon \to \infty} \frac{\alpha_1}{\alpha_N} = \lim_{\varepsilon \to 0} \frac{\alpha_2}{\alpha_N} = \frac{16}{3} \frac{\mu}{\lambda + 3\mu}.$$

Therefore, for this choice of α, there exist two constants $c_1, c_2 > 0$ which are *independent* of the aspect ration ε and such that

$$c_1 a_E(\boldsymbol{u}, \boldsymbol{u}) \leq s_E(\boldsymbol{u}, \boldsymbol{u}) \leq c_2 a_E(\boldsymbol{u}, \boldsymbol{u})$$

for all $\boldsymbol{u} \in \ker \boldsymbol{\pi}^{\nabla}$. We can conclude that the stabilization provided by α_N is stable with respect to the aspect ratio, at least for quadrilaterals. Let us now try to explain the motivation back the introduction of α_N. We denote by λ_i the singular values of N_c and introduce the following averages

$$\lambda_{\text{arithm}} = (\sum_{i=1}^{d} \lambda_i^2)^{\frac{1}{2}} \quad \text{and} \quad \lambda_{\text{harm}} = (\sum_{i=1}^{d} \lambda_i^{\frac{1}{2}})^2,$$

which, for simplicity, we refer to as *arithmetic* and *harmonic* averages. Note that the matrix $\boldsymbol{N}_c$, which is given in (26), accounts for the geometry and the unit of each coefficient is a unit length. We could therefore interpret the values of λ_{arihm} and λ_{harm} as characteristic lengths of the cell. Using these values, we can rewrite the scaling coefficients as

$$\alpha_G = \frac{1}{\lambda_{\text{arithm}}^2} |E| \operatorname{tr}(\hat{\boldsymbol{C}}) \quad \text{and} \quad \alpha_N = \frac{1}{\lambda_{\text{harm}}^2} \frac{|E|}{9} \operatorname{tr}(\hat{\boldsymbol{C}}),$$

so that the difference between the two scalings is that they consider different type of averages. Let us use eigenmodes to estimate the energy in each direction. For simplicity, we consider the Laplace equation and the normalized energy of the mode $\phi_i(x) = \cos(\frac{\pi}{2h_i} x_i)$ in the i-th direction is given by

$$\frac{\int_K |\nabla \phi|^2 \, dx}{\int_K |\phi|^2 \, dx} = \frac{\pi^2}{(2h_i)^2},$$

from which we infer that a typical scale for the energy in the direction x_i is given by $\frac{1}{h_i^2}$. If we consider a linear combination of such unidirectional functions and neglect the interactions between them, then we are naturally led to consider the sum

$$\sum_{i=1}^{d} \frac{1}{h_i^2}$$

as a typical scale for the energy. To obtain a typical length, we end up by taking the harmonic average as defined above.

6 Numerical test cases

The great advantage of VEM methods is that they are valid for very general grids including non-convex cells and more than one face between two cells, [5]. This property can be used to avoid curved faces on general cells, simply by triangulating the surface. The VEM theory does not cover curved surfaces and in the next examples we investigate the need for triangulation in 3D.

6.1 A two-dimensional compaction case

Case description: We consider a rectangular domain made of an isotropic material with the following properties, $\rho = 3 \times 10^3 \mathrm{kg\,m}^{-3}$, $E = 3 \times 10^8 \mathrm{Pa}$ and $\nu = 0.3$. The vertical length of the grid is $L_y = 15\mathrm{m}$ and the horizontal length will by determined by the aspect ratio L_y/L_x. Different values of the aspect ratios will be tested. The boundary conditions are zero displacement at the bottom, rolling boundary condition on the sides, that is no displacement in the normal direction and no force in the tangential direction. At the top, we have no force and free displacement. Even if the model is two-dimensional, we have to set up boundary conditions for the third dimension, perpendicular to the plane, as the material is going to expand or withdraw in this direction due to the Poisson ratio. We impose zero displacement in the perpendicular direction, the other standard choice being no force in that direction. The load term is gravitation, that is, a constant vertical vector pointing downwards and we simulate the situation where the material is going to subside by the effect of its own weight, hence the name of *compaction*. An analytical solution is available for this case and given by

$$\boldsymbol{u} = [0, \gamma(L_y^2 - (y - L_y)^2)] \quad \gamma = \frac{g\rho}{2C_{2,2}}, \tag{62}$$

where $C_{2,2}$ is the second diagonal coefficient of the stiffness matrix $\boldsymbol{C}$. We start with a Cartesian grid that we twist in order to avoid artifact effects from symmetries. We will refer to this grid as the *twisted grid*. We consider a variation of this grid where we add extra degrees of freedom in the form of extra nodes on the horizontal edges, see Figure 6.1. The motivation for introducing such extra nodes is explained in the next paragraph.

Results: We test the three different implementations of the load term, as described in Section 4. It is important to note that the first alternative, which uses the projection operator, see the definition in (36), is *exact* in the case we are considering. Indeed, since $\boldsymbol{f} = \rho \boldsymbol{g}$ is constant, we have

$$\boldsymbol{\pi}_i^0(\boldsymbol{f}) = \boldsymbol{f}$$

so that no error is introduced by the assembly of the load term. In the remaining, we will refer to this implementation of the load as the *exact* load term. In comparison, the third method is not exact, as the potential function, here given by $\psi = \rho g y$, is approximated by a cell-valued function. For the stabilization term, we test the two scaling variables α_G and α_N presented in Section 5.

We start with the scaling variable α_G taken from [9] and the exact load implementation. We use the grid with extra nodes. For such grid, each cell gets two extra degrees of freedom. However, these extra degrees of freedom do not enrich the approximation space as they do in the case of a finite element method. The VEM method retains the same degree of accuracy, that is first order in our case. The extra basis functions introduced by the extra degrees of freedom are handled by the stabilization term. But the stabilization term only guarantees that these extra functions do not break the ellipticity of the system but it is an artificial term which cannot add any accuracy. Therefore, by adding an extra node on the edges, we increase the relative importance of the stabilization term, so that its deficiency will be more apparent. As predicted by the results of Section 5, we observe a severe dependence on the aspect ratio. When the aspect ratio is minimal, that is $L_x/L_y = 1$, then the solution is close to the analytical one but, when the aspect ratio is increased to $L_x/L_y = 10$, by stretching the grid in the horizontal direction, the results deteriorate severely, see the top panels in Figure 3. We run the same simulations but, instead of the exact load term, we use the load term computed by the discrete gradient operator. Then, the results do not deteriorate as the aspect ratio is increased.

In Figure 4, we plot the error in displacement as a function of the aspect ratio (from 1 to 100) for the different grid cases and the three implementation of the load term. The left figure shows that the exact load and nodal load calculations fail for the grid with extra nodes. The error apparently follows a second order growth, that is $err \sim (L_y/L_x)^2$. The plot on the right shows the error for the twisted grid without extra nodes for the exact load computation and the error for both grids for the discrete gradient approach. All the methods give reasonable results, but the exact load calculation seems to deteriorate more than the others. The discrete gradient approach is stable in both cases. Note that, if we had used a grid without disturbance, all the methods would give exact results for the grid without extra nodes on the faces while the extra node case will still fail for the exact load calculation. The reason is that, in the non disturbed case with no extra nodes, all the implementations of the load term give the same result in the case of a constant vertical load term.

Finally in Figure 5, we consider the scaling α_N introduced in (60), which is stable with respect to aspect ratio. The error does not grow as the aspect ratio is increased, as opposed to α_G. The use of α_N deteriorates the solution computed using the discrete gradient approach, while it significantly improves the solution using the exact method. However, this conclusion is difficult to extend to more general cases. The value of α_N has been derived from an analysis done on regular quadrilaterals and we observe that the stability properties extend to a twisted Cartesian grid. However, separate studies would have to be done for more complicated shapes and also in 3D, where the situation is expected to be more complicated. Indeed, while in 2D the aspect ratio is described by a scalar quantity namely $\varepsilon = \frac{\Delta x}{\Delta y}$, in 3D we need 2 values, say $\frac{\Delta x}{\Delta z}$ and $\frac{\Delta y}{\Delta z}$, the third quantity $\frac{\Delta x}{\Delta y}$ being imposed by the fact that we will anyway require isotropic stability. It means that a scalar approximation of the stabilization term, as given in (51) and also in [3], will not be enough. This problem was noticed in [2], and the exact stabilization term corresponding to finite element was used there to study a poro-elastic response function in the 3D case.

Comment: We do not really understand why the discrete gradient approach (Alternative 2) performs significantly better than the projection approach which is exact in this case (Alternative 1). However, we note some fundamental differences between the force-based methods (Alternatives 1 and

2) and the discrete gradient approach, which may help to understand the differences in the results. As explained in the previous section, see (34), the difference between the methods is in how they divide the weights between the nodes. All the force-based methods divide forces according to a weight for each node associated with volume integrals. These weights are equal for *all* Cartesian directions. In contrast, the discrete gradient method uses weights associated with surface integrals, so that the weights can depend on the direction, and the corresponding degrees of freedom. These weights can be associated with the projected area of the faces associated with a node divided by the projection of the cell in the same direction. This is most easily seen from the expression in equation (43). In the case of the extra nodes on the edges, these nodes will have associated weights in the horizontal direction only due to the tilt of the grid and the weights will in the simple case be doubled of the corner nodes while the exact case will give all nodes the same weights. In [9], the method using node quadrature (Alternative 2) is considered, this will in the above case give a smaller weight to the midpoint and behave worse for the case with extra node, as seen in the left panel of Figure 4.

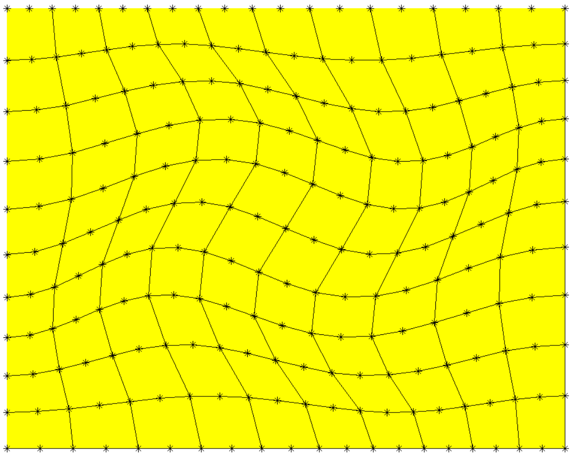


Fig. 2 A twisted Cartesian grid is obtained by starting from a regular Cartesian grid and moving the nodes, here by using a smooth given displacement field. We plot the grid that is obtained after adding one extra node on each horizontal edge. Such grid is used to demonstrate the failure of the stabilization term where the aspect ratio is increased. The grid plotted here is the reference grid with aspect ratio, by definition, equal to $L_x/L_y = 1$.

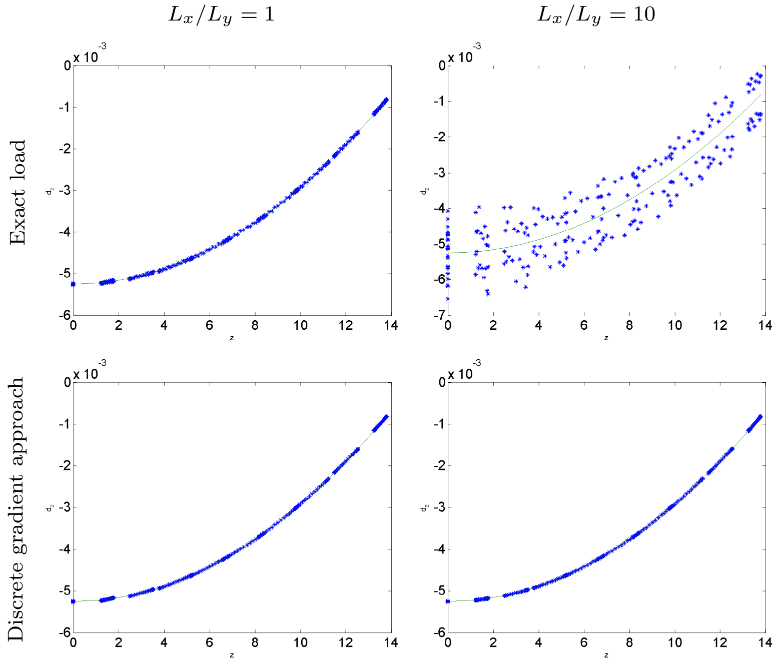


Fig. 3 We plot the computed displacement in the vertical direction for the 2D compaction example. The result for every node of the grid is represented as a dot, where the x-coordinate of the node corresponds to the vertical position of the node and the y-coordinate corresponds to the value of the vertical displacement computed at the node. The analytical solution is plotted as a continuous line. For these plots, the twisted Cartesian grid with an extra node on each horizontal edges, see Figure 6.1, has been used. The left column is for aspect ratio 1 and the right is for aspect ratio 10. For the first row, the exact load calculation based on the exact integration of the VEM basis function has been used, while the lower row corresponds to the discrete gradient approach. We use the scaling factor α_G as proposed in [9], see (52). We observe that, for the exact load computation, the solution quickly deteriorates when the aspect ratio is increased while the results remain good for the discrete gradient approach.

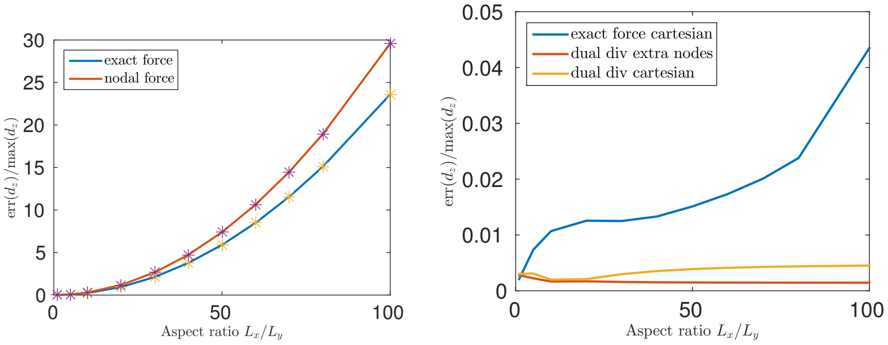


Fig. 4 Plots of the maximum error in the vertical displacement as a function of the grid aspect ratio. The left figure shows results for the exact integration method in the case with extra nodes on horizontal faces. We observe that the method fails as the error blows up. The extra points in this plot are reference points that indicate a quadratic scaling of the error with respect aspect ratio. In the right figure, the results are shown for the exact method on the twisted grid without extra nodes and for the discrete gradient approach on the same grid with and without the extra nodes.

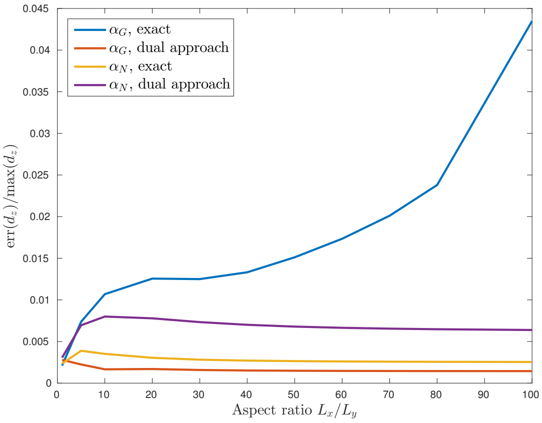


Fig. 5 Comparison between the two scaling constants α_G and α_N. We plot the error of the vertical displacement, as the aspect ratios is increased. Here, we use the grid without the extra nodes. We observe that the scaling constant α_N yields stability with respect to aspect ratio, independently of which method is used to compute the load term.

6.2 Compaction 3D

In order to investigate the performance of the VEM method on real reservoir geometries, we use two grids which includes standard features of subsurface models. The first one is based on a local sedimentary model called *sbed* . The model was used for upscaling permeability. Our version is $15m \times 15 \times 3m$ with logical Cartesian dimensions $15 \times 15 \times 333$. The grid reflects two of the basic properties of a sedimentary process, which are the layering and erosion processes. For this type of grid, the challenge is the degenerate cells and the large aspect ratios. The second model that we consider is taken from the open reservoir model of Norne. The data for this model is freely available in the open dataset of the Open Porous Media initiative [14]. We extract a part of this model, pad it on all sides to embed it in a regular prism, so that we can simply impose side boundary conditions and directly compare the solution with the analytic solution of a pure gravitational compression. The final full model and the embedded model with faults are shown in Figure 9.

Both models use a corner-point grids, which is a standard in the industry. A corner-point grid has an underlying two dimensional structure which is used to index the pillars. Let us denote by $p_{i,j}$ and $q_{i,j}$ the bottom and top and the pillar that is indexed by (i, j). For each region contained between the four pillars (i, j), $(i + 1.j)$, $(i + 1, j + 1)$ and (i, j), points are defined on each of this pillar in equal number. We denote those points by $x^k_{i',j'}$ for $i' \in \{i, i + 1\}$ and $j' \in \{j, j + 1\}$. Then, the region between the four pillars is meshed with hexahedrons with eight corner points given such as $x^{k'}_{i',j'}$ for $i' \in \{i, i + 1\}$, $j' \in \{j, j+1\}$ and $k' \in \{k, k+1\}$. This construction naturally leads to irregular cell shapes and faces that are not planar, see the illustrations given in Figure 6 . Therefore we end up outside the theoretical framework of the VEM method, which only cover planar polygonal faces. However, the computation of the stiffness matrix for VEM relies on geometrical properties that are all available, either as exact or approximated values (such as face areas, face normals, etc.), in the case of a corner-point grid, so that the stiffness matrix can be assembled and a solution computed. To evaluate the error that is introduced by this geometrical approximation, we compare the solution obtained this way with

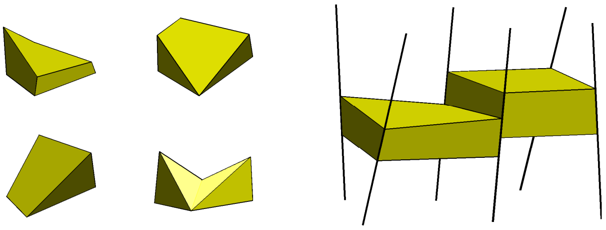


Fig. 6 On the right, two neighboring cells of a corner-point grid. On the left, examples of the irregular cell shapes that the corner-point format can produce.

the solution that is obtained after triangulating the non planar surfaces, by adding a point in the middle of the faces. For such grid, the faces will be planar and the theoretical framework of the VEM method applies.

In Figure 7, we show the effect of compression with two types of load given by a constant gravitational force and a constant load applied on the top surface. For both loads, the analytical solutions can be computed and they are respectively, quadratic and linear in z. We consider both the original corner-point grid and the triangulated grid. By triangulated grid, we mean a grid where the faces ares triangulated, as we just explained. For all these cases, the VEM method gives accurate results, given that we use the discrete gradient approach to compute the load term. The other alternatives simply fail in this case, by errors that are larger than the span of the exact solution. In the *sbed* model, the pillars are all vertical lines, which implies that the vertical faces are planar. For the linear case corresponding to a constant load on the top surface, we see that the triangulated version gives exact result, as predicted by the VEM theory, since in this case all the surfaces are planar and the solution is linear. For the original grid, we get an error due to the curved top and bottom faces in each cell. For the pure gravitational case, both grids give comparable results. Thus, we can conclude that in practice, it may not be worth triangulating the faces because it introduces more degrees of freedom without significantly improving the accuracy of the solution. We consider the case of a flipped model for Norne in Figure 8. In this way, we can investigate the effect of having non planar surface in the vertical direction. Typically, for the cells of the original reservoir, we have $\frac{\Delta x}{\Delta z} \approx \frac{\Delta y}{\Delta z} \gg 1$ so that, by flipping the model, we can observe the consequence of inverting the correlation between the aspect ratio and the direction of gravitation. The results are similar. However the triangulated case which is exact for linear compression highlights that the error of different types of nodes have different errors, see explanation in the caption of Figure 8.

Besides features like layering and erosion, the Norne case introduces also fault structures. Such grids are far from ideal for numerical calculation, but the VEM method shows very robust behavior. In Figure 9, we look at the difference between the original model and a model where all the pillars are straightened up and made vertical. In this way, the curved sides in the vertical direction are eliminated. The analytical solution is unchanged as we recall that the whole Norne model is anyway embedded in a regular prism. The results on Norne confirm those obtained for the *sbed* model and show that effects of curvature on the faces can be neglected. This indicates that, for many practical applications, the VEM method can be used directly on the original grid of reservoirs without deteriorating the accuracy of the results.

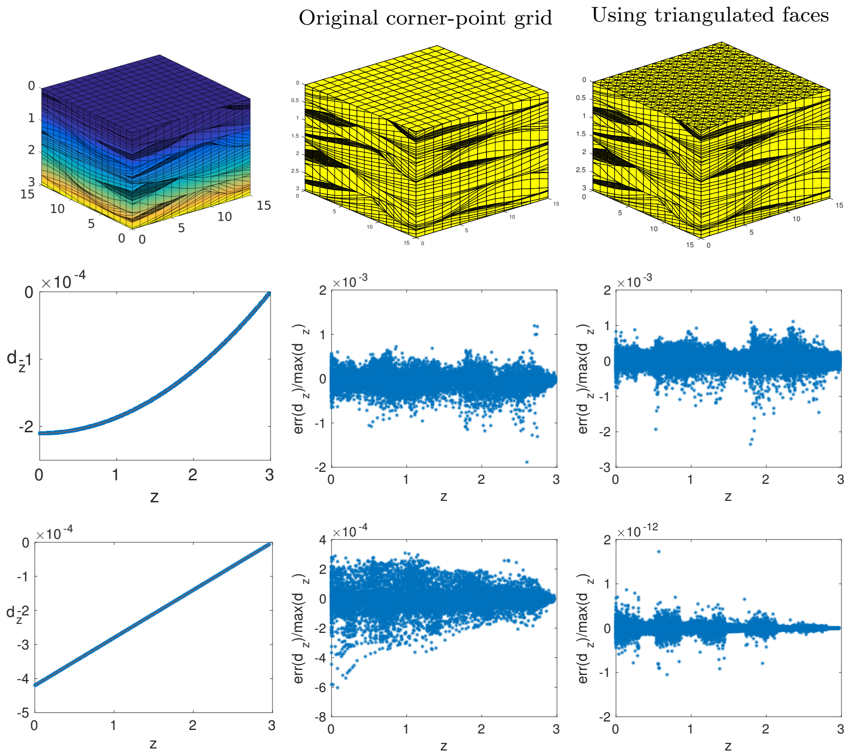
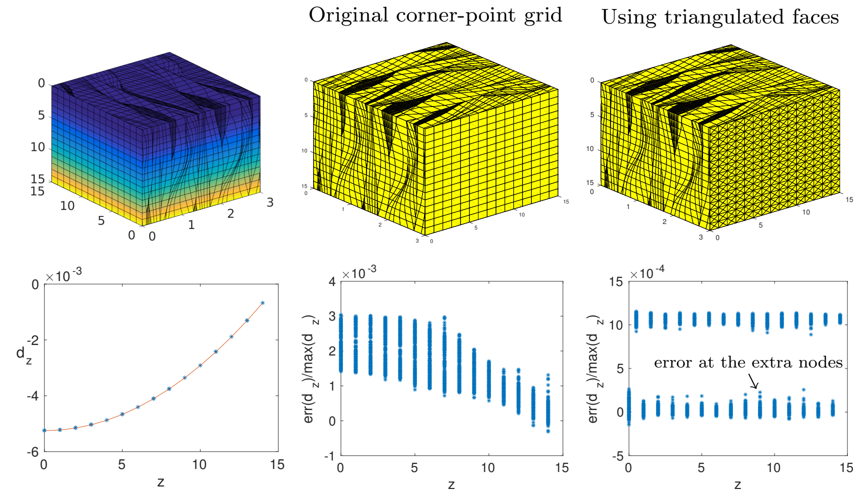


Fig. 7 Effect of compression on the *sbed* model. The first row shows a plot of the vertical deformation on the grid (left), the original grid (middle) and the same grid where the surface are triangulated (right). Two types of loads are considered: pure gravitational compression (second row), load at the top surface (third row). The first column shows the displacement obtained for each loading case, which is very close to the analytical solution. The remaining plots show the errors for the original cornerpoint grid with curved faces (middle column) and the triangulated grid with only planar faces (right column).

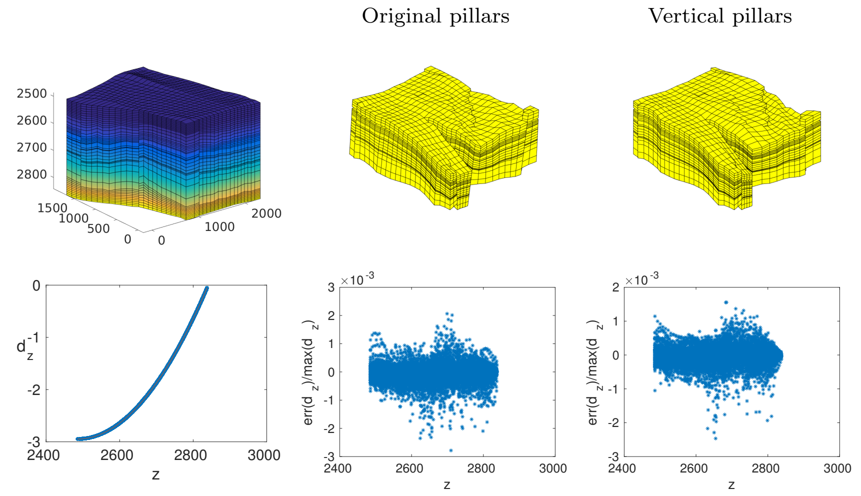


Original corner-point grid

Using triangulated faces

error at the extra nodes

Fig. 8 Effect of compression on a fliped *sbed* model. The first row shows a plot of the vertical deformation on the grid (left), the original grid (middle) and the same grid where the surface are triangulated (right). We consider only the case with gravitation load. The first column shows the displacement. The remaining plots show the errors for the original corner-point grid (middle column) and the triangulated grid (right column). On the plot at the lower right, we observe that the error splits clearly between the type of nodes, the extra face node at the bottom and the other at the top.



Original pillars

Vertical pillars

Fig. 9 Effect of compression on a part of the Norne model. The first column shows the plot the vertical deformation on the grid (left), the original grid after removing the padding (middle) and the straightened grid where the pillars are made vertical (right). The figure in the upper left corner shows the bounding box which is used for the calculation, while the two other grids show the embedded Norne grid. The second row show the results for pure gravitational compression. The first column shows the vertical displacement while the second and third show the errors in the vertical displacement for the original and triangulated grid.

7 Conclusion

We have demonstrated how geomechanical calculations can be done directly on complex geological models frequently encountered in reservoir modeling, by using the flexibility of the VEM method which can handle general geometries. In this method, the energy is not computed exactly for each basis element functions. We demonstrate that this approximation can come at the cost of large errors for deformed grids, if not care is taken when defining the approximate bilinear form. In particular we study the effect of the load term calculation and show that, with stabilization terms and load term calculations presented earlier in the literature, even simple 2D cases fails severely when the aspect ratio is increased. We found that both the choices of discretization and of the load term calculation are in combination responsible for the failure. Using the exact equivalence with FEM on quadrilateral grid, we presented a modification of the discretization that makes the method more robust in the 2D case. In addition, we demonstrated that a calculation of the load in term of a gradient of a potential was robust in 2D and the only approach which gave sufficient accuracy in 3D. This holds in particular for grid cells that are outside the reach of FEM, such as those containing hanging nodes. The VEM theory does not cover curved faces, which are common in subsurface models. We saw that for our tests the error associated with this feature was negligible comparable with other errors, with the natural exception of the case when VEM gives the exact solution (linear displacement).

Acknowledgements This work has been partially funded by the Research Council of Norway through grants no. 215641 from the CLIMIT programme.

References

[1] B. Ahmad et al. “Equivalent projectors for virtual element methods”. In: *Computers & Mathematics with Applications* 66.3 (2013), pp. 376–391.

[2] O. Andersen, H. M. Nilsen, and S. Gasda. “Modelling Geomechanical Impact of CO_2 Injection Using Precomputed Response Functions”. In: *ECMOR XV – 15^{th} European Conference on the Mathematics of Oil Recovery, Amsterdam, Netherlands, 29 August - 1 September 2016*. EAGE. 2016.

[3] L. Beirão da Veiga, F. Brezzi, and L. D. Marini. “Virtual elements for linear elasticity problems”. In: *SIAM Journal on Numerical Analysis* 51.2 (2013), pp. 794–812.

[4] L. Beirão da Veiga, K. Lipnikov, and G. Manzini. *Mimetic Finite Difference Method for Elliptic Problems*. Vol. 11. Springer, 2014.

[5] L. Beirão da Veiga et al. “Basic principles of virtual element methods”. In: *Mathematical Models and Methods in Applied Sciences* 23.01 (2013), pp. 199–214.

[6] L. Beirão da Veiga et al. "The hitchhiker's guide to the virtual element method". In: *Mathematical models and methods in applied sciences* 24.08 (2014), pp. 1541–1573.

[7] P. G. Bergan and M. K. Nygård. "Finite elements with increased freedom in choosing shape functions". In: *Int. J. Numer. Meth. Engng.* 20.4 (Apr. 1984), pp. 643–663. ISSN: 1097-0207.

[8] Z. Ding. "A proof of the trace theorem of Sobolev spaces on Lipschitz domains". In: *Proceedings of the American Mathematical Society* 124.2 (1996), pp. 591–600.

[9] A. L. Gain, C. Talischi, and G. H. Paulino. "On the Virtual Element Method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes". In: *Computer Methods in Applied Mechanics and Engineering* 282 (2014), pp. 132–160.

[10] E. J. Gringarten et al. "New Grids for Robust Reservoir Modeling". In: *SPE Annual Technical Conference and Exhibition* (2008).

[11] K. Lie et al. "Open-source MATLAB implementation of consistent discretisations on complex grids". In: *Comput. Geosci.* 16 (2 2012), pp. 297–322. ISSN: 1420-0597.

[12] B. Mallison et al. "Unstructured Cut-Cell Grids for Modeling Complex Reservoirs". In: *SPE Journal* 19.02 (Apr. 2014), pp. 340–352.

[13] *The MATLAB Reservoir Simulation Toolbox, Version 2016a.* July 2016. URL: `http://www.sintef.no/MRST/`.

[14] Open Porous Media initiative. *Open Datasets.* `http://wwww.opm-project.org`. 2015. URL: `http://www.opm-project.org`.

[15] D. K. Ponting. "Corner point geometry in reservoir simulation". In: *ECMOR I-1st European Conference on the Mathematics of Oil Recovery.* 1989.