Finite Elements in Analysis and Design 259 (2026) 104572

Contents lists available at ScienceDirect

Finite Elements in Analysis & Design

journal homepage: www.elsevier.com/locate/finel

A finite cell method for digital elevation models and its application to geology

Viktor Haunsperger [a], Jörg Robl [a], Andreas Schröder [b,*]

[a] Department of Environment & Biodiversity, University of Salzburg, Hellbrunner Straße 34, 5020 Salzburg, Austria
[b] Department of Mathematics, University of Salzburg, Hellbrunner Straße 34, 5020 Salzburg, Austria

ARTICLE INFO

MSC:
65N30
65N85

Keywords:
Finite cell method
Digital elevation model
Geology

ABSTRACT

In this article the finite cell method is adapted to a digital elevation model, that can be used to represent highly complex surface topographies of mountainous regions. The proposed approach relies on the use of two grids: a regular hexahedron grid for the use of the finite element method based on trilinear shape functions and a hexahedron grid on which the numerical integration is based. A specific tetrahedron decomposition of the integration grid guarantees that a triangle surface mesh that linearly interpolates the digital elevation model is represented exactly. The decomposition is done by selecting all hexahedrons with a non-empty intersection with this mesh and dividing them into six tetrahedrons along a prescribed diagonal. A key result of this article is that the relative interior of the edges of these tetrahedrons has at most one intersection point with the surface triangles, which enables to subdivide them into tetrahedra again using a fixed number of decomposition patterns. This procedure allows for an efficient finite element assembling routine where the resulting quadrature mesh does not need to be stored in its entirety. In numerical experiments the convergence properties of the approach are studied using several example geometries. Furthermore, the proposed finite cell method is applied to a geological application in which the stress distribution in the Hochkönig Massif (Eastern Alps) is computed. By providing accurate stress computations for the near-surface rock mass, this approach assists in evaluating rock masses close to failure and helps to assess landslide risks.

1. Introduction

Computational simulations and, in particular, the computation of stress distributions play an important role in many applications in geology and geomechanics. They are often based on stress modeling with (linear) elasticity and the use of finite element methods (FEM) [1–3]. Such simulations enable the assessment of the stability of steep topography and assists in predicting rock masses close to failure, landslides and mass movements [4]. They usually require a high computational accuracy in the surface area and therefore an accurate capturing of the surface topography. Typically, the topography of mountainous areas is described by a digital elevation model (DEM), in which, e.g., elevation values are assigned to data points of a square grid. However, the complex boundary geometries in geological applications can make the direct use of FEM problematic, particularly when very fine structures need to be resolved by meshing, as this could lead to a high number of degrees of freedom and, therewith, a high number of unknowns in the resulting system of linear equations. FEM computations using an exact representation of the topography (as given by a DEM) could therefore be very costly in terms of memory and computing time, maybe even impossible. Fictitious domain methods or immersed

* Corresponding author.
E-mail address: andreas.schroeder@plus.ac.at (A. Schröder).

https://doi.org/10.1016/j.finel.2026.104572
Received 25 January 2026; Received in revised form 21 April 2026; Accepted 23 April 2026
Available online 12 May 2026

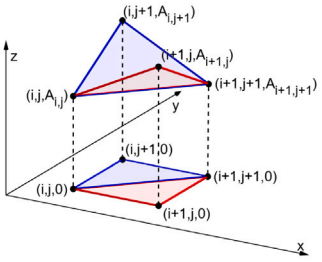
boundary methods offer a way out in this context [5,6]. They are well-established variants of the FEM, which are developed to resolve complex boundary geometries without meshing in terms of the degrees of freedom. The basic idea of these methods is to embed the physical domain with complex geometry into a domain of a simple shape (e.g., an axis-parallel rectangular hexahedron) and to mesh this embedding domain with mesh elements that also have simple shapes or other advantageous properties. FEM is then applied to the mesh of the embedding domain. The complex boundary geometry is taken into account via adapted numerical integration schemes with respect to the variational formulation of the underlying model or via other techniques like additional (penalty) boundary terms or Lagrange multipliers. This process essentially simplifies the application of the FEM since, for instance, tensor product shape functions on axis-parallel hexahedrons can be applied or the regular structure of the mesh of the embedding domain can be further exploited.

One of the important fictitious domain methods is the finite cell method (FCM) [7,8], which has become widely established in recent years. The FCM enables computations on highly complex geometries and has been applied to a vast number of problems including thermo-elasticity [9] or other multi-physics problems, e.g. piezoelectricity [10], (geometrical) non-linearities [11,12], bio-mechanics [13–15], elasto-plasticity [16], foamed materials [17,18] as well as shell problems [19], wave propagation [20–22] and topology optimization [23]. The method is characterized by the fact that the integrands of the variational formulation of the problem are weighted by a factor that is 1 in the physical domain and that has a small value in the fictitious domain (i.e. the embedding domain without the physical domain). This guarantees that the variational formulation is still uniquely solvable. However, the introduction of the weighting factor leads to a model error that has to be taken into account, for instance, in the convergence analysis or the derivation of error controls [24–28]. The main focus of the FCM lies on the application of specific numerical integration schemes that are adapted to the boundary geometry and that enables computation of the weighted integrals with sufficient accuracy. There are a variety of numerical integration techniques that are used within the FCM. These techniques include, for instance, the octree- or spacetree-based generation of integration grids which can be extended by strategies for adaptive positioning of subdivision points (e.g., smart octrees) [29,30], by data compression techniques to decrease the number of integration sub-cells [31,32] or by problem-specific strategies, such as in the case of porous domains, where boolean finite cell methods can be applied [33,34]. Furthermore, the position of the integration points and weights in quadrature rules can be optimized [35] or, if the integration points are fixed, only the weights can be computed in such a way so that the resulting quadrature rule has a high order. The latter approach is known as moment fitting [36] and can be combined with the divergence theorem in order to determine the integrals of the moment fitting approach by computing boundary integrals [37] or with techniques to improve the numerical stability of the numerical integration scheme, e.g., non-negative moment fitting techniques [38,39].

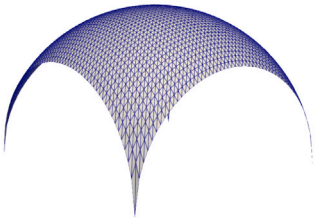
In this paper we introduce an FCM approach that is specifically adapted to the use of a DEM that is defined by elevation values on a regular square grid with equidistant points. We call this approach DEM-FCM. The focus is on the computation of displacements and stress distributions in linear elasticity and applications to geology. In the approach the DEM is simply given by a matrix in which the elevation values are stored (in the sense of z-coordinates). The row and column indices of the matrix are the corresponding x- and y-coordinates, i.e. the underlying square grid of the DEM is an integer grid (which can, of course, be converted into any real number grid). Generally, a DEM can also be a surface mesh of triangles or (as in our case) can be used to construct such a mesh by using three neighboring data points of the square grid. Here, we select two sets of three data points along the diagonal from the bottom left to the top right of a square to construct a triangle surface mesh (referred to here as DEM surface), i.e. each square of the DEM grid produces two triangles of the surface mesh with this specific orientation. The proposed DEM-FCM relies on the use of two grids: first, a regular grid that consists of axis-parallel rectangular hexahedrons and that is used for the FEM (FEM grid) and second, a grid of axis-parallel rectangular hexahedrons on which the numerical integration is based and which we call a DEM grid. To be more specific, we call a hexahedron decomposition a DEM grid if all elements with a non-empty intersection with the DEM surface have a length of 1 in the direction of their x- and y-coordinates. This ensures that the hexahedrons with this property coincide exactly with the integer square grid of the DEM with respect to these coordinates. To generate such a DEM grid we propose a simple octree-based decomposition algorithm and show its efficiency in terms of the size of the DEM. The fundamental idea for the use of a DEM grid in the numerical integration is to decompose all hexahedrons with a non-empty intersection with the DEM surface into six tetrahedrons, where the decomposition is done along the diagonal that already defines the triangle surface mesh. We show that the relative interior of the edges of these tetrahedrons has at most one intersection point with the surface triangles, so that they can be subdivided into tetrahedrons using a fixed number of decomposition patterns (namely four). This subdivision ensures that the DEM surface is represented exactly by the facets of the resulting tetrahedrons and can therefore be seen as a DEM fitted tetrahedron decomposition. The decomposition patterns for the tetraherdron subdivisions are similar to those proposed in [40], where decomposition patterns for tetrahedrons as well as hexehedrons are introduced in the context of a marching volume polytope algorithm.

The underlying FEM approach is based on trilinear shape functions on hexahedrons. An extension of the DEM-FCM approach to higher-order shape functions will be investigated in a subsequent work. For the computation of the weighted integrals of the variational formulation we use a DEM grid (as introduced above) and, in particular, DEM fitted tetrahedron decompositions. Furthermore, we apply exact quadrature rules for polynoms on tetrahedrons [41]. The stiffness matrix and the load vector are assembled using connectivity matrices that assign the components of the local stiffness matrices and local load vectors to those of their global counterparts. It is essential that a DEM grid and the DEM fitted tetrahedron decompositions are only generated for each single hexahedron of the FEM grid during the loop of this assembling process. This means that the mesh used for numerical integration does not need to be stored in its entirety, which significantly reduces memory requirements.

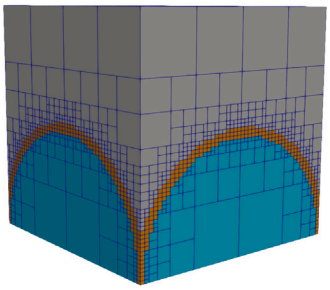
In several numerical experiments we study the convergence properties of the DEM-FCM and verify whether it yields convergence results similar to those obtained by a standard lowest-order finite element method. For this purpose, we perform the same numerical



(a) S_{ij}^{+} in blue and S_{ij}^{-} in red.



(b) The sets $\mathcal{S}$ and $\mathbb{S}$.



(c) DEM grid $\mathcal{G}$ with the sets $\mathcal{G}^{*}$ in orange, $\mathcal{G}^{-}$ in turquoise and $\mathcal{G}^{+}$ in gray.

Fig. 1. Visualization of S_{ij}^{+} and S_{ij}^{-}, $\mathcal{S}$, $\mathbb{S}$ and $\mathcal{G}$.

experiments with the FEM software ELMER [42] and compare them to the results of the DEM-FCM. In the experiments we use simple example geometries in order to ensure a best possible comparison. It is shown that the DEM-FCM and ELMER yield very similar convergence results with the expected convergence orders. The DEM-FCM shows slightly better convergence behavior when the discretization error is plotted against the degrees of freedom.

The applicability of DEM-FCM is discussed in a geological application in which the stress distribution in the Hochkönig Massif is computed in order to enable the prediction of rock failures and landslides in this mountainous area. The DEM-FCM allows for stress computations with very high spatial resolution of the surface, as the DEM data set is being fully exploited. The results of the computational simulations is in accordance with the current state of geological knowledge and show that the DEM-FCM can be applied as an efficient tool for stress simulations based on digital elevation models. We refer to [43] for further application-oriented simulations in geology.

The paper is structured as follows: Section 2 presents the matrix-oriented DEM and the DEM surface on which the DEM-FCM is based. In Section 3 the concept of DEM grids is introduced and the octree-based algorithm for the generation of DEM grids is discussed. DEM-fitted tetrahedron decompositions are introduced in Section 4. In particular, all possible decomposition patterns are specified. Section 5 introduces the DEM-FCM, describes appropriate algorithms for the assembling of the stiffness matrix and load vector and specifies the incorporation of boundary conditions. The numerical verification of the DEM-FCM is discussed in Section 6. The application of the DEM-FCM to a geological problem is described in Section 7.

In this paper the following notations are used: The superscript $\pm$ indicates that a statement or definition with this superscript applies to both the $+$ and $-$ variants; any subsequent $\pm$ or $\mp$ should be interpreted accordingly. The open sphere with radius $\epsilon > 0$ at the point $P \in \mathbb{R}^3$ is defined as $B_\epsilon(P) := \{Q \in \mathbb{R}^3;\ \|Q - P\| < \epsilon\}$.

2. A matrix-oriented DEM

A DEM that is defined by evaluation values on a regular square grid with equidistant points can be represented by a matrix $A \in [a,b]^{n\times m}$ for some given $n, m \in \mathbb{N}$, $a, b \in \mathbb{R}$ and $a < b$. The elevation at a point $(i,j) \in \{1,\dots,n\}\times\{1,\dots,m\}$ is represented by the matrix component A_{ij}. To describe a surface represented by this DEM we define the triangles $S_{ij}^{\pm} := \operatorname{conv}(V_{ij}, V_{ij}^{\pm}, \hat{V}_{ij})$ for $1 \le i < n$ and $1 \le j < m$ with $i, j \in \mathbb{N}$ by using the vertices

$$V_{ij} := (i, j, A_{ij}), \quad V_{ij}^{+} := (i, j+1, A_{i,j+1}), \quad V_{ij}^{-} := (i+1, j, A_{i+1,j}), \quad \hat{V}_{ij} := (i+1, j+1, A_{i+1,j+1})$$

where $\operatorname{conv}(P_1,\dots,P_k)$ denotes the convex hull of the points $P_1,\dots,P_k \in \mathbb{R}^d$ with $k \in \mathbb{N}$ and $d \in \{2,3\}$. Note that the triangles $S_{ij}^{\pm}$ result from dividing the rectangle $[i, i+1]\times[j, j+1]$ along the diagonal from (i,j) to $(i+1, j+1)$ and taking the corresponding entries of the matrix A as third vertex coordinates. We refer to Fig. 1(a), where these triangles are illustrated. The union of all these triangles is denoted by

$$\mathcal{S} := \bigcup_{\substack{1\le i<n\\ 1\le j<m}} \{S_{ij}^{+}, S_{ij}^{-}\}.$$

The set $\mathcal{S}$ describes a surface mesh of triangles with the nodes (i, j, A_{ij}), $1 \le i \le n$ and $1 \le j \le m$. The DEM surface is given by

$$\mathbb{S} := \bigcup \mathcal{S},$$

where $\bigcup \mathcal{M} := \bigcup_{M\in\mathcal{M}} M$ for $\mathcal{M} \subset \mathcal{P}(\mathbb{R}^3)$. Defining the mapping $\mathfrak{s} : [1,n]\times[1,m] \to [a,b]$ as

$$\mathfrak{s}(x,y) := \begin{cases} (i+1-x)A_{ij} + (y-j)A_{i+1,j+1} + (j-i+x-y)A_{i+1,j}, & (x,y) \in \operatorname{conv}((i,j),(i+1,j),(i+1,j+1)),\\ (j+1-y)A_{ij} + (x-i)A_{i+1,j+1} + (j-i+x-y)A_{i,j+1}, & (x,y) \in \operatorname{conv}((i,j),(i,j+1),(i+1,j+1)) \end{cases}$$

we find that $\mathbb{S}$ is the graph of $\mathfrak{s}$ (i.e. $\mathbb{S} = \{(x,y,\mathfrak{s}(x,y));\ (x,y) \in [1,n]\times[1,m]\}$) since $\mathfrak{s}$ is a piecewise linear interpolant to the nodes (i, j, A_{ij}). Obviously, $\mathbb{S}$ is simply connected and $\mathfrak{s}$ is continuous.

Algorithm 1 Generating a DEM grid

```
function GenerateDEMGrid(Ĥ ∈ ℋ)
    𝒢₀ ← ∅
    𝒢₁ ← ∅
    𝒢₂ ← ∅
    𝒟 ← {Ĥ}
    repeat
        for all H = [i, i+k] × [j, j+l] × [z, z+v] ∈ 𝒟 do
            𝒟 ← 𝒟 \ {H}
            if int(H) ∩ 𝕊 ≠ ∅ then
                if k = l = 1 then
                    𝒢₀ ← 𝒢₀ ∪ {H}
                else
                    𝒟 ← 𝒟 ∪ ℛ(H)
                end if
            else
                if z < A_ij then
                    𝒢₁ ← 𝒢₁ ∪ {H}
                else
                    𝒢₂ ← 𝒢₂ ∪ {H}
                end if
            end if
        end for
    until 𝒟 = ∅
    return (𝒢₀, 𝒢₁, 𝒢₂)
end function
```

The edges of $S_{ij}^{\pm}$ are denoted by $e_{ij} := \operatorname{conv}(V_{ij}, \hat{V}_{ij})$, $e_{ij}^{\pm} := \operatorname{conv}(V_{ij}, V_{ij}^{\pm})$ and $\hat{e}_{ij}^{\pm} := \operatorname{conv}(\hat{V}_{ij}, V_{ij}^{\pm})$. We set $\mathcal{E}_{ij}^{\pm} := \{e_{ij}, e_{ij}^{\pm}, \hat{e}_{ij}^{\pm}\}$ and observe that the edges of $S_{ij}^{\pm}$ are each contained in the stripes $\mathcal{K}_{ij}^{\pm} := \{K_{ij}, K_{ij}^{\pm}, \hat{K}_{ij}^{\pm}\}$ with $K_{ij} := \operatorname{conv}((i,j),(i+1,j+1)) \times \mathbb{R}$ and

$$K_{ij}^{+} := \{i\} \times [j, j+1] \times \mathbb{R}, \quad \hat{K}_{ij}^{+} := [i, i+1] \times \{j+1\} \times \mathbb{R}, \quad K_{ij}^{-} := [i, i+1] \times \{j+1\} \times \mathbb{R}, \quad \hat{K}_{ij}^{-} := \{i+1\} \times [j, j+1] \times \mathbb{R},$$

more precisely $e_{ij} \subset K_{ij}$, $e_{ij}^{\pm} \subset K_{ij}^{\pm}$ and $\hat{e}_{ij}^{\pm} \subset \hat{K}_{ij}^{\pm}$. Fig. 1(b) shows the sets S and $\mathbb{S}$ for the matrix $A \in \mathbb{R}^{(2r+1)\times(2r+1)}$ with $r = 25$ and

$$A_{ij} := 1 + \left(2r^2 - (i-r-1)^2 - (j-r-1)^2\right)^{1/2}, \qquad 1 \le i, j \le 2r+1. \tag{1}$$

The matrix A represents a DEM describing the surface of the sphere with radius r and midpoint $(r+1, r+1, 1)$. We refer to [44] where datasets in the form of matrix-oriented DEMs are available for various regions in Austria.

3. DEM grids

We denote by $\mathcal{H}$ the set of all rectangular hexahedrons with vertices in $\mathbb{B} \cap (\mathbb{N}^2 \times \mathbb{R})$ with $\mathbb{B} := [1, n] \times [1, m] \times [a, b]$, i.e.

$$\mathcal{H} := \{[i, i+k] \times [j, j+l] \times [z, z+v] \subset \mathbb{B} : \ i, j, k, l \in \mathbb{N}, \ z, v \in \mathbb{R}, \ v > 0\}$$

and call k, l and v in this definition the x-, y- and z-length of a rectangular hexahedron in $\mathcal{H}$, respectively. A set $\mathcal{G}$ is called a grid of a rectangular hexahedron $\hat{H} \in \mathcal{H}$, if $\mathcal{G} \subseteq \mathcal{H}$ and $\mathcal{G}$ is a decomposition of $\hat{H}$, i.e. $\bigcup \mathcal{G} = \hat{H}$ and $\operatorname{relint}(H_1) \cap \operatorname{relint}(H_2) = \emptyset$ for all $H_1, H_2 \in \mathcal{G}$ with $H_1 \neq H_2$. By $\operatorname{int}(M)$ and $\operatorname{relint}(M)$ we denote the interior and relative interior of a subset $M \subset \mathbb{R}^3$, respectively. The set of all rectangular hexahedrons of a grid $\mathcal{G}$ of $\hat{H}$ that have a non-empty intersection of their interior with the surface $\mathbb{S}$ is denoted by $\mathcal{G}^*$, i.e. $\mathcal{G}^* := \{H \in \mathcal{G};\ \operatorname{int}(H) \cap \mathbb{S} \neq \emptyset\}$. The condition

$$\operatorname{int}(H) \cap \mathbb{S} \neq \emptyset \tag{2}$$

for $H = [i, i+k] \times [j, j+l] \times [z, z+v] \in \mathcal{H}$ with $H \subset \hat{H}$ arising in this definition can be checked very easily as the following statements show.

Lemma 1. *Let $(\check{r}, \check{s}) \in \{i, \dots, i+k-1\} \times \{j, \dots, j+l-1\}$ and $\star \in \{+, -\}$ and $(x, y, w) \in S_{\check{r}\check{s}}^{\star}$ with $z < w < z+v$. Then, (2) holds.*

Proof. There exists $\epsilon > 0$ with $\emptyset \neq \operatorname{relint}(B_\epsilon(x, y, w) \cap H) \subset \operatorname{int}(H)$. We set $M := \operatorname{relint}(B_\epsilon(x, y, w) \cap S_{\check{r}\check{s}}^{\star}) \neq \emptyset$ and find $M \subset \mathbb{S}$ and $M \subset \operatorname{relint}(B_\epsilon(x, y, w) \cap H) \subset \operatorname{int}(H)$. Thus, $M \subset \operatorname{int}(H) \cap \mathbb{S}$, which gives (2). □

Theorem 2. *(2) holds if and only if there exists* $(r,s) \in \{i,\dots,i+k\}\times\{j,\dots,j+l\}$ *with* $z < A_{rs} < z+v$ *or there exist* $(r,s),(\bar r,\bar s) \in \{i,\dots,i+k\}\times\{j,\dots,j+l\}$ *such that* $A_{rs} \le z$ *and* $z+v \le A_{\bar r\bar s}$.

Proof. Let $(r,s) \in \{i,\dots,i+k\}\times\{j,\dots,j+l\}$ with $z < A_{rs} < z+v$. There exists $(\check r,\check s) \in \{i,\dots,i+k-1\}\times\{j,\dots,j+l-1\}$ and $\star \in \{+,-\}$ such that $(r,s,A_{rs}) \in S^{\star}_{\check r\check s}$. Thus, we get (2) from Lemma 1 with $(x,y,w) := (r,s,A_{rs})$. Let $(r,s),(\bar r,\bar s) \in \{i,\dots,i+k\}\times\{j,\dots,j+l\}$ such that $A_{rs} \le z$ and $z+v \le A_{\bar r\bar s}$. There exists $(\check r,\check s) \in \{i,\dots,i+k-1\}\times\{j,\dots,j+l-1\}$ and $\star \in \{+,-\}$ such that $e := \mathrm{conv}((r,s,A_{rs}),(\bar r,\bar s,A_{\bar r\bar s})) \subset S^{\star}_{\check r\check s}$. We have $e \cap \mathrm{int}(H) \neq \emptyset$ as $v > 0$. Thus, we obtain (2) from Lemma 1 with some $(x,y,w) \in e \cap \mathrm{int}(H)$.

Let (2) be true. Then, there exist $(\check r,\check s) \in \{i,\dots,i+k-1\}\times\{j,\dots,j+l-1\}$ and $\star \in \{+,-\}$ such that $\mathrm{int}(H) \cap S^{\star}_{\check r\check s} \neq \emptyset$. Setting

$$I^{\star} := \{(\check r,\check s),(\check r+1,\check s+1)\} \cup \begin{cases} (\check r,\check s+1), & \star = +,\\ (\check r+1,\check s), & \star = - \end{cases}$$

we conclude the existence of $(r,s) \in I^{\star}$ with $z < A_{rs} < z+v$ or the existence of $(r,s),(\bar r,\bar s) \in I^{\star}$ with $A_{rs} \le z$ and $z+v \le A_{\bar r\bar s}$. □

The set of all rectangular hexahedrons of a grid $\mathcal{G}$ of $\hat H \in \mathcal{H}$ that are above $\mathbb{S}$ is denoted by $\mathcal{G}^+$ and the set of those that are below $\mathbb{S}$ is denoted by $\mathcal{G}^-$. Here, we say that $M \subset \mathbb{R}^3$ is below $\tilde M \subset \mathbb{R}^3$ if $\sup \mathbb{L}_M(x,y) \le \inf \mathbb{L}_{\tilde M}(x,y)$ for all $(x,y) \in \mathbb{R}^2$ with $\mathbb{L}_M(x,y) := \{z \in \mathbb{R};\ (x,y,z) \in M\} \neq \emptyset$ and $\mathbb{L}_{\tilde M}(x,y) \neq \emptyset$. Accordingly, M is said to be above $\tilde M$ if $\tilde M$ is below M. In the same way we define 'strictly below' and 'strictly above', but with the $<$-sign instead of the $\le$-sign. Note that $\inf \mathbb{L}_{\mathbb{S}}(x,y) = \sup \mathbb{L}_{\mathbb{S}}(x,y) = \mathfrak{s}(x,y)$ for all $(x,y) \in [1,n]\times[1,m]$. Thus, $H = [i,i+k]\times[j,j+l]\times[z,z+v] \in \mathcal{G}^{\pm}$ if and only if $H \in \mathcal{G}$ and

$$\pm\,\mathfrak{s}(x,y) \le \pm(z+v_{\pm}) \tag{3}$$

for all $(x,y) \in [i,i+k]\times[j,j+l]$, where $v_+ := 0$ and $v_- := v$. In particular, if $H \in \mathcal{G}^{\pm}$, it holds

$$\pm\,A_{ij} \le \pm(z+v_{\pm}). \tag{4}$$

Theorem 3. *Let* $\mathcal{G}$ *be a grid of* $\hat H \in \mathcal{H}$. *Then,* $\mathcal{G} = \mathcal{G}^* \dot\cup \mathcal{G}^- \dot\cup \mathcal{G}^+$.

Proof. Let $H = [i,i+k]\times[j,j+l]\times[z,z+v] \in \mathcal{G}$. Assuming $H \in \mathcal{G}^- \cap \mathcal{G}^+$. Then, (4) implies $z+v \le A_{ij} \le z$, which is a contradiction to $v > 0$. Thus, $\mathcal{G}^- \cap \mathcal{G}^+ = \emptyset$. If $H \in \mathcal{G}^*$, then there exists $(x,y) \in (i,i+k)\times(j,j+l)$ with $z < \mathfrak{s}(x,y) < z+v$ or, equivalently, $\pm(z+v_{\pm}) < \pm\mathfrak{s}(x,y)$, which contradicts (3). Thus, $\mathcal{G}^* \cap \mathcal{G}^{\pm} = \emptyset$. If $H \notin \mathcal{G}^*$, then $z+v \le \mathfrak{s}(x,y)$ or $\mathfrak{s}(x,y) \le z$ for all $(x,y) \in (i,i+k)\times(j,j+l)$. As $\mathfrak{s}$ is continuous, we have $H \in \mathcal{G}^-$ or $H \in \mathcal{G}^+$, which completes the proof. □

The sets $\mathcal{G}^-$ and $\mathcal{G}^+$ can easily be determined (provided that $\mathcal{G}^*$ is available). This is stated in the following theorem.

Theorem 4. *Let* $\mathcal{G}$ *be a grid of* $\hat H \in \mathcal{H}$. *Then,*

$$\mathcal{G}^{\pm} = \left\{ [i,i+k]\times[j,j+l]\times[z,z+v] \in \mathcal{G}\backslash\mathcal{G}^*;\ \mp(z+v_{\mp}) < \mp A_{ij} \right\}.$$

Proof. Let $H = [i,i+k]\times[j,j+l]\times[z,z+v] \in \mathcal{G}$. If $H \in \mathcal{G}^{\pm}$, then we have $H \notin \mathcal{G}^*$, see Theorem 3. Moreover, (3) implies $z+v > z \ge \mathfrak{s}(i,j) = A_{ij}$ in the case $H \in \mathcal{G}^+$ and $z < z+v \le \mathfrak{s}(i,j) = A_{ij}$ in the case $H \in \mathcal{G}^-$, i.e. $\mp(z+v_{\mp}) < \mp A_{ij}$.

Let $H \notin \mathcal{G}^*$ and $\mp(z+v_{\mp}) < \mp A_{ij}$. We conclude from Theorem 3 that $H \in \mathcal{G}^+$ or $H \in \mathcal{G}^-$. Assuming $H \in \mathcal{G}^{\mp}$, then (4) would imply $\mp(z+v_{\mp}) < \mp A_{ij} \le \mp(z+v_{\mp})$, which is obviously a contradiction. Thus, $H \in \mathcal{G}^{\pm}$. □

We say that a grid $\mathcal{G}$ of $\hat H \in \mathcal{H}$ is a DEM grid (with respect to A) if for all $H = [i,i+k]\times[j,j+l]\times[z,z+v] \in \mathcal{G}^*$ it holds $k = l = 1$, i.e. all rectangular hexahedrons in $\mathcal{G}$ that intersect the surface $\mathbb{S}$ have x- and y-length 1. Trivially, a grid of $\hat H$ of which all rectangular hexahedrons have x- and y-length 1 is a DEM grid. A simple (octree-based) algorithm for generating a DEM grid $\mathcal{G}$ of $\hat H$ (as well as the sets $\mathcal{G}^*$, $\mathcal{G}^-$ and $\mathcal{G}^+$) is given by Algorithm 1. For this purpose, we specify the (possibly asymmetric) decomposition

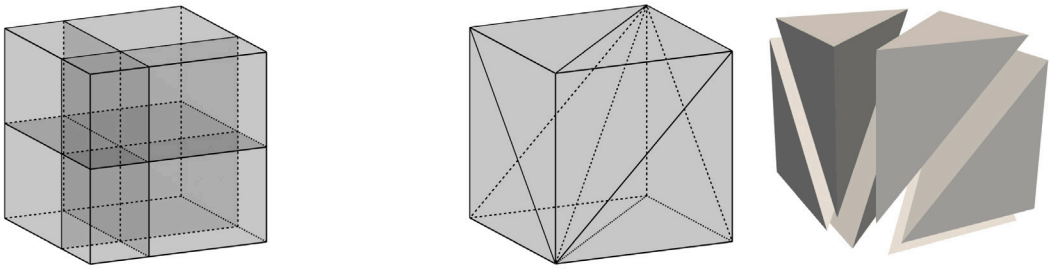
$$\mathcal{R}(H) := \{R_i(H);\ \mathrm{int}(R_i(H)) \neq \emptyset,\ 1 \le i \le 8\} \tag{5}$$

of a rectangular hexahedron $H := [i,i+k]\times[j,j+l]\times[z,z+v] \in \mathcal{H}$ by

$$\begin{aligned}
R_1(H) &:= [i,i+\lfloor k/2\rfloor]\times[j,j+\lfloor l/2\rfloor]\times[z,z+v/2],\\
R_2(H) &:= [i+\lfloor k/2\rfloor,i+k]\times[j,j+\lfloor l/2\rfloor]\times[z,z+v/2],\\
R_3(H) &:= [i,i+\lfloor k/2\rfloor]\times[j+\lfloor l/2\rfloor,j+l]\times[z,z+v/2],\\
R_4(H) &:= [i+\lfloor k/2\rfloor,i+k]\times[j+\lfloor l/2\rfloor,j+l]\times[z,z+v/2],\\
R_5(H) &:= [i,i+\lfloor k/2\rfloor]\times[j,j+\lfloor l/2\rfloor]\times[z+v/2,z+v],\\
R_6(H) &:= [i+\lfloor k/2\rfloor,i+k]\times[j,j+\lfloor l/2\rfloor]\times[z+v/2,z+v],\\
R_7(H) &:= [i,i+\lfloor k/2\rfloor]\times[j+\lfloor l/2\rfloor,j+l]\times[z+v/2,z+v],\\
R_8(H) &:= [i+\lfloor k/2\rfloor,i+k]\times[j+\lfloor l/2\rfloor,j+l]\times[z+v/2,z+v],
\end{aligned}$$

i.e. $\mathcal{R}(H)$ is given by halving the lengths of H and keeping the x- and y-coordinates of the vertices at integer values by rounding down. The latter, in particular, ensures

$$\mathcal{R}(H) \subset \mathcal{H}. \tag{6}$$



(a) Asymmetric decomposition $\mathcal{R}(H)$.

(b) Rectangular hexahedron H divided into the set of six tetrahedrons $\mathcal{T}(H)$.

Fig. 2. Visualization of decompositions.

Note that the condition $\operatorname{int}(R_i(H)) \neq \emptyset$ in (5) implies that only rectangular hexahedrons with a positive volume are contained in $\mathcal{R}(H)$. Moreover, $\mathcal{R}(H)$ can be a bisection or a quartering of H, which occur if k or l is equal to 1.

Fig. 2(a) shows the decomposition $\mathcal{R}(H)$ of $H := [1,4]^3$ with rounded x- and y-lengths. Fig. 1(c) shows a DEM grid of $\hat{H} := [1,50]^3$ with respect to the matrix A defined in (1), which results from the application of Algorithm 1.

Theorem 5. *Let $(\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2) := \text{GenerateDEMGrid}(\hat{H})$ for $\hat{H} \in \mathcal{H}$. Then, $\mathcal{G} := \mathcal{G}_0 \cup \mathcal{G}_1 \cup \mathcal{G}_2$ is a DEM grid of $\hat{H}$ and $\mathcal{G}^* = \mathcal{G}_0$, $\mathcal{G}^- = \mathcal{G}_1$ and $\mathcal{G}^+ = \mathcal{G}_2$. Furthermore, Algorithm 1 needs at most*

$$\max\left\{\lceil \log_2(n) \rceil, \lceil \log_2(m) \rceil\right\}$$

runs of its repeat-until-loop.

Proof. By definition of the decomposition $\mathcal{R}(H)$ of a rectangular hexahedron $H \in \mathcal{H}$, it holds $\operatorname{relint}(R) \cap \operatorname{relint}(\tilde{R}) = \emptyset$ for $R, \tilde{R} \in \mathcal{R}(H)$ with $R \neq \tilde{R}$ and

$$H = \bigcup \mathcal{R}(H). \tag{7}$$

This together with (6) means that replacing H by $\mathcal{R}(H)$ preserves the property of $\mathcal{G} \cup \mathcal{D}$ in Algorithm 1 to be a grid of $\hat{H}$ and, thus, the returned set $\mathcal{G}$ is indeed a grid of $\hat{H}$. Moreover, $\mathcal{G}$ is a DEM grid as the algorithm specifically refines rectangular hexahedrons with a non-empty intersection with the DEM surface $\mathbb{S}$ until their x- and y-lengths both reach 1. Thus, the identities $\mathcal{G}^* = \mathcal{G}_0$, $\mathcal{G}^+ = \mathcal{G}_1$ and $\mathcal{G}^- = \mathcal{G}_2$ follow from the definition of a DEM grid and Theorem 4.

In the worst case, the x- and y-lengths of $\hat{H}$ are n and m, respectively, and all rectangular hexahedrons generated by Algorithm 1 have a non-empty intersection with the surface $\mathbb{S}$. Hence, the algorithm terminates if all rectangular hexahedrons have x- and y-length 1. Without loss of generality, let $n \geq m$, i.e. it takes longer for the x-length to reach 1 than for the y-length. Since the x-length is halved after every run it reaches 1 after a maximum of $\lceil \log_2(n) \rceil$ runs and no rectangular hexahedron will be further refined, i.e. the algorithm terminates after at most $\lceil \log_2(n) \rceil$ runs. □

4. DEM-fitted tetrahedron decomposition

Let $H = [i, i+1] \times [j, j+1] \times [z, z+v] \in \mathcal{G}^*$ be a rectangular hexahedron of a DEM grid $\mathcal{G}$ of $\hat{H} \in \mathcal{H}$. In this section, we construct a decomposition $\mathcal{Z}^+(H) \cup \mathcal{Z}^-(H)$ of H into tetrahedrons which is fitted to the DEM represented by the matrix A, i.e. each tetrahedron of $\mathcal{Z}^+(H)$ is above and each of $\mathcal{Z}^-(H)$ is below $\mathbb{S}$. We start with defining the decomposition $\mathcal{T}(H)$ of H into 6 tetrahedrons by

$$\mathcal{T}(H) := \{T_1^+(H), T_2^+(H), T_3^+(H), T_1^-(H), T_2^-(H), T_3^-(H)\}$$

and

$$\begin{aligned}
T_1^+(H) &:= \operatorname{conv}((i,j,z),(i,j,z+v),(i+k,j,z+v),(i+k,j+l,z+v)),\\
T_2^+(H) &:= \operatorname{conv}((i,j,z),(i+k,j,z),(i+k,j,z+v),(i+k,j+l,z+v)),\\
T_3^+(H) &:= \operatorname{conv}((i,j,z),(i+k,j,z),(i+k,j+l,z),(i+k,j+l,z+v)),\\
T_1^-(H) &:= \operatorname{conv}((i,j,z),(i,j,z+v),(i,j+l,z+v),(i+k,j+l,z+v)),\\
T_2^-(H) &:= \operatorname{conv}((i,j,z),(i,j+l,z),(i,j+l,z+v),(i+k,j+l,z+v)),\\
T_3^-(H) &:= \operatorname{conv}((i,j,z),(i,j+l,z),(i+k,j+l,z),(i+k,j+l,z+v)).
\end{aligned}$$

We refer to Fig. 2(b), where this decomposition is shown. Note that as this specific decomposition is constructed along the diagonal from (i,j) to $(i+1,j+1)$ it holds $\operatorname{relint}(S_{ij}^{\pm}) \cap \operatorname{int}(T_{\ell}^{\mp}(H)) = \emptyset$ for all $\ell \in \{1,2,3\}$. As a consequence there holds

$$T_{\ell}^{\pm}(H) \cap \mathbb{S} = T_{\ell}^{\pm}(H) \cap S_{ij}^{\pm}. \tag{8}$$

In the following we denote the vertices of a tetrahedron T by $V_1(T), \dots, V_4(T) \in \mathbb{R}^3$, i.e.

$$T = \operatorname{conv}(V_1(T), V_2(T), V_3(T), V_4(T)).$$

The set of its edges is denoted by $\mathcal{E}(T)$, i.e. $\mathcal{E}(T) := \{e_1(T), \dots, e_6(T)\}$ with

$$\begin{aligned} e_1(T) &:= \operatorname{conv}\{V_1(T), V_2(T)\}, & e_2(T) &:= \operatorname{conv}\{V_2(T), V_3(T)\}, \\ e_3(T) &:= \operatorname{conv}\{V_3(T), V_1(T)\}, & e_4(T) &:= \operatorname{conv}\{V_1(T), V_4(T)\}, \\ e_5(T) &:= \operatorname{conv}\{V_2(T), V_4(T)\}, & e_6(T) &:= \operatorname{conv}\{V_3(T), V_4(T)\}. \end{aligned}$$

Furthermore, we denote the unique vertical edge of T by $e'(T) \in \mathcal{E}(T)$, i.e. $e'(T) = \{(r(T), s(T))\} \times [z, z+v]$ with $r(T) \in \{i, i+1\}$ and $s(T) \in \{j, j+1\}$. With these preparations, we make the following statements.

Lemma 6. *Let $e \in \mathcal{E}(T_{\ell}^{\pm}(H))$ and $\ell \in \{1,2,3\}$. Then,*

$$\operatorname{relint}(e) \cap \mathbb{S} = \operatorname{relint}(e) \cap S_{ij}^{\pm} = \operatorname{relint}(e) \cap \bigcup \mathcal{E}_{ij}^{\pm}.$$

Proof. It holds $\operatorname{relint}(S_{ij}^{\pm}) \cap K = \emptyset$ for all $K \in \mathcal{K}_{ij}^{\pm}$, which implies $e \cap \operatorname{relint}(S_{ij}^{\pm}) = \emptyset$ and, thus,

$$\operatorname{relint}(e) \cap S_{ij}^{\pm} = \operatorname{relint}(e) \cap \bigcup \mathcal{E}_{ij}^{\pm}.$$

Thus, we conclude from (8) that

$$\operatorname{relint}(e) \cap \mathbb{S} = \operatorname{relint}(e) \cap (T_{\ell}^{\pm}(H) \cap \mathbb{S}) = \operatorname{relint}(e) \cap (T_{\ell}^{\pm}(H) \cap S_{ij}^{\pm}) = \operatorname{relint}(e) \cap S_{ij}^{\pm} = \operatorname{relint}(e) \cap \bigcup \mathcal{E}_{ij}^{\pm}. \quad \square$$

Theorem 7. *Let $e \in \mathcal{E}(T_{\ell}^{\pm}(H))$ and $\ell \in \{1,2,3\}$. Then, $\operatorname{relint}(e) \cap \mathbb{S}$ is either empty or $\operatorname{relint}(e)$ or consists of a single point.*

Proof. We obtain from Lemma 6 that $\operatorname{relint}(e) \cap S_{ij}^{\pm}$ is either empty, a line segment or consists of a single point. Let $\operatorname{relint}(e) \cap \mathbb{S}$ be a line segment. We have $\operatorname{relint}(e) \subset \operatorname{relint}(K)$ for some $K \in \mathcal{K}_{ij}^{\star}$ and $\star \in \{+,-\}$ which implies $e = \tilde{e}$ for $\tilde{e} \in \mathcal{E}_{ij}^{\star}$ with $\tilde{e} \subset K$ as both edges e and $\tilde{e}$ cross the stripe K from one side to the other. Consequently, $\operatorname{relint}(e) \cap \mathbb{S} = \operatorname{relint}(e)$. $\square$

Theorem 8. *Let $T \in \mathcal{T}(H)$. Then, $\operatorname{int}(T) \cap \mathbb{S} \neq \emptyset$ if and only if there exists $e \in \mathcal{E}(T)$ such that $\operatorname{relint}(e) \cap \mathbb{S}$ is a single point.*

Proof. Let $\ell \in \{1,2,3\}$ and $\star \in \{+,-\}$ with $T = T_{\ell}^{\star}(H)$ and set $S := S_{ij}^{\star}$. If $\operatorname{relint}(e) \cap \mathbb{S}$ consist of a single point $P \in \mathbb{R}^3$ for an $e \in \mathcal{E}(T)$, there exists $\epsilon > 0$ such that $B_{\epsilon}(P) \cap e \subset \operatorname{relint}(e)$. We conclude from Lemma 6 that $\{P\} = \operatorname{relint}(e) \cap S$ and, thus, $B_{\epsilon}(P) \cap \operatorname{relint}(S) \neq \emptyset$. Taking the location of S and T to each other into account we have $B_{\epsilon}(P) \cap \operatorname{relint}(S) \subset B_{\epsilon}(P) \cap \operatorname{int}(T)$. Using (8) we obtain

$$\operatorname{int}(T) \cap \mathbb{S} = \operatorname{int}(T) \cap \operatorname{relint}(S) \supset B_{\epsilon}(P) \cap \operatorname{relint}(S) \neq \emptyset.$$

Finally, let $\operatorname{int}(T) \cap \mathbb{S} \neq \emptyset$. From (8) we obtain

$$\operatorname{int}(T) \cap S \neq \emptyset. \tag{9}$$

Let $e := e'(T)$ and let $Y = (Y_1, Y_2, Y_3) \in \mathcal{V} := \{V_{ij}, V_{ij}^{\star}, \hat{V}_{ij}\}$ be the vertex of S on the straight line containing e, i.e. $\mathcal{V} \cap \{(r(T), s(T))\} \times \mathbb{R} = \{Y\}$. In the case that $Y \in \operatorname{relint}(e)$ (which means $Y_3 \in (z, z+v)$) we immediately get from (6) that $\operatorname{relint}(e) \cap \mathbb{S} = \{Y\}$. Assume $Y \notin \operatorname{relint}(e)$ and let $W^{\min} = (W_1^{\min}, W_2^{\min}, W_3^{\min}) \in \mathcal{V}$ and $W^{\max} = (W_1^{\max}, W_2^{\max}, W_3^{\max}) \in \mathcal{V}$ with

$$W_3^{\min} = \min_{(U_1,U_2,U_3) \in \mathcal{V}} U_3, \quad W_3^{\max} = \max_{(U_1,U_2,U_3) \in \mathcal{V}} U_3. \tag{10}$$

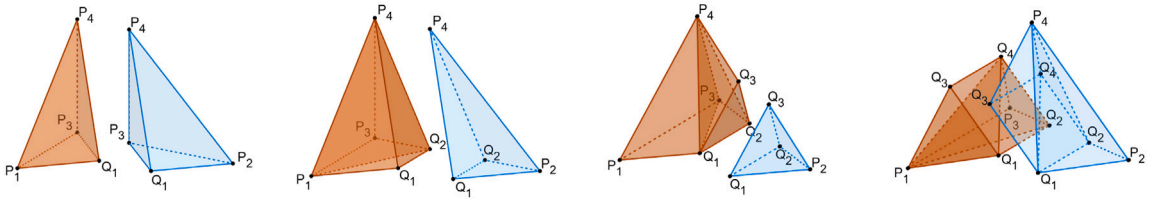
Moreover, let $e_1^W, e_2^W \in \mathcal{E}(T)$ for $W \in \{W^{\min}, W^{\max}\}$ be those two edges of T so that $e, e_1^W, e_2^W \subset K \in \mathcal{K}_{ij}^{\star}$, $W \in K$ and e_1^W is strictly below e_2^W. In the case that $Y_3 \leq z$ we observe that $W^{\max}$ is strictly above $\hat{e} := e_2^{W^{\max}}$ as otherwise (10) would imply that S is a facet of T or strictly below T which contradicts (9). We see that Y is strictly below $\hat{e}$ and that $\hat{e}, s \subset K$ with $s := \operatorname{conv}(Y, W^{\max})$. Furthermore, $\hat{e}$ and s cross the stripe K from one side to the other. Thus, $\operatorname{relint}(\hat{e}) \cap s$ consists of a single point. From Lemma 6 we conclude that $\operatorname{relint}(\hat{e}) \cap \mathbb{S}$ consists of that point. In the case that $Y_3 \geq z+v$ we show that $\operatorname{relint}(\hat{e}) \cap \mathbb{S}$ consists of a single point in the same way by defining $\hat{e} := e_1^{W^{\min}}$ and exchanging 'strictly above' by 'strictly below' and $W^{\max}$ by $W^{\min}$. $\square$

The next step in the construction of the decomposition is to introduce the intersection indicator $\mathrm{i}_H : \mathcal{T}(H) \to \mathcal{J}$ which is defined as

$$\mathrm{i}_H(T)_r := \begin{cases} 1, & \operatorname{relint}(e_r(T)) \cap \mathbb{S} \text{ is a single point}, \\ 0, & \text{else}, \end{cases}$$

Table 1
Intersection indicator $\mathfrak{i}_H$, case index $\mathfrak{c}$ and permutation mapping $\mathfrak{p}$.

$\mathfrak{i}_H$	$\mathfrak{c}$	$\mathfrak{p}$	$\mathfrak{i}_H$	$\mathfrak{c}$	$\mathfrak{p}$
(0, 0, 0, 0, 0, 0)	0	(1, 2, 3, 4)	(1, 0, 0, 0, 0, 0)	1	(1, 2, 3, 4)
(0, 0, 1, 0, 0, 0)	1	(3, 1, 2, 4)	(0, 0, 0, 1, 0, 0)	1	(1, 4, 3, 2)
(0, 0, 0, 0, 1, 0)	1	(2, 4, 3, 1)	(0, 0, 0, 0, 0, 1)	1	(4, 3, 1, 2)
(1, 1, 0, 0, 0, 0)	2	(1, 2, 3, 4)	(1, 0, 1, 0, 0, 0)	2	(3, 1, 2, 4)
(1, 0, 0, 1, 0, 0)	2	(4, 1, 2, 3)	(1, 0, 0, 0, 1, 0)	2	(1, 2, 4, 3)
(0, 1, 1, 0, 0, 0)	2	(2, 3, 1, 4)	(0, 1, 0, 0, 1, 0)	2	(4, 2, 3, 1)
(0, 1, 0, 0, 0, 1)	2	(2, 3, 4, 1)	(0, 0, 1, 1, 0, 0)	2	(3, 1, 4, 2)
(0, 0, 1, 0, 0, 1)	2	(1, 3, 4, 2)	(0, 0, 0, 1, 1, 0)	2	(2.4, 1, 3)
(0, 0, 0, 1, 0, 1)	2	(1, 4, 3, 2)	(0, 0, 0, 0, 1, 1)	2	(3, 4, 2, 1)
(1, 1, 0, 0, 1, 0)	3	(1, 2, 3, 4)	(1, 0, 1, 1, 0, 0)	3	(3, 1, 2, 4)
(0, 1, 1, 0, 0, 1)	3	(2, 3, 1, 4)	(0, 0, 0, 1, 1, 1)	3	(2, 4, 3, 1)
(1, 1, 0, 1, 0, 1)	4	(1, 2, 3, 4)	(1, 0, 1, 0, 1, 1)	4	(3, 1, 2, 4)
(0, 1, 1, 1, 1, 0)	4	(2, 3, 1, 4)			



(a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4

Fig. 3. Example of refinement of a tetrahedron T for the case index $\mathfrak{c} \in \{1, \dots, 4\}$ into the tetrahedrons $\mathcal{Z}_{\mathfrak{c}}^{f}(T)$ (orange) and $\mathcal{Z}_{\mathfrak{c}}^{-f}(T)$ (blue).

where $\mathcal{J} \subset \{0,1\}^6$ is given in Table 1. The mapping $\mathfrak{i}_H$ is well-defined (and surjective) as $\mathcal{J}$ reflects all possible intersections of the edges of $T \in \mathcal{T}(H)$ with $\mathbb{S}$ resulting in a single point. Theorem 7 implies $\mathfrak{i}_H(T)_r = 0$ if and only if $\operatorname{relint}(e_r(T)) \cap \mathbb{S}$ is empty or $\operatorname{relint}(e_r(T))$. From Theorem 8 we conclude that $\mathfrak{i}_H(T) \neq (0,0,0,0,0,0)$ if and only if $\operatorname{int}(T) \cap \mathbb{S} \neq \emptyset$. Note $\mathfrak{i}_H(T)$ can easily be computed since $\operatorname{relint}(e) \cap \mathbb{S}$ with $e \in \mathcal{E}(T)$ coincides with the intersection of one edge in $\mathcal{E}_{ij}^{\star}$ with $T = T_{\ell}^{\star}$ for some $\ell \in \{1,2,3\}$ and $\star \in \{+,-\}$ (see Lemma 6).

Next, we introduce the case index $\mathfrak{c} : \mathcal{J} \to \{0, \dots, 4\}$ and the permutation mapping $\mathfrak{p} : \mathcal{J} \to \{1, \dots, 4\}^4$ by using Table 1. We refer to [40] for more details on describing decompositions of a hexahedron with these mappings. Herewith, we define $\mathfrak{f}_H : \mathcal{T}(H) \to \{+,-\}$ as

$$\mathfrak{f}_H(T) := \begin{cases} +, & V_{\mathfrak{p}(\mathfrak{i}_H(T))_1}(T) \text{ is strictly above } \mathbb{S}, \\ -, & \text{else.} \end{cases}$$

Note that $\mathfrak{f}_H(T) = +$ if and only if $A_{rs} < w$ with $V_{\mathfrak{p}(\mathfrak{i}_H(T))_1}(T) = (r, s, w)$.

Now, we define $\mathcal{Z}^{\pm}(H)$ in the following way: For $T \in \mathcal{T}(H)$ set $P_i := V_{p_i}(T)$ with $i \in \{1, \dots, 4\}$ and $p := \mathfrak{p}(\mathfrak{i}_H(T))$. Furthermore, let $r_1 < \dots < r_k$ uniquely fulfill $\mathfrak{i}_H(T)_{r_s} = 1$ and let Q_s be the single point in $\operatorname{relint}(e_{r_s}(T)) \cap \mathbb{S}$ where $1 \le s \le k$ and $k := \sum_{i=1}^{6} \mathfrak{i}_H(T)_i$. Setting $f := \mathfrak{f}_H(T)$ we define

$$\begin{aligned}
\mathcal{Z}_1^{f}(T) &:= \{\operatorname{conv}(P_1, Q_1, P_3, P_4)\}, \\
\mathcal{Z}_2^{f}(T) &:= \{\operatorname{conv}(P_1, Q_1, Q_2, P_4), \operatorname{conv}(P_1, Q_2, P_3, P_4)\}, \\
\mathcal{Z}_3^{f}(T) &:= \{\operatorname{conv}(Q_1, P_1, P_3, P_4), \operatorname{conv}(P_3, Q_1, Q_2, P_4), \operatorname{conv}(Q_1, Q_2, Q_3, P_4)\}, \\
\mathcal{Z}_4^{f}(T) &:= \{\operatorname{conv}(P_1, Q_4, Q_3, Q_1), \operatorname{conv}(Q_1, Q_2, P_1, Q_4), \operatorname{conv}(P_3, P_1, Q_2, Q_4)\}, \\
\mathcal{Z}_4^{-f}(T) &:= \{\operatorname{conv}(Q_1, Q_3, Q_4, P_4), \operatorname{conv}(Q_1, Q_2, Q_4, P_4), \operatorname{conv}(Q_1, P_4, Q_2, P_2)\}
\end{aligned}$$

$$\begin{aligned}
\mathcal{Z}_1^{-f}(T) &:= \{\operatorname{conv}(Q_1, P_2, P_3, P_4)\}, \\
\mathcal{Z}_2^{-f}(T) &:= \{\operatorname{conv}(Q_1, P_2, Q_2, P_4)\}, \\
\mathcal{Z}_3^{-f}(T) &:= \{\operatorname{conv}(Q_1, Q_2, Q_3, P_2)\},
\end{aligned}$$

and

$$\mathcal{Z}_0^{\pm}(T) := \begin{cases} \{T\}, & T \text{ is above/below } \mathbb{S}, \\ \emptyset, & \text{else,} \end{cases}$$

where $+$ is assigned to 'above' and $-$ to 'below'. Eventually, we specify

$$\mathcal{Z}^{\pm}(H) := \bigcup_{T \in \mathcal{T}(H)} \mathcal{Z}_{\mathfrak{c}(\mathfrak{i}_H(T))}^{\pm}(T).$$

Note in the case $\mathfrak{c}(\mathfrak{i}_H(T)) = 0$ we have that $\mathcal{Z}_0^{\pm}(T) = \{T\}$ if and only if $\pm(z + v_{\pm}) < \pm A_{r(T),s(T)}$. In view of Fig. 3 it is clear that $\mathcal{Z}^{+}(H) \cup \mathcal{Z}^{-}(H)$ is indeed a decomposition of H into tetrahedrons.

Algorithm 2 Assembling of K and L

function ASSEMBLE($\mathcal{F} \subset \hat{\mathcal{G}}$)
  $K \leftarrow 0 \in \mathbb{R}^{d\times d}$
  $L \leftarrow 0 \in \mathbb{R}^{d}$
  for all $\hat{H} \in \mathcal{F}$ **do**
    $(\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2) \leftarrow$ GenerateDEMGrid($\hat{H}$)
    $\hat{K} \leftarrow K_{\hat{H}}(\mathcal{G}_1) + \gamma K_{\hat{H}}(\mathcal{G}_2)$
    $\hat{L} \leftarrow L_{\hat{H}}(\mathcal{G}_1)$
    $\hat{G} \leftarrow 0 \in \mathbb{R}^{24}$
    for all $H \in \mathcal{G}_0$ **do**
      $(\mathcal{Z}_1, \mathcal{Z}_2) \leftarrow (\mathcal{Z}^-(H), \mathcal{Z}^+(H))$
      $\hat{K} \leftarrow \hat{K} + K_{\hat{H}}(\mathcal{Z}_1) + \gamma K_{\hat{H}}(\mathcal{Z}_2)$
      $\hat{L} \leftarrow \hat{L} + L_{\hat{H}}(\mathcal{Z}_1)$
      $\hat{G} \leftarrow \hat{G} + G_{\hat{H}}(H)$
    end for
    $w \leftarrow w_{\hat{H}}$
    $\hat{K} \leftarrow w K_{\hat{H}}$
    $\hat{L} \leftarrow w L_{\hat{H}} + \hat{G}$
    $c \leftarrow c_{\hat{H}}$
    for $\kappa = 1, \dots, 24$ **do**
      $L_{c_\kappa} \leftarrow L_{c_\kappa} + \hat{L}_\kappa$
      for $\ell = 1, \dots, 24$ **do**
        $K_{c_\kappa, c_\ell} \leftarrow K_{c_\kappa, c_\ell} + \hat{K}_{\kappa\ell}$
      end for
    end for
  end for
  return (K, L)
end function

Theorem 9. *Let $Z^\pm \in \mathcal{Z}^\pm(H)$. Then, $Z^\pm$ is above/below $\mathbb{S}$.*

Proof. Let $Z^\pm \subset T$ for some $T \in \mathcal{T}(H)$. In the case that $\mathfrak{c}(\mathfrak{i}_H(T)) = 0$ the assertion trivially holds. Let $\mathfrak{c}(\mathfrak{i}_H(T)) \geq 1$ and define

$$\mathbb{Z}^\pm := \bigcup \mathcal{Z}^\pm_{\mathfrak{c}(\mathfrak{i}_H(T))}(T).$$

The vertex $P_1 := V_{\mathfrak{p}(\mathfrak{i}_H(T))_1}$ is strictly above or strictly below $\mathbb{S}$ because P_1 does not lay on $\mathbb{S}$, see Fig. 3. Hence, $f := \mathfrak{f}_H(T) = \pm$ if and only if P_1 is strictly above/below $\mathbb{S}$. Since $P_1 \in \mathbb{Z}^f$ and $\mathrm{int}(\mathbb{Z}^f) \cap \mathbb{S} = \emptyset$ we conclude that $\mathbb{Z}^f$ is above/below $\mathbb{S}$ if and only if $f = \pm$. As $\mathbb{S}$ separates $\mathbb{Z}^f$ from $\mathbb{Z}^{-f}$ (see Fig. 3), $\mathbb{Z}^{-f}$ is below/above $\mathbb{S}$ if and only if $f = \pm$. Taking all these cases into account we conclude that $\mathbb{Z}^\pm$ is above/below $\mathbb{S}$. From $Z^\pm \subset \mathbb{Z}^\pm$ we get the assertion. □

Remark 1. The sets $\mathcal{Z}^\pm_2(T)$, $\mathcal{Z}^\pm_3(T)$ and $\mathcal{Z}^\pm_4(T)$ with $T \in \mathcal{T}(H)$ may be optimized with respect to the aspect ratios of the tetrahedrons (or other geometrical features). Here, we use fixed configurations of tetrahedrons in each decomposition case given by the case index $\mathfrak{c}$, which may result in possibly unfavorable aspect ratios. Alternatively, one may use a Delaunay algorithm [45] for appropriate decompositions of T above or below $\mathbb{S}$.

5. DEM-FCM – A Finite Cell Method for Digital Elevation Models

Let $\hat{\mathcal{G}}$ be a regular grid of $\mathbb{B}$, i.e. there exist $1 = x_1 < \cdots < x_N = n$, $1 = y_1 < \cdots < y_M = m$ and $a = z_1 < \cdots < z_B = b$ with $x_1, \dots, x_N, y_1, \dots, y_M \in \mathbb{N}$ such that

$$\hat{\mathcal{G}} = \{[x_i, x_{i+1}] \times [y_j, y_{j+1}] \times [z_k, z_{k+1}];\ (i, j, k) \in \{1, \dots, N-1\} \times \{1, \dots, M-1\} \times \{1, \dots, B-1\}\}.$$

Furthermore, let $\mathcal{F} \subset \hat{\mathcal{G}}$ with $\hat{\mathcal{G}}^- \cup \hat{\mathcal{G}}^* \subset \mathcal{F}$ and assume a DEM grid $\mathcal{G}_{\hat{H}}$ for each $\hat{H} \in \mathcal{F}$ (with respect to A). We define $\mathcal{M}^\pm := \bigcup_{\hat{H}\in\mathcal{F}} \mathcal{M}^\pm_{\hat{H}}$ with $\mathcal{M}^\pm_{\hat{H}} := \mathcal{G}^\pm_{\hat{H}} \cup \bigcup_{H\in\mathcal{G}^*_{\hat{H}}} \mathcal{Z}^\pm(H)$ for $\hat{H} \in \mathcal{F}$. Theorem 9 implies that the meshes $\mathcal{M}^+$ and $\mathcal{M}^-$ exactly resolve the surface mesh $\mathcal{S}$, i.e. $\bigcup \mathcal{M}^\pm$ is above/below $\mathbb{S}$ and $\mathbb{S} = \partial(\bigcup \mathcal{M}^+) \cap \partial(\bigcup \mathcal{M}^-)$, where ∂M denotes the boundary of a set $M \subset \mathbb{R}^3$.

We consider a boundary value problem of linear elasticity based on Hooke's law on $\Omega := \mathrm{int}(\bigcup \mathcal{M}^-)$. To this end, let the (linearized) strain and stress tensor be defined as $\epsilon(v) := \frac{1}{2}\left(\nabla v + (\nabla v)^T\right)$ and $\sigma(v) := \lambda\,\mathrm{tr}(\epsilon(v))I + 2\mu\epsilon(v)$ for a displacement field $v \in (H^1(\Omega))^3$. Here, tr denotes the trace of a matrix, $I \in \mathbb{R}^3$ is the identity matrix and λ and μ are the Lamé constants. The boundary value problem is to find a displacement field $u = (u_1, u_2, u_3) \in (H^1(\Omega))^3$ such that

$$-\mathrm{div}(\sigma(u)) = f \text{ in } \Omega \tag{11a}$$

Algorithm 3 Constraining of K' and L'

function CONSTRAIN($K' \in \mathbb{R}^{d\times d}$, $L' \in \mathbb{R}^d$, $\bar{\mathcal{N}} \subset \mathcal{N}$, $\mathcal{D} \subset \{1,2,3\}$)
  for all $(p,r) \in \bar{\mathcal{N}} \times \mathcal{D}$ **do**
    for all $(q,s) \in \bar{\mathcal{N}} \times \mathcal{D}$ **do**
      $K'_{\iota(p,s),\iota(q,s)} \leftarrow 0$
      $K'_{\iota(q,s),\iota(p,r)} \leftarrow 0$
    end for
    $K'_{\iota(p,r),\iota(p,r)} \leftarrow 1$
    $L'_{\iota(p,r)} \leftarrow 0$
  end for
  return (K', L')
end function

Algorithm 4 Computation of the solution vector ζ

function DEM-FCM($\mathcal{F} \subset \hat{\mathcal{G}}$)
  $(K, L) \leftarrow$ Assemble($\mathcal{F}$)
  $(K', L') \leftarrow$ Constrain($K, L, \mathcal{N}_D^a, \{1,2,3\}$)
  $(K', L') \leftarrow$ Constrain($K', L', \bar{\mathcal{N}}_D, \{1,2\}$)
  Solve $K'\zeta = L'$
  return ζ
end function

$$u = 0 \text{ on } \Gamma_D^a, \tag{11b}$$

$$u_1 = u_2 = 0 \text{ on } \Gamma_D, \tag{11c}$$

$$\sigma(u)n = g \text{ on } \Gamma_N. \tag{11d}$$

The Dirichlet boundary parts are defined as $\Gamma_D^a := [1,n]\times[1,m]\times\{a\}$ (bottom of $\mathbb{B}$) and $\Gamma_D := \hat{\Gamma}_D \cap \partial\Omega$ with $\hat{\Gamma}_D := \partial\mathbb{B}\backslash(\Gamma_D^a \cup \Gamma_D^b)$ (side surfaces). The Neumann boundary part is defined as $\Gamma_N := \text{relint}(\mathbb{S})$ with its outer normal n. We refer to Fig. 10(a) where these boundary parts are illustrated for the geological application of Section 7. The volume force is given by $f \in (L^2(\Omega))^3$ and the surface force by $g \in (L^2(\Gamma_N))^3$. With

$$V := \left\{v = (v_1, v_2, v_3) \in (H^1(\Omega))^3;\ v = 0 \text{ on } \Gamma_D^a,\ v_1 = v_2 = 0 \text{ on } \Gamma_D\right\}$$

the variational formulation of (11) consists in finding a displacement field $u \in V$ such that

$$a(u,v) = \ell(v) \tag{12}$$

for all $v \in V$, where

$$a(u,v) := \int_\Omega \epsilon(v) : \sigma(u)\, d\kappa, \quad \ell(v) := \int_\Omega f\cdot v\, d\kappa + \int_{\Gamma_N} g\cdot v\, ds.$$

We conclude from the Lax–Milgram lemma that (12) has a unique solution as the bilinear form a is continuous and V-elliptic and the linear form ℓ is continuous.

To discretize (12) we apply the FCM with $\text{int}(\bigcup \mathcal{F})$ as embedding domain of Ω and a weighting factor $0 < \gamma \ll 1$ on the fictitious domain $\Omega^+ := \text{int}(\bigcup \mathcal{M}^+)$. The FEM grid is given by $\mathcal{F}$ and defines the finite element space V_h as

$$V_h := \{v_h \in W_h;\ v_{h|\Gamma_D^a} = 0,\ v_{1|\hat{\Gamma}_D} = v_{2|\hat{\Gamma}_D} = 0\} \tag{13}$$

with the unconstrained finite element space

$$W_h := \left\{ v_h \in \left(C^0\left(\bigcup \mathcal{F}\right)\right)^3;\ \forall \hat{H} \in \mathcal{F} :\ v_{h|\hat{H}} \in \mathbb{P}_1 \right\}, \tag{14}$$

where $\mathbb{P}_1$ is the space of trilinear polynomials, i.e. $\mathbb{P}_1 := \text{span}\{1, x, y, z, xy, xz, yz, xyz\}$. The FCM discretization is to find $u_h \in V_h$ such that

$$a_\gamma(u_h, v_h) = \ell(v_h) \tag{15}$$

for all $v_h \in V_h$, where

$$a_\gamma(u_h, v_h) := a(u_h, v_h) + \gamma \int_{\Omega^+} \epsilon(v_h) : \sigma(u_h)\, d\kappa.$$

Again, we obtain from the Lax–Milgram lemma that (15) has a unique solution as a_γ is continuous and V_h-elliptic. In the following we call this discretization approach DEM-FCM.

The solution of (15) is obtained by solving a linear system consisting of a stiffness matrix and a load vector (as usual in finite element methods). We start with defining the index sets

$$\mathcal{N} := \{(i,j,k) \in \hat{\mathcal{N}};\ \exists \hat{H} \in \mathcal{F} :\ (x_i, y_j, z_k) \in \hat{H}\}, \quad \mathcal{N}_D^a := \hat{\mathcal{N}}_D^a \cap \mathcal{N}, \quad \mathcal{N}_D := \hat{\mathcal{N}}_D \cap \mathcal{N}$$

with

$$\begin{aligned}
\hat{\mathcal{N}} &:= \{1,\dots,N\} \times \{1,\dots,M\} \times \{1,\dots,B\},\\
\hat{\mathcal{N}}_D^a &:= \{1,\dots,N\} \times \{1,\dots,M\} \times \{1\},\\
\hat{\mathcal{N}}_D &:= \{1,N\} \times \{1,\dots,M\} \times \{2,\dots,B\} \cup \{1,\dots,N\} \times \{1,M\} \times \{2,\dots,B\}.
\end{aligned}$$

These sets represent the nodes of $\mathcal{F}$, the nodes of $\mathcal{F}$ on Γ_D^a and the nodes of $\mathcal{F}$ on $\hat{\Gamma}_D$, respectively. Next, let $\{\eta_p\}_{p\in\mathcal{N}}$ be the uniquely determined piecewisely trilinear basis functions on $\mathcal{F}$, i.e. $\eta_{p|\hat{H}} \in \mathbb{P}_1$ for all $\hat{H} \in \mathcal{F}$ and $\eta_p(x_{q_1}, y_{q_2}, z_{q_3}) = \delta_{pq}$ for all $p = (p_1,p_2,p_3), q = (q_1,q_2,q_3) \in \mathcal{N}$ (where δ is the Kronecker delta). Herewith, we define the unconstrained stiffness matrix $K \in \mathbb{R}^{d\times d}$ and the unconstrained load vector $L \in \mathbb{R}^d$ with $d := \dim W_h = 3|\mathcal{N}|$ as

$$K_{\iota(p,r),\iota(q,s)} := a_\gamma(\eta_q e_s, \eta_p e_r), \quad L_{\iota(p,r)} := \ell(\eta_p e_r)$$

where $e_1, e_2, e_3 \in \mathbb{R}^3$ are the unit vectors and $\iota : \mathcal{N} \times \{1,2,3\} \to \{1,\dots,d\}$ is a bijective numbering. We get the constrained stiffness matrix $K' \in \mathbb{R}^{d\times d}$ and the constrained load vector $L' \in \mathbb{R}^d$ (in which the Dirichlet conditions are incorporated) by setting

$$K'_{\iota(p,r),\iota(q,s)} := \begin{cases} \delta_{pq}\delta_{rs}, & p,q \in \mathcal{N}_D^a,\\ \delta_{pq}\delta_{rs}, & p,q \in \mathcal{N}_D,\ r,s \in \{1,2\}, \quad (p,r),(q,s) \in \mathcal{N} \times \{1,2,3\},\\ K_{\iota(p,r),\iota(q,s)}, & \text{else} \end{cases}$$

$$L'_{\iota(p,r)} := \begin{cases} 0, & p \in \mathcal{N}_D^a,\\ 0, & p \in \mathcal{N}_D,\ r \in \{1,2\}, \quad (p,r) \in \mathcal{N} \times \{1,2,3\}.\\ L_{\iota(p,r)}, & \text{else} \end{cases}$$

Note that K' is symmetric and positive definite. Thus, the linear system

$$K'\zeta = L' \tag{16}$$

has a unique solution $\zeta \in \mathbb{R}^d$. The solution u_h of (15) is then given by

$$u_h = \sum_{q\in\mathcal{N},\, s\in\{1,2,3\}} \zeta_{\iota(q,s)} \eta_q e_s. \tag{17}$$

To assemble K and L we use the trilinear shape functions defined on the unit cube $[-1,1]^3$ as well as local stiffness matrices, local load vectors and connectivity matrices. The (usual) trilinear shape functions $\phi := (\phi_1,\dots,\phi_8) : [-1,1]^3 \to \mathbb{R}^8$ are defined as

$$\begin{aligned}
\phi_1(x,y,z) &:= \frac{1}{8}(1-x)(1-y)(1-z), & \phi_2(x,y,z) &:= \frac{1}{8}(1+x)(1-y)(1-z),\\
\phi_3(x,y,z) &:= \frac{1}{8}(1+x)(1+y)(1-z), & \phi_4(x,y,z) &:= \frac{1}{8}(1-x)(1+y)(1-z),\\
\phi_5(x,y,z) &:= \frac{1}{8}(1-x)(1-y)(1+z), & \phi_6(x,y,z) &:= \frac{1}{8}(1+x)(1-y)(1+z),\\
\phi_7(x,y,z) &:= \frac{1}{8}(1+x)(1+y)(1+z), & \phi_8(x,y,z) &:= \frac{1}{8}(1-x)(1+y)(1+z).
\end{aligned}$$

Let $\hat{H} = [x_i, x_{i+1}] \times [y_j, y_{j+1}] \times [z_k, z_{k+1}] \in \mathcal{F}$. We note that

$$\eta_{p|\hat{H}} = \phi_{\rho_{\hat{H}}(p)} \circ F_{\hat{H}}^{-1} \tag{18}$$

for all $p \in \mathcal{N}_{\hat{H}} := \{i,i+1\} \times \{j,j+1\} \times \{k,k+1\}$, where the bijective numbering $\rho_{\hat{H}} : \mathcal{N}_{\hat{H}} \to \{1,\dots,8\}$ is defined as

$$\begin{aligned}
\rho_{\hat{H}}((i,j,k)) &:= 1, & \rho_{\hat{H}}((i+1,j,k)) &:= 2,\\
\rho_{\hat{H}}((i+1,j+1,k)) &:= 3, & \rho_{\hat{H}}((i,j+1,k)) &:= 4,\\
\rho_{\hat{H}}((i,j,k+1)) &:= 5, & \rho_{\hat{H}}((i+1,j,k+1)) &:= 6,\\
\rho_{\hat{H}}((i+1,j+1,k+1)) &:= 7, & \rho_{\hat{H}}((i,j+1,k+1)) &:= 8
\end{aligned}$$

and the bijective mapping $F_{\hat{H}} : [-1,1]^3 \to [x_i, x_{i+1}] \times [y_i, y_{i+1}] \times [z_i, z_{i+1}]$ is given by

$$F_{\hat{H}}(x,y,z) := \frac{1}{2}\begin{pmatrix} x_i + x_{i+1} + (x_{i+1} - x_i)x\\ y_i + y_{i+1} + (y_{i+1} - y_i)y\\ z_i + z_{i+1} + (z_{i+1} - z_i)z \end{pmatrix}$$

with the determinant of its Jacobian

$$w_{\hat{H}} := \frac{1}{8}(x_{i+1}-x_i)(y_{i+1}-y_i)(z_{i+1}-z_i).$$

We set

$$\xi_x := \frac{2}{x_{i+1}-x_i}\partial_x\phi, \quad \xi_y := \frac{2}{y_{i+1}-y_i}\partial_y\phi, \quad \xi_z := \frac{2}{z_{i+1}-z_i}\partial_z\phi,$$

where $\partial_x, \partial_y, \partial_z$ denote the partial derivatives with respect to x, y, z. With this we define the functions $\mathbf{E}, \mathbf{S} : [-1,1]^3 \to \mathbb{R}^{24\times 6}$ as

$$\mathbf{E} := \begin{pmatrix} \xi_x & 0 & 0 & \xi_y & \xi_z & 0 \\ 0 & \xi_y & 0 & \xi_x & 0 & \xi_z \\ 0 & 0 & \xi_z & 0 & \xi_x & \xi_y \end{pmatrix}, \quad \mathbf{S} := \begin{pmatrix} (\lambda+2\mu)\xi_x & \lambda\xi_x & \lambda\xi_x & \mu\xi_y & \mu\xi_z & 0 \\ \lambda\xi_y & (\lambda+2\mu)\xi_y & \lambda\xi_y & \mu\xi_x & 0 & \mu\xi_z \\ \lambda\xi_z & \lambda\xi_z & (\lambda+2\mu)\xi_z & 0 & \mu\xi_x & \mu\xi_y \end{pmatrix}$$

and note that the rows of $\mathbf{E}$ and $\mathbf{S}$ evaluated at $(x,y,z) \in [-1,1]^3$ coincide with the engineering strain components $(\epsilon_{11}, \epsilon_{22}, \epsilon_{33}, 2\epsilon_{12}, 2\epsilon_{13}, 2\epsilon_{23})$ and the stress components $(\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23})$, respectively, applied to $\phi_t \circ F_{\hat{H}}^{-1} e_r$ for $t \in \{1,\dots,8\}$ and (then) for $r \in \{1,2,3\}$ and evaluated at $F_{\hat{H}}(x,y,z)$. In particular, it holds

$$\left(\epsilon(\phi_t \circ F_{\hat{H}}^{-1} e_r) : \sigma(\phi_l \circ F_{\hat{H}}^{-1} e_s)\right) \circ F_{\hat{H}} = \left(\mathbf{E}\mathbf{S}^\top\right)_{\nu(t,r),\nu(l,s)}$$

for all $(t,r),(l,s) \in \{1,\dots,8\} \times \{1,2,3\}$, where the bijective numbering $\nu : \{1,\dots,8\} \times \{1,2,3\} \to \{1,\dots,24\}$ is defined as $\nu(t,r) := 8(r-1)+t$. Thus, we conclude from (18)

$$\left(\epsilon(\eta_p e_r) : \sigma(\eta_q e_s)\right) \circ F_{\hat{H}} = \left(\pi_{\hat{H}} \mathbf{E}\mathbf{S}^\top \pi_{\hat{H}}^\top\right)_{\iota(p,r),\iota(q,s)} \tag{19}$$

for all $(p,s),(q,r) \in \mathcal{N} \times \{1,2,3\}$, where the so-called connectivity matrix $\pi_{\hat{H}} \in \mathbb{R}^{d\times 24}$ is defined as

$$\pi_{\hat{H},\iota(p,r),\nu(l,s)} := \begin{cases} \delta_{\rho_{\hat{H}}(p),l}\delta_{rs} & p \in \mathcal{N}_{\hat{H}}, \\ 0, & \text{else}, \end{cases} \quad (p,r) \in \mathcal{N} \times \{1,2,3\},\ (l,s) \in \{1,\dots,8\} \times \{1,2,3\}.$$

Moreover, for a function $h : D \to \mathbb{R}^3$ with $D \subset \hat{H}$ we have

$$\left(h \cdot (\phi_t \circ F_{\hat{H}}^{-1} e_r)\right) \circ F_{\hat{H}} = (\mathbf{P} h \circ F_{\hat{H}})_{\nu(t,r)}$$

for all $(t,r) \in \{1,\dots,8\} \times \{1,2,3\}$, where $\mathbf{P} : [-1,1]^3 \to \mathbb{R}^{24\times 3}$ is given by

$$\mathbf{P} := \begin{pmatrix} \phi & 0 & 0 \\ 0 & \phi & 0 \\ 0 & 0 & \phi \end{pmatrix}.$$

Using again (18) we obtain

$$(h \cdot (\eta_p e_r)) \circ F_{\hat{H}} = (\pi_{\hat{H}} \mathbf{P} h \circ F_{\hat{H}})_{\iota(p,r)} \tag{20}$$

for all $(p,r) \in \mathcal{N} \times \{1,2,3\}$. With these preparations at hand we define the local stiffness matrix $K_{\hat{H}} \in \mathbb{R}^{24\times 24}$ as $K_{\hat{H}} := w_{\hat{H}}(K_{\hat{H}}^- + \gamma K_{\hat{H}}^+)$ with $K_{\hat{H}}^\pm := K_{\hat{H}}(\mathcal{M}_{\hat{H}}^\pm)$ and

$$K_{\hat{H}}(\mathcal{M}) := \sum_{M\in\mathcal{M}} \int_{F_{\hat{H}}^{-1}(M)} \mathbf{E}\mathbf{S}^\top \, d(x,y,z), \quad \mathcal{M} \subset \mathcal{M}_{\hat{H}}^\pm. \tag{21}$$

The local load vector $L_{\hat{H}} \in \mathbb{R}^{24}$ is defined as $L_{\hat{H}} := w_{\hat{H}} L_{\hat{H}}^- + G_{\hat{H}}$ with $L_{\hat{H}}^- := L_{\hat{H}}(\mathcal{M}_{\hat{H}}^-)$, $G_{\hat{H}} := \sum_{H\in\mathcal{G}_{\hat{H}}}^* G_{\hat{H}}(H)$ and

$$L_{\hat{H}}^-(\mathcal{M}) := \sum_{M\in\mathcal{M}} \int_{F_{\hat{H}}^{-1}(M)} \mathbf{P} f \circ F_{\hat{H}} \, d(x,y,z), \quad \mathcal{M} \subset \mathcal{M}_{\hat{H}}^-, \tag{22}$$

$$G_{\hat{H}}(H) := \sum_{\star\in\{+,-\}} \int_{S_{\alpha\beta}^\star \cap H} \mathbf{P} \circ F_{\hat{H}}^{-1} g \, ds, \quad H = [\alpha,\alpha+1] \times [\beta,\beta+1] \times [z,z+v] \in \mathcal{G}_{\hat{H}}^*. \tag{23}$$

The assembling of the stiffness matrix and the load vector can be formulated as follows:

Theorem 10. *It holds*

$$K = \sum_{\hat{H}\in\mathcal{F}} \pi_{\hat{H}} K_{\hat{H}} \pi_{\hat{H}}^\top, \quad L = \sum_{\hat{H}\in\mathcal{F}} \pi_{\hat{H}} L_{\hat{H}}.$$

Proof. Let $(p,s),(q,r) \in \mathcal{N} \times \{1,2,3\}$. From (19), (20) and the change of variables formula we obtain

$$w_{\hat{H}} \left(\pi_{\hat{H}} K_{\hat{H}}^\pm \pi_{\hat{H}}^\top\right)_{\iota(p,r),\iota(q,s)} = \sum_{M\in\mathcal{M}_{\hat{H}}^\pm} \int_{F_{\hat{H}}^{-1}(M)} \left(\pi_{\hat{H}} (\mathbf{E}\mathbf{S}^\top) \pi_{\hat{H}}^\top\right)_{\iota(p,r),\nu(q,s)} w_{\hat{H}} \, d(x,y,z)$$

$$= \sum_{M\in\mathcal{M}^{\pm}_{\hat{H}}} \int_{F_{\hat{H}}^{-1}(M)} \big(\epsilon(\eta_p e_r) : \sigma(\eta_q e_s)\big)\circ F_{\hat{H}}\, w_{\hat{H}}\, d(x,y,z)$$
$$= \sum_{M\in\mathcal{M}^{\pm}_{\hat{H}}} \int_{M} \epsilon(\eta_p e_r) : \sigma(\eta_q e_s)\, d(\hat{x},\hat{y},\hat{z})$$

and

$$w_{\hat{H}}\left(\pi_{\hat{H}} L^{-}_{\hat{H}}\right)_{\iota(p,r)} = \sum_{M\in\mathcal{M}^{-}_{\hat{H}}} \int_{F_{\hat{H}}^{-1}(M)} \left(\pi_{\hat{H}}\mathbf{P} f\circ F_{\hat{H}}\right)_{\iota(p,r)} w_{\hat{H}}\, d(x,y,z)$$
$$= \sum_{M\in\mathcal{M}^{-}_{\hat{H}}} \int_{F_{\hat{H}}^{-1}(M)} (f\cdot(\eta_p e_r))\circ F_{\hat{H}}\, w_{\hat{H}}\, d(x,y,z)$$
$$= \sum_{M\in\mathcal{M}^{-}_{\hat{H}}} \int_{M} f\cdot(\eta_p e_r)\, d(\hat{x},\hat{y},\hat{z}).$$

Thus, we conclude

$$K_{\iota(p,r),\iota(q,s)} = a_\gamma(\eta_q e_s, \eta_p e_r)$$
$$= \sum_{\hat{H}\in\hat{G}} \left(\sum_{M\in\mathcal{M}^{-}_{\hat{H}}} \int_M \epsilon(\eta_p e_r) : \sigma(\eta_q e_s)\, d(\hat{x},\hat{y},\hat{z}) + \gamma \sum_{M\in\mathcal{M}^{+}_{\hat{H}}} \int_M \epsilon(\eta_p e_r) : \sigma(\eta_q e_s)\, d(\hat{x},\hat{y},\hat{z}) \right)$$
$$= \sum_{\hat{H}\in\hat{G}} \left(w_{\hat{H}} \left(\pi_{\hat{H}} K^{-}_{\hat{H}} \pi^{\top}_{\hat{H}}\right)_{\iota(p,r),\iota(q,s)} + \gamma w_{\hat{H}} \left(\pi_{\hat{H}} K^{+}_{\hat{H}} \pi^{\top}_{\hat{H}}\right)_{\iota(p,r),\iota(q,s)} \right)$$
$$= \sum_{\hat{H}\in\hat{G}} \left(\pi_{\hat{H}} K_{\hat{H}} \pi^{\top}_{\hat{H}}\right)_{\iota(p,r),\iota(q,s)}.$$

Moreover, reusing (20) we have

$$L_{\iota(p,r)} = \ell(\eta_p e_r)$$
$$= \sum_{\hat{H}\in\hat{G}} \left(\sum_{M\in\mathcal{M}^{-}_{\hat{H}}} \int_M f\cdot(\eta_p e_r)\, d(\hat{x},\hat{y},\hat{z}) + \sum_{H=[\alpha,\alpha+1]\times[\beta,\beta+1]\times[z,z+v]\in\mathcal{G}^{*}_{\hat{H}}} \sum_{\star\in\{+,-\}} \int_{S^{\star}_{\alpha\beta}\cap H} g\cdot(\eta_p e_r)\, ds \right)$$
$$= \sum_{\hat{H}\in\hat{G}} \left(w_{\hat{H}} \left(\pi_{\hat{H}} L^{-}_{\hat{H}}\right)_{\iota(p,r)} + \sum_{H\in\mathcal{G}^{*}_{\hat{H}}} \left(\pi_{\hat{H}} G_{\hat{H}}(H)\right)_{\iota(p,r)} \right)$$
$$= \sum_{\hat{H}\in\hat{G}} \left(\pi_{\hat{H}} L_{\hat{H}}\right)_{\iota(p,r)}. \quad \square$$

The specific sparse structure of the connectivity matrix $\pi_{\hat{H}}$ for $\hat{H}\in\mathcal{F}$ can be expressed by the vector $c_{\hat{H}}\in\mathbb{R}^{24}$ defined as

$$c_{\hat{H},\nu(l,s)} := \iota(\rho^{-1}_{\hat{H}}(l), s), \quad (l,s)\in\{1,\dots,8\}\times\{1,2,3\}.$$

Note that

$$\{1,\dots,d\} = \left\{ c_{\hat{H},\kappa};\ \hat{H}\in\mathcal{F},\ \kappa\in\{1,\dots,24\} \right\}. \tag{24}$$

Lemma 11. *Let $\hat{H}\in\mathcal{F}$. Then, $\pi_{\hat{H},\theta\kappa} = \delta_{\theta,c_{\hat{H},\kappa}}$ for all $\theta\in\{1,\dots,d\}$ and all $\kappa\in\{1,\dots,24\}$.*

Proof. Let $(p,r)\in\mathcal{N}\times\{1,2,3\}$ with $\theta=\iota(p,r)$ and $(l,s)\in\{1,\dots,8\}\times\{1,2,3\}$ with $\kappa=\nu(l,s)$. In the case $p\in\mathcal{N}_{\hat{H}}$ we have

$$\pi_{\hat{H},\theta\kappa} = \pi_{\hat{H},\iota(p,r),\nu(l,s)} = \delta_{l,\rho_{\hat{H}}(p)}\delta_{rs} = \delta_{\iota(\rho^{-1}_{\hat{H}}(l),s),\iota(p,s)}\delta_{\iota(p,r),\iota(p,s)} = \delta_{\iota(\rho^{-1}_{\hat{H}}(l),s),\iota(p,r)} = \delta_{\theta,c_{\hat{H},\nu(l,s)}} = \delta_{\theta,c_{\hat{H},\kappa}}.$$

For $p\notin\mathcal{N}_{\hat{H}}$ we immediately obtain $\pi_{\hat{H},\theta\kappa}=0$ from the definition of $\pi_{\hat{H}}$. Moreover, we have $p\neq\rho^{-1}_{\hat{H}}(l)$ and, thus, $\iota(p,r)\neq\iota(\rho^{-1}_{\hat{H}}(l),s)$. Hence, we get

$$\delta_{\theta,c_{\hat{H},\kappa}} = \delta_{\iota(p,r),c_{\hat{H},\nu(l,s)}} = \delta_{\iota(p,r),\iota(\rho^{-1}_{\hat{H}}(l),s)} = 0. \quad \square$$

Theorem 12. *It holds*

$$K_{c_{\hat{H},\kappa},c_{\hat{H},\ell}} = \sum_{\hat{H}\in\mathcal{F}} K_{\hat{H},\kappa\ell}, \quad L_{c_{\hat{H},\kappa}} = \sum_{\hat{H}\in\mathcal{F}} L_{\hat{H},\kappa}$$

for all $\kappa,\ell\in\{1,\dots,24\}$.

Proof. From $c_{\hat{H},\kappa} \neq c_{\hat{H},\tilde{\kappa}}$ for $\kappa \neq \tilde{\kappa}$ we get $\delta_{c_{\hat{H},\kappa},c_{\hat{H},\tilde{\kappa}}} = \delta_{\kappa,\tilde{\kappa}}$. Hence, applying Theorem 10 and Lemma 11 yields

$$K_{c_{\hat{H},\kappa},c_{\hat{H},\ell}} = \sum_{\hat{H}\in\mathcal{F}} \sum_{\tilde{\kappa},\tilde{\ell}=1}^{24} \pi_{\hat{H},c_{\hat{H},\kappa},\tilde{\kappa}} K_{\hat{H},\tilde{\kappa},\tilde{\ell}} \pi_{\hat{H},c_{\hat{H},\ell},\tilde{\ell}} = \sum_{\hat{H}\in\mathcal{F}} \sum_{\tilde{\kappa},\tilde{\ell}=1}^{24} \delta_{c_{\hat{H},\kappa},c_{\hat{H},\tilde{\kappa}}} K_{\hat{H},\tilde{\kappa},\tilde{\ell}} \delta_{c_{\hat{H},\ell},c_{\hat{H},\tilde{\ell}}} = \sum_{\hat{H}\in\mathcal{F}} K_{\hat{H},\kappa\ell}$$

and

$$L_{c_{\hat{H},\kappa}} = \sum_{\hat{H}\in\mathcal{F}} \sum_{\tilde{\kappa}=1}^{24} \pi_{\hat{H},c_{\hat{H},\kappa},\tilde{\kappa}} L_{\hat{H},\tilde{\kappa}} = \sum_{\hat{H}\in\mathcal{F}} \sum_{\tilde{\kappa}=1}^{24} \delta_{c_{\hat{H},\kappa},c_{\hat{H},\tilde{\kappa}}} L_{\hat{H},\tilde{\kappa}} = \sum_{\hat{H}\in\mathcal{F}} L_{\hat{H},\kappa}. \quad \square$$

Algorithm 2 describes the assembling of K and L resulting from Theorem 12 (where (24) is taken into account), i.e. $(K, L) = \text{Assemble}(\mathcal{F})$. Note that

$$K_{\hat{H}} = w_{\hat{H}} \left(K_{\hat{H}}(G^-_{\hat{H}}) + \gamma K_{\hat{H}}(G^-_{\hat{H}}) + \sum_{H\in\mathcal{G}_{\hat{H}}}^{*} \left(K_{\hat{H}}(\mathcal{Z}^+(H)) + \gamma K_{\hat{H}}(\mathcal{Z}^+(H)) \right) \right)$$

and

$$L_{\hat{H}} = w_{\hat{H}} \left(L_{\hat{H}}(G^-_{\hat{H}}) + \sum_{H\in\mathcal{G}_{\hat{H}}}^{*} \left(L_{\hat{H}}(\mathcal{Z}^-(H)) \right) \right).$$

The assembling is based on a loop through the elements of $\mathcal{F}$, which is very typical for finite element methods. Furthermore, a two-time application of Algorithm 3 generates the constrained stiffness matrix K' and the constrained load vector L' from their unconstrained counterparts K and L, i.e.

$$(K', L') = \text{Constrain}(\text{Constrain}(K, L, \mathcal{N}^a_D, \{1,2,3\}), \mathcal{N}_D, \{1,2\}).$$

The overall algorithm for the computation of the solution vector $\zeta \in \mathbb{R}^d$ in (17) is given by Algorithm 4, i.e. $\zeta = \text{DEM–FCM}(\mathcal{F})$.

In practice, appropriate numerical integration schemes can be used to (approximately) compute the integrals in (21)–(23). For this purpose, let $\hat{H} \in \mathcal{F}$, $M \in \mathcal{M}^{\pm}_{\hat{H}}$ and set $\hat{M} := F^{-1}_{\hat{H}}(M)$. We define $E_{\hat{M}} : B_{\hat{M}} \to \hat{M}$ as

$$E_{\hat{M}}(\bar{x},\bar{y},\bar{z}) := \begin{cases} F_{\hat{M}}(\bar{x},\bar{y},\bar{z}), & \hat{M} \text{ is a hexahedron,} \\ P_1 + (P_2 - P_1)\bar{x} + (P_3 - P_1)\bar{y} + (P_4 - P_1)\bar{z}, & \hat{M} \text{ is a tetrahedron} \end{cases}$$

where $B_{\hat{M}}$ is $[-1,1]^3$ if $\hat{M}$ is a hexahedron and the 3-dimensional unit simplex

$$\bar{T}_3 := \text{conv}((0,0,0),(1,0,0),(0,1,0),(0,0,1))$$

if $\hat{M} = \text{conv}(P_1, P_2, P_3, P_4)$ with $P_1, \ldots, P_4 \in [-1,1]^3$ is a tetrahedron. Note that the absolute value of the determinant of the Jacobian of $E_{\hat{M}}$ is constant and is given by

$$\mu_{\hat{M}} := |\det\left(\nabla E_{\hat{M}}(\bar{x},\bar{y},\bar{z})\right)| = \begin{cases} w_{\hat{M}}, & \hat{M} \text{ is a hexahedron,} \\ |\det\begin{pmatrix} P_2 - P_1 & P_3 - P_1 & P_4 - P_1 \end{pmatrix}|, & \hat{M} \text{ is a tetrahedron.} \end{cases}$$

With these preparations we approximately compute the integrals in (21) and (22) by

$$\int_{\hat{M}} \mathbf{E}\mathbf{S}^\top \, d(x,y,z) = \mu_{\hat{M}} \int_{B_{\hat{M}}} \mathbf{E}\circ E_{\hat{M}} \mathbf{S}\circ E^\top_{\hat{M}} \, d(\bar{x},\bar{y},\bar{z}) \approx \mu_{\hat{M}} \sum_{\lambda=1}^{Q_{\hat{M}}} \omega_{\hat{M},\lambda} \mathbf{E}\left(E_{\hat{M}}(G_{\hat{M},\lambda})\right) \mathbf{S}(E_{\hat{M}}(G_{\hat{M},\lambda}))^\top$$

and

$$\int_{\hat{M}} \mathbf{P} f\circ F_{\hat{H}} \, d(x,y,z) = \mu_{\hat{M}} \int_{B_{\hat{M}}} \mathbf{P}\circ E_{\hat{M}} f\circ F_{\hat{H}}\circ E_{\hat{M}} \, d(\bar{x},\bar{y},\bar{z}) \approx \mu_{\hat{M}} \sum_{\lambda=1}^{Q_{\hat{M}}} \omega_{\hat{M},\lambda} \mathbf{P}(E_{\hat{M}}(G_{\hat{M},\lambda}) f(F_{\hat{H}}(E_{\hat{M}}(G_{\hat{M},\lambda}))),$$

where $G_{\hat{M},\lambda} \in B_{\hat{M}}$ and $\omega_{\hat{M},\lambda} \in \mathbb{R}$, $\lambda = 1, \ldots, Q_{\hat{M}}$, denote some quadrature points and weights, respectively. If $\hat{M}$ is a hexahedron, one may choose the Gauss quadrature rule with 8 points, i.e. $Q_{\hat{M}} := 8$, $G_{\hat{M},\lambda} \in \{-3^{-1/2}, 3^{-1/2}\}^3$ and $\omega_{\hat{M},\lambda} := 1$. If $\hat{M}$ is a tetrahedron, one may apply an appropriate numerical integration formula on $\bar{T}_3$, [41]. Note that if $\hat{\mathcal{G}}_{\hat{H}} = \{\hat{H}\}$ then $\hat{M} = [-1,1]^3$ and the computation of the integrals in (21) and (22) simplifies to

$$\int_{\hat{M}} \mathbf{E}\mathbf{S}^\top \, d(x,y,z) \approx \sum_{\lambda=1}^{Q_{\hat{M}}} \omega_{\hat{M},\lambda} \mathbf{E}\left(G_{\hat{M},\lambda}\right) \mathbf{S}\left(G_{\hat{M},\lambda}\right)^\top$$

and

$$\int_{\hat{M}} \mathbf{P} f\circ F_{\hat{H}} \, d(x,y,z) \approx \sum_{\lambda=1}^{Q_{\hat{M}}} \omega_{\hat{M},\lambda} \mathbf{P}(G_{\hat{M},\lambda}) f(F_{\hat{H}}(G_{\hat{M},\lambda})).$$

To compute the integral in (23) we observe that $S^\star_{\alpha\beta} \cap H = \bigcup\{S^\star_{\alpha\beta} \cap T;\ T \in \mathcal{Z}^-(H)\}$ and that $S^\star_{\alpha\beta} \cap \operatorname{relint}(T)$ with $T \in \mathcal{Z}^-(H)$ is empty or a triangular facet of T given by $\operatorname{conv}(P_1, P_2, P_3)$ with some $P_1, P_2, P_3 \in \mathbb{R}^3$. In the latter, we define the linear transformation $E^\star_{\alpha\beta} : \bar{T}_2 \to S^\star_{\alpha\beta} \cap T$ on the 2-dimensional unit simplex $\bar{T}_2 := \operatorname{conv}((0,0),(1,0),(0,1))$ by $E^\star_{\alpha\beta}(\bar{x},\bar{y}) := P_1 + (P_2 - P_1)\bar{x} + (P_3 - P_1)\bar{y}$. With

$$\mu^\star_{\alpha\beta} := |\det\left(\nabla E^{\star\top}_{\alpha\beta} \nabla E^\star_{\alpha\beta}\right)|^{1/2} = \|(P_2 - P_1) \times (P_3 - P_1)\|$$

and some quadrature points $G_\lambda \in \bar{T}_2$ and weights $\omega_\lambda \in \mathbb{R}$, $\lambda = 1, \dots, Q$, given by a suitable numerical integration formula on $\bar{T}_2$, see, e.g., [41], we approximate

$$\int_{S^\star_{\alpha\beta} \cap T} \mathbf{P} \circ F^{-1}_{\hat{H}}\, g\, ds = \mu^\star_{\alpha\beta} \int_{\bar{T}_2} \mathbf{P} \circ F^{-1}_{\hat{H}} \circ E^\star_{\alpha\beta} g \circ E^\star_{\alpha\beta}\, d(\bar{x},\bar{y}) \approx \mu^\star_{\alpha\beta} \sum_{\lambda=1}^{Q} \omega_\lambda \mathbf{P}(F^{-1}_{\hat{H}}(E^\star_{\alpha\beta}(G_\lambda))) g(E^\star_{\alpha\beta}(G_\lambda)).$$

Remark 2. The proposed computation of the local stiffness matrix $K_{\hat{H}}$ as well as of the local load vector $L_{\hat{H}}$ only require the efficient evaluation of the shape functions ϕ and their partial derivatives ξ_x, ξ_y and ξ_z at the transformed quadrature points $E_{\hat{M}}(G_{\hat{M}\lambda})$ and a copy of these vectors into the (block) matrices **E**, **S** and **P**.

Remark 3. A natural choice for the set $\mathcal{F}$ is $\hat{\mathcal{G}}^- \cup \hat{\mathcal{G}}^*$. In this case the fictitious domain Ω^+ as well as the number of unknowns in the linear system (16) are minimal. Choosing $\mathcal{F}$ as $\hat{\mathcal{G}}$ leads to the maximum number of unknowns $d = 3NMB$ in (16). Moreover, it holds $\mathcal{N} = \hat{\mathcal{N}}$, $\mathcal{N} = \hat{\mathcal{N}}^a_D$ and $\mathcal{N}_D = \hat{\mathcal{N}}_D$ and the bijective numbering ι can simply be specified by

$$\iota(p,r) := 3BM(p_1 - 1) + 3B(p_2 - 1) + 3(p_3 - 1) + r$$

for $p = (p_1, p_2, p_3) \in \mathcal{N}$.

Remark 4. The for-loop through the set $\mathcal{F}$ in Algorithm 2 can be realized via a three times for-loop through the sets $\{1, \dots, N-1\}$, $\{1, \dots, M-1\}$ and $\{1, \dots, B-1\}$ with loop indices i, j and k and the query $H = [x_i, x_{i+1}] \times [y_j, y_j + 1] \times [z_k, z_{k+1}] \in \mathcal{F}$. In the case of $\mathcal{F} = \hat{\mathcal{G}}^- \cup \hat{\mathcal{G}}^*$ this query is given by (2) or (2) is not true and $z_k < A_{ij}$.

The for-loop through $\hat{\mathcal{N}} \times D$ in Algorithm 3 may be replaced by a specific iteration which efficiently exploits the structure of the particular sets $\mathcal{N}^a_D \times \{1,2,3\}$ and $\mathcal{N}_D \times \{1,2\}$.

Remark 5. In Algorithm 2 the sets $\mathcal{G}_0$, $\mathcal{G}_1$ and $\mathcal{G}_2$ as well as $\mathcal{Z}_1$ and $\mathcal{Z}_2$ are only available within the loop through $\mathcal{F}$. This means that the sets $\mathcal{M}^+$ and $\mathcal{M}^-$ are never stored in memory, which drastically reduces the memory amount for complex geometries.

Remark 6. The stiffness matrix K is symmetric. However, Algorithms 2 and 3 do not take this property into account. Exploiting the symmetry of K may lead to further improvements with respect to efficiency and memory.

6. Numerical verification

To verify that the DEM-FCM yields similar numerical results to standard lowest-order finite element schemes, we conduct several benchmark studies. For this purpose, we discuss the convergence behavior of the numerical solutions computed by the DEM-FCM and the FEM software ELMER [42]. We consider domains with simple shapes so that the problem configurations can also be easily realized in ELMER. They are specified by a matrix $A \in [1,129]^{128\times128}$ representing the underlying DEM. For the DEM-FCM we always use consecutive uniform refinements of the rectangular hexahedron $\mathbb{B} := [1,129]^3$ based on divisions into eight rectangular subhexahedrons of the same size yielding the regular grid $\hat{\mathcal{G}}$ of $\mathbb{B}$. We also use such uniform refinements of Ω in the benchmark study of Section 6.1 together with piecewise trilinear polynomials for ELMER. When we use ELMER in Sections 6.2–6.4 we apply consecutive uniform refinements of appropriate initial tetrahedron meshes. The refinements are based on subdivisions into eight subtetrahedrons of the same size. Furthermore, we use piecewise linear polynomials on these meshes in the case of ELMER. Within the DEM-FCM we apply Algorithm 1 for generating a DEM grid $\mathcal{G}_{\hat{H}}$ for each $\hat{H} \in \mathcal{F}$, where we take $\mathcal{F} := \hat{\mathcal{G}}^- \cup \hat{\mathcal{G}}^*$, see Remark 3. We examine the error of the exact solution u of (12) and the computed solution $u^\star_k \in V^\star_k$ by applying the energy norm given by $\|v\|_a := a(v,v)^{1/2}$ for $v \in V$. Here, $\star \in \{\mathrm{FCM}, \mathrm{ELM}\}$ indicates the use of the DEM-FCM or ELMER and $V^\star_k$ is the finite element space (13) associated to the refinement level $k \in \{0, \dots, n\}$ with maximum refinement level n. We expect an algebraic convergence behavior in terms of the mesh size, i.e.

$$\|u - u^\star_k\|_a \le C^\star {h^\star_k}^{\kappa^\star}, \tag{25}$$

where $\kappa^\star > 0$ is the convergence order, $C^\star > 0$ is a constant and $h^\star_k > 0$ denotes the mesh size of the refinement level k. As we assume uniform refinements we have $h^\star_k := h^\star_0 2^{-k}$ with an initial mesh size $h^\star_0 > 0$. Alternatively, we may express the algebraic convergence in terms of the (unconstrained) degrees of freedom, i.e.

$$\|u - u^\star_k\|_a \le D^\star {d^\star_k}^{-\delta^\star}. \tag{26}$$

Here, the convergence order is denoted by $\delta^\star > 0$ and the constant by $D^\star > 0$. The number of degrees of freedom is given by $d^\star_k := \dim W^\star_k$ with the unconstrained finite element space $W^\star_k$ given by (14).

Exploiting the Galerkin orthogonality, i.e.

$$a(u - u_k^\star, v_k^\star) = 0$$

for all $v_k^\star \in V_k^\star$, we have

$$\|u - u_k^\star\|_a^2 = a(u - u_k^\star, u - u_k^\star) = a(u - u_k^\star, u + u_k^\star) = a(u,u) - a(u_k^\star, u_k^\star).$$

Thus, we obtain from (25) and (26)

$$a(u,u) - a_k^\star \leq C^{\star 2}(h_0^\star 2^{-k})^{2\kappa^\star} \tag{27}$$

and

$$a(u,u) - a_k^\star \leq D^{\star 2} d_k^{\star -2\delta^\star} \tag{28}$$

with $a_k^\star := a(u_k^\star, u_k^\star)$. As the exact solution u is usually unknown, $a(u,u)$ in (27) and (28) has to be approximated by a reference value a_{ref}, which can be computed, for instance, by using the Richardson extrapolation, i.e. $a_{\text{ref}} := a_{n,n}^\star$ with

$$a_{k,0}^\star := a_k^\star, \quad a_{k,j}^\star := a_{k,j-1}^\star + \frac{a_{k,j-1}^\star - a_{k-1,j-1}^\star}{2^j - 1}, \quad k = 0, \dots, n,\ j = 1, \dots, k,$$

where an expansion of $a(u,u)$ in terms of the mesh size is assumed to be held [46, Ch.3.4]. Eventually, setting $q_k^\star := a_{\text{ref}} - a_k^\star$ and using (27) and (28) we approximate the convergence orders $\kappa^\star$ and $\delta^\star$ by

$$\kappa_k^\star := \frac{\ln(q_k^\star / q_{k+1}^\star)}{2\ln(2)}, \quad \delta_k^\star := \frac{\ln(q_k^\star / q_{k+1}^\star)}{2\ln(d_{k+1}^\star / d_k^\star)}$$

and the constants $C^\star$ and $D^\star$ by $C_k^\star := e_k^\star (h_0^\star 2^{-k})^{-\kappa_k^\star}$ and $D_k^\star := e_k^\star d_k^{\star \delta_k^\star}$ with $e_k^\star := q_k^{\star 1/2}$, where $k \in \{0, \dots, n-1\}$. Note that $e_k^\star$ is an approximation of the error $\|u - u_k^\star\|_a$.

In the benchmark studies we compute the quantities $\alpha_k^\star$, $\kappa_k^\star$, $C_k^\star$, $\delta_k^\star$ and $D_k^\star$ up to a maximum refinement level $n = 7$. Since both the DEM-FCM and ELMER use lowest-order finite elements on uniformly refined meshes, we expect that these quantities vary in similar ranges. Note that the convergence orders $\kappa^\star$ and $\delta^\star$ are restricted by 1 and 1/3, respectively, and may be further reduced as a result of the possibly low regularity of the exact solution. This behavior should also be observable for the approximated convergence orders. Moreover, we discuss $C_k^\star$ and $D_k^\star$, which can also provide insight into the efficiency of the approaches. We use the extrapolated value $a_{\text{Ref}} := a_{7,7}^{\text{ELM}}$ as a reference value in the first three benchmark studies. In the last benchmark study, we again use ELMER to compute a suitable reference value, but on a fine mesh that does not result from uniform refinements. Note that when plotting $e_k^\star$ versus $d_k^\star$ in a log–log diagram, $-\delta_k^\star$ can be interpreted as the slope and $D_k^\star$ as the vertical intercept of the resulting graph (which should be approximately a straight line).

All benchmark studies employ the same weak formulation, material properties and boundary conditions. They differ only in the geometry of their domains. The Lamé constants are given by

$$\lambda := \frac{\nu E}{(1-2\nu)(1+\nu)}, \quad \mu := \frac{E}{2(1+\nu)} \tag{29}$$

with Young's Modulus $E := 53.3 \cdot 10^9$ Pa and Poisson's Ratio $\nu := 0.37$. The volume force is set to

$$f := \rho g \tag{30}$$

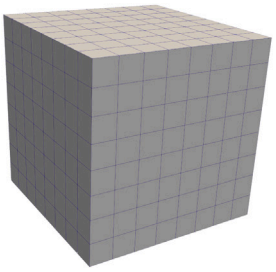
with density $\rho := 2660\ \text{kg/m}^3$ and gravitational acceleration $g := 9.81\ \text{m/s}^2$. The surface force g is set to 0.

The linear system arising from the discretization is solved iteratively using a conjugate gradient (CG) method with Jacobi preconditioning. A stopping criterion based on the relative residual norm is applied, whereby iterations are terminated once the Euclidean norm of the residual is below 10^{-10} relative to the norm of the right-hand side, i.e. $\|K'\zeta' - L'\|/\|L'\| \leq 10^{-10}$. Numerical experiments performed with ELMER employ a BiCGStab solver with ILU(0) preconditioning and the same stopping criterion. The system matrices resulting from the FCM are known to exhibit less favorable conditioning compared to standard finite elements. This is primarily related to the use of material penalization in fictitious regions, which can lead to strong contrasts in stiffness contributions at the element level. In the simulations, this behavior is observable in the conditioning of the system, but it does not significantly affect solver robustness or convergence rates. The iterative scheme consistently converges within a reasonable number of iterations for the problem sizes considered.

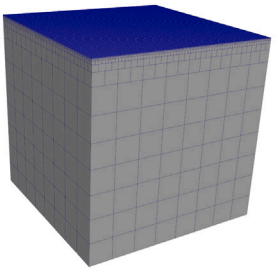
6.1. Cube benchmark

In this benchmark study the matrix A is defined as $A_{ij} := 128.5$ for $i, j \in \{1, \dots, 129\}$. As the resulting domain Ω is a little smaller than $\text{int}(\mathbb{B})$, the DEM-FCM differs from a classical lowest-order finite element method on hexahedrons. Indeed, the DEM surface $\mathbb{S} = [1, 129]^2 \times \{128.5\}$ is resolved by $\mathcal{M}^+$, which only consists of tretrahedrons for all refinement levels $k = 0, \dots, 7$.

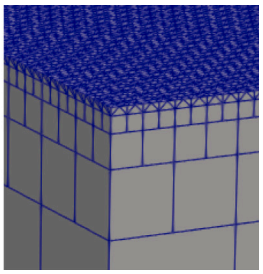
The Figs. 4(a) and 4(b) show the uniform refinements for ELMER and $\mathcal{M}^-$ of the DEM-FCM for the refinement level $k = 3$. In Fig. 4(c) a zoom of $\mathcal{M}^-$ is depicted which shows tetrahedrons at the DEM surface. In this benchmark study the initial mesh sizes are given by $h_0^{\text{FCM}} = 3^{1/2} \cdot 128$ and $h_0^{\text{ELM}} = (2 \cdot 128^2 + 127.5^2)^{1/2}$.



(a) Uniform refinements of the initial mesh of level $k = 3$.



(b) $\mathcal{M}^-$ for the refinement level $k = 3$.



(c) Zoom of $\mathcal{M}^-$.

Fig. 4. Refinements in the cube benchmark.

Table 2
$d_k^\star$ and $a_k^\star$ in the cube benchmark.

k	d_k^{FCM}	d_k^{ELM}	a_k^{FCM}	a_k^{ELM}
0	$2.4 \cdot 10^1$	$2.4 \cdot 10^1$	$6.132223102190417 \cdot 10^7$	$6.1322231021905437 \cdot 10^7$
1	$8.1 \cdot 10^1$	$8.1 \cdot 10^1$	$7.665278877738148 \cdot 10^7$	$7.6652788777381808 \cdot 10^7$
2	$3.75 \cdot 10^2$	$3.75 \cdot 10^2$	$8.048542821625091 \cdot 10^7$	$8.0485428216250882 \cdot 10^7$
3	$2.187 \cdot 10^3$	$2.187 \cdot 10^3$	$8.144358807573235 \cdot 10^7$	$8.1443588075732365 \cdot 10^7$
4	$1.4739 \cdot 10^4$	$1.4739 \cdot 10^4$	$8.168312804070397 \cdot 10^7$	$8.1683128040715098 \cdot 10^7$
5	$1.07811 \cdot 10^5$	$1.07811 \cdot 10^5$	$8.174301303211235 \cdot 10^7$	$8.1743013031962499 \cdot 10^7$
6	$8.23875 \cdot 10^5$	$8.23875 \cdot 10^5$	$8.175798427319297 \cdot 10^7$	$8.1757984279837385 \cdot 10^7$
7	$6.440067 \cdot 10^6$	$6.440067 \cdot 10^6$	$8.176172700066817 \cdot 10^7$	$8.1761727091829434 \cdot 10^7$
$a_{7,7}^\star$			$8.176297440545857 \cdot 10^7$	$8.176297469584754 \cdot 10^7$

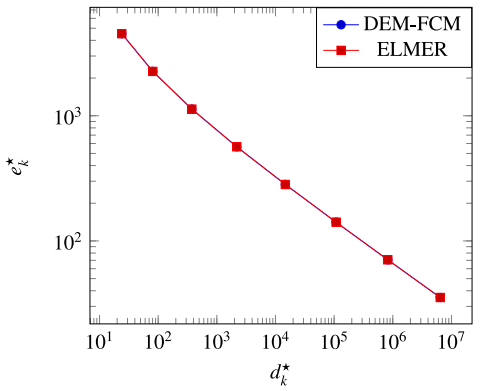
Table 3
$\kappa_k^\star$ and $C_k^\star$ in the cube benchmark.

k	κ_k^{FCM}	κ_k^{ELM}	C_k^{FCM}	C_k^{ELM}
0	$1.00014 \cdot 10^0$	$1.00014 \cdot 10^0$	$2.03771 \cdot 10^1$	$2.04047 \cdot 10^1$
1	$1.00014 \cdot 10^0$	$1.00014 \cdot 10^0$	$2.03797 \cdot 10^1$	$2.04064 \cdot 10^1$
2	$1.00014 \cdot 10^0$	$1.00014 \cdot 10^0$	$2.0382 \cdot 10^1$	$2.04082 \cdot 10^1$
3	$1.00014 \cdot 10^0$	$1.00014 \cdot 10^0$	$2.03834 \cdot 10^1$	$2.04098 \cdot 10^1$
4	$1 \cdot 10^0$	$1 \cdot 10^0$	$2.03927 \cdot 10^1$	$2.0419 \cdot 10^1$
5	$1.0002 \cdot 10^0$	$1.0002 \cdot 10^0$	$2.03898 \cdot 10^1$	$2.04025 \cdot 10^1$
6	$9.997 \cdot 10^{-1}$	$9.997 \cdot 10^{-1}$	$2.04041 \cdot 10^1$	$2.0426 \cdot 10^1$

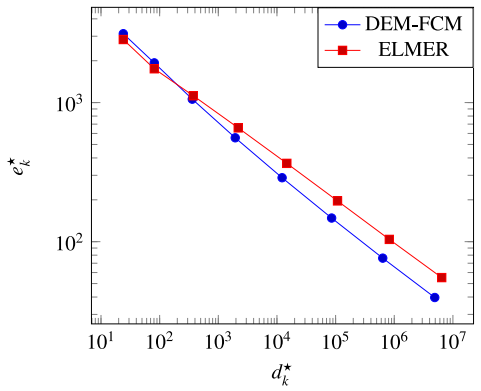
Table 4
$\delta_k^\star$ and $D_k^\star$ in the cube benchmark.

k	δ_k^{FCM}	δ_k^{ELM}	D_k^{FCM}	D_k^{ELM}
0	$5.6987 \cdot 10^{-1}$	$5.6987 \cdot 10^{-1}$	$2.76534 \cdot 10^4$	$2.76534 \cdot 10^4$
1	$4.5232 \cdot 10^{-1}$	$4.5232 \cdot 10^{-1}$	$1.64976 \cdot 10^4$	$1.64976 \cdot 10^4$
2	$3.931 \cdot 10^{-1}$	$3.931 \cdot 10^{-1}$	$1.16143 \cdot 10^4$	$1.16143 \cdot 10^4$
3	$3.6331 \cdot 10^{-1}$	$3.6331 \cdot 10^{-1}$	$9.23641 \cdot 10^3$	$9.23641 \cdot 10^3$
4	$3.483 \cdot 10^{-1}$	$3.483 \cdot 10^{-1}$	$7.9974 \cdot 10^3$	$7.9974 \cdot 10^3$
5	$3.4088 \cdot 10^{-1}$	$3.4088 \cdot 10^{-1}$	$7.33781 \cdot 10^3$	$7.33781 \cdot 10^3$
6	$3.3698 \cdot 10^{-1}$	$3.3698 \cdot 10^{-1}$	$6.95885 \cdot 10^3$	$6.95885 \cdot 10^3$

In Table 2 the number $d_k^\star$ of the degrees of freedom and the quantity $\alpha_k^\star$ are tabulated. Obviously, the DEM-FCM and ELMER coincide in both numbers (except for some less relevant digits). The Tables 3 and 4 show that the approximated convergence orders $\kappa_k^\star$ and $\delta_k^\star$ as well as the approximated constants $C_k^\star$ and $D_k^\star$ exactly coincide in the printed digits. In particular, $\kappa_k^\star$ and $\delta_k^\star$ have the expected values close to 1 and $1/3$, respectively, indicating that the solution u is sufficiently regular. Fig. 5(a) shows the error $e_k^\star$ versus $d_k^\star$ in a log–log diagram. For larger $d_k^\star$ the graphs become a straight line with the slope $-\delta_k^\star$ and the vertical intercept $D_k^\star$. Clearly, the graphs of the DEM-FCM and ELMER lie on top of each other.



(a) The error $e_k^\star$ versus the number $d_k^\star$ of degrees of freedom in the cube benchmark.



(b) The error $e_k^\star$ versus the number $d_k^\star$ of degrees of freedom in the monopitch roof benchmark.

Fig. 5. The error behavior in the cube and monopitch roof benchmark.

Table 5
$d_k^\star$ and $a_k^\star$ in the monopitch roof benchmark.

k	d_k^{FCM}	d_k^{ELM}	a_k^{FCM}	a_k^{ELM}
0	$2.4 \cdot 10^1$	$2.4 \cdot 10^1$	$2.875864120876109 \cdot 10^7$	$3.0469401998509865 \cdot 10^7$
1	$8.1 \cdot 10^1$	$8.1 \cdot 10^1$	$3.481513187826067 \cdot 10^7$	$3.5478952864913747 \cdot 10^7$
2	$3.6 \cdot 10^2$	$3.75 \cdot 10^2$	$3.742385490904834 \cdot 10^7$	$3.7289067932722837 \cdot 10^7$
3	$1.944 \cdot 10^3$	$2.187 \cdot 10^3$	$3.82318477890572 \cdot 10^7$	$3.8110975940912932 \cdot 10^7$
4	$1.224 \cdot 10^4$	$1.4739 \cdot 10^4$	$3.8460695328640774 \cdot 10^7$	$3.841004799254524 \cdot 10^7$
5	$8.5536 \cdot 10^4$	$1.07811 \cdot 10^5$	$3.8521941927145615 \cdot 10^7$	$3.8505229011554912 \cdot 10^7$
6	$6.3648 \cdot 10^5$	$8.23875 \cdot 10^5$	$3.853802365004157 \cdot 10^7$	$3.8533036486417003 \cdot 10^7$
7	$4.904064 \cdot 10^6$	$6.440067 \cdot 10^6$	$3.854224946319299 \cdot 10^7$	$3.8540781362928614 \cdot 10^7$
$a_{7,7}^\star$			$3.8543854733067326 \cdot 10^7$	$3.8543826968029365 \cdot 10^7$

Table 6
$\kappa_k^\star$ and $C_k^\star$ in the monopitch roof benchmark.

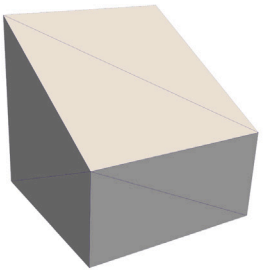
k	κ_k^{FCM}	κ_k^{ELM}	C_k^{FCM}	C_k^{ELM}
0	$6.96033 \cdot 10^{-1}$	$6.98825 \cdot 10^{-1}$	$7.28665 \cdot 10^1$	$7.20999 \cdot 10^1$
1	$8.6765 \cdot 10^{-1}$	$6.44283 \cdot 10^{-1}$	$3.24805 \cdot 10^1$	$9.24743 \cdot 10^1$
2	$9.2206 \cdot 10^{-1}$	$7.6784 \cdot 10^{-1}$	$2.61102 \cdot 10^1$	$5.73243 \cdot 10^1$
3	$9.5398 \cdot 10^{-1}$	$8.4706 \cdot 10^{-1}$	$2.34834 \cdot 10^1$	$4.45681 \cdot 10^1$
4	$9.6281 \cdot 10^{-1}$	$8.967 \cdot 10^{-1}$	$2.29455 \cdot 10^1$	$3.93936 \cdot 10^1$
5	$9.5755 \cdot 10^{-1}$	$9.195 \cdot 10^{-1}$	$2.31823 \cdot 10^1$	$3.78231 \cdot 10^1$
6	$9.4035 \cdot 10^{-1}$	$9.1245 \cdot 10^{-1}$	$2.36877 \cdot 10^1$	$3.8129 \cdot 10^1$

Table 7
$\delta_k^\star$ and $D_k^\star$ in the monopitch roof benchmark.

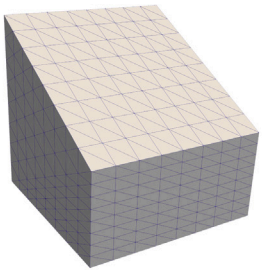
k	δ_k^{FCM}	δ_k^{ELM}	D_k^{FCM}	D_k^{ELM}
0	$3.96587 \cdot 10^{-1}$	$3.98178 \cdot 10^{-1}$	$1.10309 \cdot 10^4$	$1.00713 \cdot 10^4$
1	$4.0314 \cdot 10^{-1}$	$2.91382 \cdot 10^{-1}$	$1.13524 \cdot 10^4$	$6.29872 \cdot 10^3$
2	$3.7895 \cdot 10^{-1}$	$3.0179 \cdot 10^{-1}$	$9.8463 \cdot 10^3$	$6.69951 \cdot 10^3$
3	$3.5934 \cdot 10^{-1}$	$3.0771 \cdot 10^{-1}$	$8.487 \cdot 10^3$	$7.01193 \cdot 10^3$
4	$3.4322 \cdot 10^{-1}$	$3.1232 \cdot 10^{-1}$	$7.29205 \cdot 10^3$	$7.32899 \cdot 10^3$
5	$3.3066 \cdot 10^{-1}$	$3.1339 \cdot 10^{-1}$	$6.32234 \cdot 10^3$	$7.41997 \cdot 10^3$
6	$3.1918 \cdot 10^{-1}$	$3.0756 \cdot 10^{-1}$	$5.42293 \cdot 10^3$	$6.85419 \cdot 10^3$

6.2. Monopitch roof benchmark

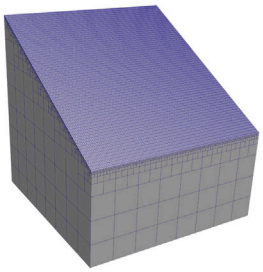
The matrix A of this benchmark study is defined as $A_{ij} := \frac{1}{2}(j-1)+65$ for $i,j \in \{1,\dots,129\}$. It represents a DEM describing the shape of a monopitch roof. The Figs. 6(a) and 6(b) show the initial tetrahedron mesh for this shape and its uniform refinement of level $k = 3$ to be used in ELMER. The initial mesh size h_0^{ELM} is 192. The set $\mathcal{M}^-$ with refinements in the interior of $\mathbb{B}$ is depicted in Fig. 6.



(a) Initial mesh for ELMER.

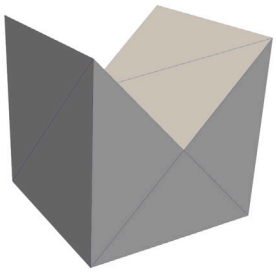


(b) Uniform refinement of the initial mesh of level $k = 3$.

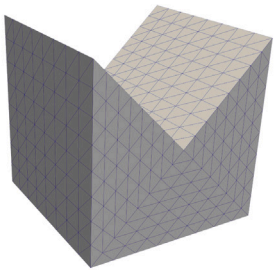


(c) $\mathcal{M}^-$ for refinement level $k = 3$.

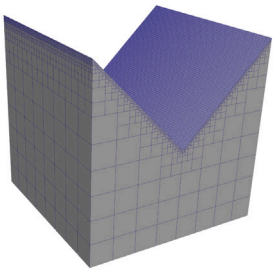
Fig. 6. Refinements in the monopitch roof benchmark.



(a) Initial tetrahedron mesh for ELMER.



(b) Uniform refinements of the initial tetrahedron mesh of level $k = 3$.



(c) $\mathcal{M}^-$ for the refinement level $k = 3$.

Fig. 7. Refinements in the notch benchmark.

Table 8
$d_k^\star$ and $a_k^\star$ in the notch benchmark.

k	d_k^{FCM}	d_k^{ELM}	a_k^{FCM}	a_k^{ELM}
0	$2.4 \cdot 10^1$	$3 \cdot 10^1$	$2.8150735686567437 \cdot 10^7$	$2.9662732807855725 \cdot 10^7$
1	$8.1 \cdot 10^1$	$1.08 \cdot 10^2$	$3.3596114887416966 \cdot 10^7$	$3.4428760019778185 \cdot 10^7$
2	$3.75 \cdot 10^2$	$5.25 \cdot 10^2$	$3.6066306464144714 \cdot 10^7$	$3.6268369580324993 \cdot 10^7$
3	$2.079 \cdot 10^3$	$3.159 \cdot 10^3$	$3.687956101188506 \cdot 10^7$	$3.6921761389167011 \cdot 10^7$
4	$1.2903 \cdot 10^4$	$2.1675 \cdot 10^4$	$3.71286716521469 \cdot 10^7$	$3.7139875821009263 \cdot 10^7$
5	$8.8407 \cdot 10^4$	$1.60083 \cdot 10^5$	$3.720565194099038 \cdot 10^7$	$3.7210895600907773 \cdot 10^7$
6	$6.48375 \cdot 10^5$	$1.229475 \cdot 10^6$	$3.723134392786253 \cdot 10^7$	$3.7234334067791574 \cdot 10^7$
7	$4.952439 \cdot 10^6$	$9.635139 \cdot 10^6$	$3.724084395367293 \cdot 10^7$	$3.7242469399890229 \cdot 10^7$
$a_{7,7}^\star$			$3.724804253264429 \cdot 10^7$	$3.724792073075803 \cdot 10^7$

In Table 5 the number $d_k^\star$ of the degrees of freedom and the quantity $a_k^\star$ are tabulated. The table shows that the DEM-FCM and ELMER differ in $d_k^\star$ but yield very similar values for $a_k^\star$. The Tables 6 and 7 show that the convergence orders $\kappa_k^\star$ and $\delta_k^\star$ are similar and close to 1 and $1/3$, respectively, which indicates that the solution u is sufficiently regular. The convergence orders for the DEM-FCM are slightly larger than those of ELMER. The constants $C_k^\star$ and $D_k^\star$ are smaller for the DEM-FCM than those of ELMER, when $d_k^\star$ is large. Both observations suggest that the DEM-FCM appears to be more efficient than ELMER in the sense that the DEM-FCM requires a smaller number of degrees of freedom than ELMER to achieve a prescribed error. This can also be seen in Fig. 5(b) showing the error $e_k^\star$ versus $d_k^\star$ in a log–log diagram. Again, both graphs become a straight line with the slope $-\delta_k^\star$ and the vertical intercept $D_k^\star$. But, the graph for the DEM-FCM is clearly below the graph for ELMER. The improved convergence behavior of the DEM-FCM could be attributed to the slightly better convergence properties of the trilinear shape functions on (axis-parallel) hexahedrons (compared to the use of linear shape functions on tetrahedrons in ELMER).

6.3. Notch benchmark

In this benchmark study the matrix A is defined as

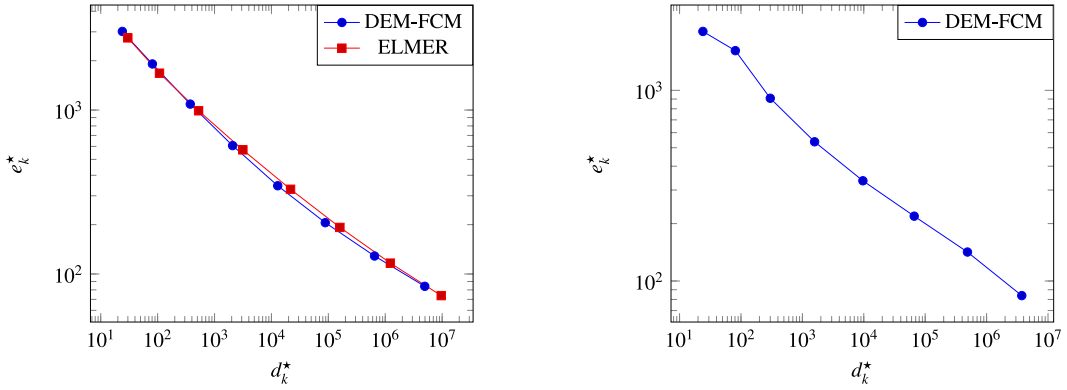
$$A_{ij} := \begin{cases} 130 - i, & i \in \{1, \dots, 65\}, \\ i, & i \in \{65, \dots, 129\} \end{cases}$$

Table 9
$\kappa_k^\star$ and $C_k^\star$ in the notch benchmark.

k	κ_k^{FCM}	κ_k^{ELM}	C_k^{FCM}	C_k^{ELM}
0	$6.58488 \cdot 10^{-1}$	$7.1404 \cdot 10^{-1}$	$8.60561 \cdot 10^{1}$	$6.72785 \cdot 10^{1}$
1	$8.14 \cdot 10^{-1}$	$7.626 \cdot 10^{-1}$	$4.1382 \cdot 10^{1}$	$5.40604 \cdot 10^{1}$
2	$8.4085 \cdot 10^{-1}$	$7.9333 \cdot 10^{-1}$	$3.71581 \cdot 10^{1}$	$4.8087 \cdot 10^{1}$
3	$8.1364 \cdot 10^{-1}$	$7.9706 \cdot 10^{-1}$	$4.06729 \cdot 10^{1}$	$4.75318 \cdot 10^{1}$
4	$7.4823 \cdot 10^{-1}$	$7.726 \cdot 10^{-1}$	$4.83164 \cdot 10^{1}$	$5.04285 \cdot 10^{1}$
5	$6.75327 \cdot 10^{-1}$	$7.2325 \cdot 10^{-1}$	$5.56423 \cdot 10^{1}$	$5.49595 \cdot 10^{1}$
6	$6.14003 \cdot 10^{-1}$	$6.58965 \cdot 10^{-1}$	$6.00443 \cdot 10^{1}$	$5.87582 \cdot 10^{1}$

Table 10
$\delta_k^\star$ and $D_k^\star$ in the notch benchmark.

k	δ_k^{FCM}	δ_k^{ELM}	D_k^{FCM}	D_k^{ELM}
0	$3.75195 \cdot 10^{-1}$	$3.86365 \cdot 10^{-1}$	$9.93728 \cdot 10^{3}$	$1.02472 \cdot 10^{4}$
1	$3.6813 \cdot 10^{-1}$	$3.3424 \cdot 10^{-1}$	$9.63301 \cdot 10^{3}$	$8.02919 \cdot 10^{3}$
2	$3.4026 \cdot 10^{-1}$	$3.0638 \cdot 10^{-1}$	$8.16633 \cdot 10^{3}$	$6.74348 \cdot 10^{3}$
3	$3.089 \cdot 10^{-1}$	$2.8685 \cdot 10^{-1}$	$6.42652 \cdot 10^{3}$	$5.76131 \cdot 10^{3}$
4	$2.6945 \cdot 10^{-1}$	$2.678 \cdot 10^{-1}$	$4.42416 \cdot 10^{3}$	$4.76341 \cdot 10^{3}$
5	$2.34914 \cdot 10^{-1}$	$2.459 \cdot 10^{-1}$	$2.98514 \cdot 10^{3}$	$3.66366 \cdot 10^{3}$
6	$2.09302 \cdot 10^{-1}$	$2.21826 \cdot 10^{-1}$	$2.11892 \cdot 10^{3}$	$2.61442 \cdot 10^{3}$



(a) The error $e_k^\star$ versus the number $d_k^\star$ of degrees of freedom in the notch benchmark.

(b) The error $e_k^\star$ versus the number $d_k^\star$ of degrees of freedom in the sphere benchmark.

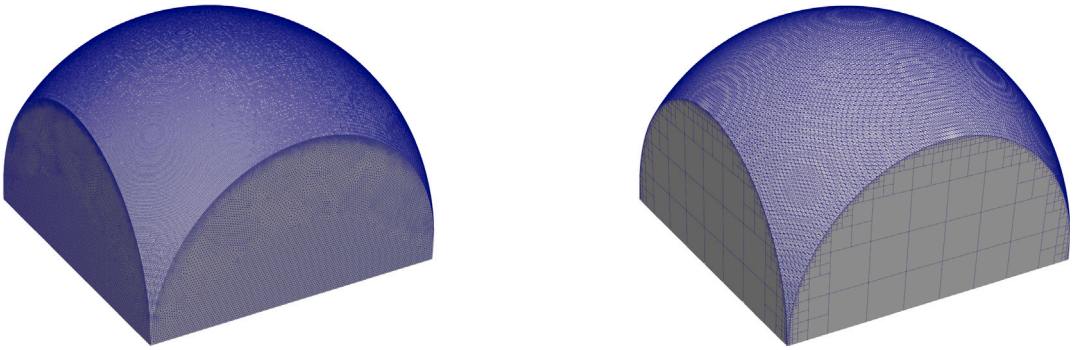
Fig. 8. The error behavior in the notch and sphere benchmark.

with $j \in \{1, \dots, 129\}$. The corresponding DEM describes the shape of a notch. The Figs. 7(a) and 7(b) show the initial tetrahedron mesh for this shape and its uniform refinement of level $k = 3$ to be used in ELMER. The initial mesh size h_0^{ELM} is $128 \cdot 2^{1/2}$. The set $\mathcal{M}^-$ with refinements in the interior of $\mathbb{B}$ is depicted in Fig. 7.

The number $d_k^\star$ of the degrees of freedom and the quantity $a_k^\star$ are tabulated in Table 8, which shows that the DEM-FCM and ELMER are different with respect to $d_k^\star$ but yield again very similar values for $a_k^\star$. The Tables 9 and 10 show that the convergence orders $\kappa_k^\star$ and $\delta_k^\star$ are similar, but significantly smaller than 1 and 1/3, respectively. This indicates that the solution u has a reduced regularity. The differences of the convergence orders and the constants $C_k^\star$ and $D_k^\star$ (also shown in these tables) are rather small and not as pronounced as in the monopitch roof benchmark of Section 6.2. Moreover, they seem to slightly decrease. These observations can also be seen in Fig. 8(a) which shows the error $e_k^\star$ versus $d_k^\star$ in a log–log diagram. The graph appears to be slightly curved and not a truly straight line. Furthermore, we find that the curve for the DEM-FCM is slightly below the curve for ELMER, but it is not entirely clear whether the DEM-FCM can be seen to be more efficient than ELMER as in the monopitch roof benchmark.

6.4. Sphere benchmark

In the final benchmark study the matrix A is given by (1) with $r := 64$ and represents a DEM describing the surface of a sphere, see Fig. 9(b). The number N_k^{FCM} of the degrees of freedom and the quantity a_k^{FCM} are tabulated in Table 11. In this benchmark, results for ELMER in dependence of k are not available because it is difficult (or even impossible) to construct tetrahedron meshes with tetrahedrons that have a (nearly) comparable size as the hexahedrons of $\hat{\mathcal{G}}$ in the DEM-FCM and that exactly represent the surface $\mathbb{S}$ (in contrast to the previous benchmarks). Nevertheless, we use ELMER to compute a reference value that is independent



(a) Tetrahedron mesh for ELMER. (b) $\mathcal{M}^-$ for the refinement level $k = 3$.

Fig. 9. Refinements in the sphere benchmark.

Table 11
N_k^{FCM} and α_k^{FCM} in the sphere benchmark.

k	d_k^{FCM}	a_k^{FCM}
0	$2.4 \cdot 10^1$	$1.2124993932175733 \cdot 10^7$
1	$8.1 \cdot 10^1$	$1.3663774330216378 \cdot 10^7$
2	$3 \cdot 10^2$	$1.5446305427900927 \cdot 10^7$
3	$1.581 \cdot 10^3$	$1.5983639852649683 \cdot 10^7$
4	$9.675 \cdot 10^3$	$1.6159688283822805 \cdot 10^7$
5	$6.6264 \cdot 10^4$	$1.622435483888038 \cdot 10^7$
6	$4.88025 \cdot 10^5$	$1.6252108100070663 \cdot 10^7$
7	$3.739896 \cdot 10^6$	$1.6265212482635263 \cdot 10^7$
$a_{7,7}^{\mathrm{FCM}}$		$1.6278009010141218 \cdot 10^7$

Table 12
κ_k^{FCM}, C_k^{FCM}, δ_k^{FCM} and D_k^{FCM} in the sphere benchmark.

k	κ_k^{FCM}	C_k^{FCM}	δ_k^{FCM}	D_k^{FCM}
0	$3.34492 \cdot 10^{-1}$	$3.34432 \cdot 10^2$	$1.90588 \cdot 10^{-1}$	$3.73161 \cdot 10^3$
1	$8.2965 \cdot 10^{-1}$	$3.24904 \cdot 10^1$	$4.392 \cdot 10^{-1}$	$1.11261 \cdot 10^4$
2	$7.5858 \cdot 10^{-1}$	$4.32236 \cdot 10^1$	$3.1633 \cdot 10^{-1}$	$5.52097 \cdot 10^3$
3	$6.79309 \cdot 10^{-1}$	$5.62539 \cdot 10^1$	$2.599 \cdot 10^{-1}$	$3.64337 \cdot 10^3$
4	$6.16557 \cdot 10^{-1}$	$6.63358 \cdot 10^1$	$2.22083 \cdot 10^{-1}$	$2.57501 \cdot 10^3$
5	$6.2515 \cdot 10^{-1}$	$6.52653 \cdot 10^1$	$2.16998 \cdot 10^{-1}$	$2.43378 \cdot 10^3$
6	$7.5923 \cdot 10^{-1}$	$5.5247 \cdot 10^1$	$2.584 \cdot 10^{-1}$	$4.186 \cdot 10^3$

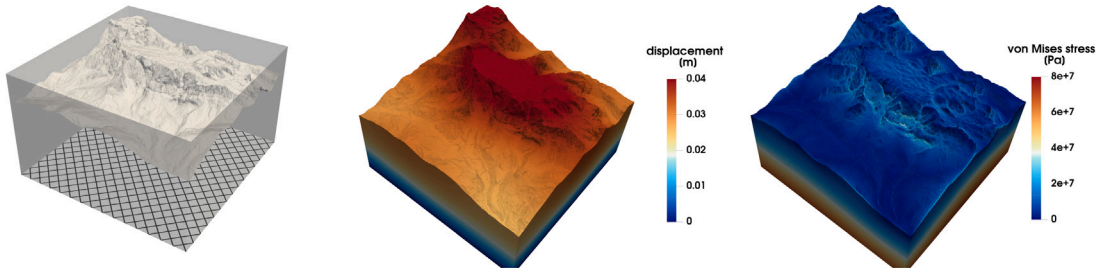
of the DEM-FCM (which is, for instance, not the case if we use $a_{7,7}^{\mathrm{FCM}}$). The reference value is $a_{\mathrm{ref}} := 1.6272241867850427 \cdot 10^7$ and is computed by using a tetrahedron mesh generated by SMesh [47], see Fig. 9(a). The number of degrees of freedom for computing this value is 6 248 772, which is significantly larger than d_7^{FCM}.

The Table 12 lists the convergence orders κ_k^{FCM} and δ_k^{FCM} as well as the constants C_k^{FCM} and D_k^{FCM}. The convergence orders are significantly smaller than 1 and $1/3$, respectively, indicating that the solution u may have a reduced regularity. Fig. 8(b) shows the error $e_k^\star$ versus $d_k^\star$ in a log–log diagram. The graph appears to be a straight line (at least on average).

7. Application: Hochkönig topography

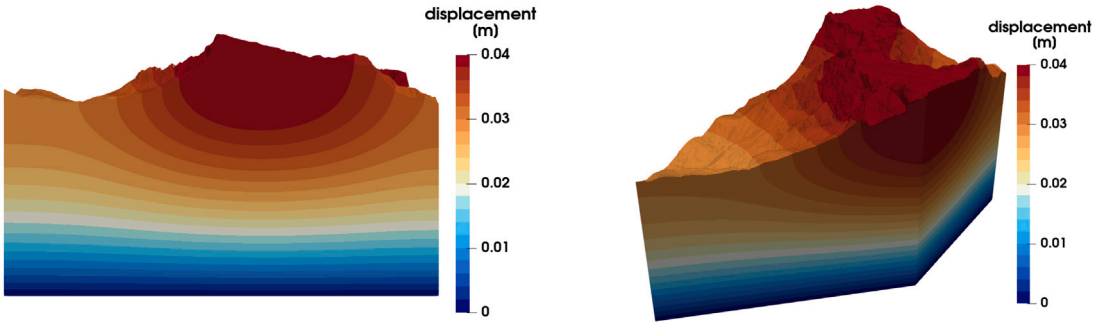
To demonstrate the applicability of the DEM-FCM on realistic geological data, we study the Hochkönig Massif (Austrian UNESCO Global Geopark *Erz der Alpen*, Northern Calcareous Alps). For this purpose, we use a DEM provided by the Infrastructure for Spatial Information in the European Community – INSPIRE, cf. [44]. The DEM has a size of $n = m = 1001$ with 10 m horizontal resolution, i.e. the domain extent is 10 km.

An essential problem in geological applications is the evaluation of rock stability. When shear stresses exceed a certain threshold (e.g. rock strength), the rock fails and can eventually trigger a landslide [48,49]. The topography of the study area consists of a high plateau and is bounded by almost vertical cirque walls and ridges, with local relief varying between a few hundred meters and one kilometer. These topographic features and their high variability make this specific topography prone to high stresses and thus potentially sites of landslides. In the following we demonstrate that the DEM-FCM can directly be utilized for high-resolution DEM data, enabling elastic stress analysis of topographies for use in assessing landslide hazards.



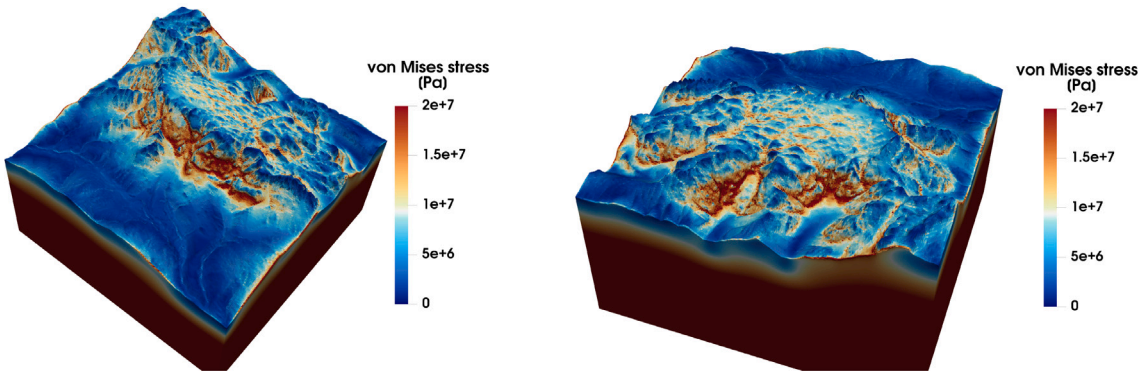
(a) Γ_D^a in hatched lines, Γ_D in grey, Γ_N in light grey. (b) Displacement field. (c) von Mises stress

Fig. 10. (a) Boundary parts, (b) displacement field and (c) von Mises stress in the Hochkönig massif.



(a) Side view (single cut) of displacement magnitude with 20 uniform bins. (b) Oblique/diagonal view (double cut) of displacement magnitude with 20 uniform bins.

Fig. 11. Cross-sectional visualizations of the computed displacement field (meters), shown with discrete binning to improve contour readability.

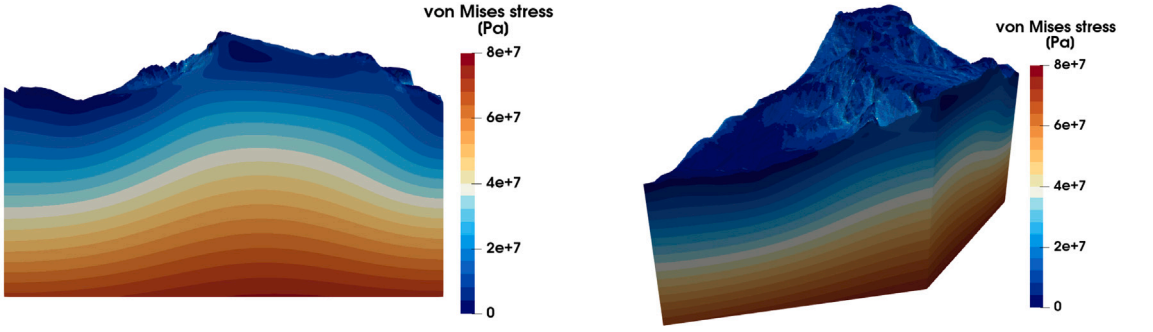


(a) Diagonal view (color scale adjusted to 2×10^7 Pa). (b) Rear/oblique view emphasizing ridge–valley localization.

Fig. 12. von Mises stress (Pa) computed with the DEM-FCM on the Hochkönig DEM. Color scaling is chosen to reveal localized high-stress regions.

The rock mass is modeled as homogeneous Dachstein limestone characterized by Young's modulus $E := 68.24 \cdot 10^9$ Pa, Poisson ratio $\nu := 0.30$ and density $\rho = 2660\ \mathrm{kg\,m^{-3}}$ (specifying the Lamé constants as in (29) and the volume force as in (30)), cf. [50–52]. The surface force g is assumed to be 0. We use a single, spatially invariant set of material parameters, since detailed three-dimensional information about the internal distribution of lithologies and their mechanical properties is generally not available (e.g., based on measurements) at the scale of entire mountain massifs. Although variations in elastic properties and strength can locally influence the magnitude and orientation of stresses, the large-scale patterns of gravitational (dead-load) stresses are primarily determined by topography and geometry and less by moderate differences in material properties [53].

Applying the DEM-FCM we use a regular grid $\hat{\mathcal{G}}$ of $\mathbb{B} := [1, 1001]^3$ consisting of cubes of the same size with length 4 in each direction and employ Algorithm 1 for generating a DEM grid $\mathcal{G}_{\hat{H}}$ for each $\hat{H} \in \mathcal{F}$, where $\mathcal{F}$ is generated as $\hat{\mathcal{G}}^- \cup \hat{\mathcal{G}}^*$, cf. Remark 3.



(a) Diagonal cross-section with 20 bins (enhancing gradient clarity).

(b) Oblique cross-section emphasizing localization at ridge–valley interfaces.

Fig. 13. Discrete-bin visualization of von Mises stress (Pa) to emphasize stress concentration gradients.

This leads to 23 709 651 degrees of freedom. The mesh $\mathcal{M}^-$ consists of 8 810 882 hexahedrons and 21 665 556 tetrahedrons in total. The resulting displacement and von Mises stress fields are depicted in Figs. 10(b), 10(c) and 11–13.

The cross-sectional displacement fields in Fig. 11 show a clear spatial pattern, with the largest displacements concentrated beneath the mountain massif in the center of the area and decreasing toward the boundaries of the area. This behavior is consistent with classical linear elastic solutions for dead loads [48,54]. The diagonal cut in Fig. 11(b) additionally shows that deformation propagates along ridgelines and into adjacent slopes, while lateral edges remain comparatively immobile due to the applied boundary conditions.

Fig. 12 displays two principal features of the von Mises stress field. First, the largest stresses are centered beneath and immediately adjacent to the most prominent topographic load (the massif) and decrease toward the boundaries. Second, narrow bands of strongly elevated stress occur at geometric transitions, notably where steep flanks meet valley floors or adjoining ridges. This localization reflects the fact that even subtle curvature variations can produce amplified stress in precisely those zones where convexities or concavities interrupt otherwise smooth slopes, which tends to guide subsequent tearing and detachment [4]. The boundaries of the domain also show an increased stress, as the rigid boundary conditions at the sides (with fixed xy-displacements and free z-displacements) restrict lateral deformation and therefore lead to concentrated stress near the domain edges. To make these effects clearer, Fig. 13 shows the stress field with the same colorbar as in Fig. 10(c) rendered with 20 uniform discrete bins for better visualization of the stress transitions.

We emphasize that the stress localization observed here is mechanically significant even in an elastic calculation. Regions of persistent tensile or elevated shear stress are known to be potential sites for microcracks and progressive damage under environmental or cyclic loading, thereby acting as precursors to fracture growth and eventual slope instability [4,55]. Hence, while the elastic von Mises field is not a direct failure criterion for brittle materials, it provides a screening metric that identifies structurally sensitive zones that require more detailed fracture or damage analysis. To relate the computed linear elastic stress fields to the onset of failure, it is useful to derive scalar measures that reflect how close the local stress state is to commonly used rock-failure criteria [4]. A straightforward quantity in this context is the maximum shear stress, expressed as half the difference between the largest and smallest principal stresses. This measure captures the intensity of deviatoric stresses and provides an indicator of the potential for shear failure, independent of assumptions about specific failure planes [43].

This case study highlights the applicability of the DEM-FCM to geological topography. Unlike FEM, which typically requires boundary-conforming meshes and extensive preprocessing to capture complex terrain, DEM-FCM operates directly on the DEM without the need for remeshing. This direct DEM-to-mesh workflow substantially reduces preprocessing overhead and allows for a direct usage of large-scale terrain data. The method works efficiently, even for problems with a large number of degrees of freedom (as demonstrated here) making it suitable for practical applications where realistic resolution and domain size are needed.

8. Summary and further discussion

The paper presents a fictitious domain approach based on the finite cell method (FCM), that is specifically adapted to a digital elevation model (DEM). The DEM used in this paper describes a surface topography in which elevation values are assigned to the data points of a square grid with equidistant points. The developed method (called DEM-FCM) enables the computation of displacements and stress distributions in linear elasticity in domains with complex surfaces. In particular, the DEM-FCM is ideally suited for geological applications, as demonstrated by computations of stress distributions in the Hochkönig massif (Eastern Alps). The method uses a specific quadrature mesh to compute weighted integrals of the variational formulation of the linear elastic problem. The mesh consists of hexahedrons and tetrahedrons (DEM-fitted tetrahedron decompositions) that exactly represent a triangle surface mesh (DEM surface) which linearly interpolates the DEM data set. The basic idea behind the generation of DEM-fitted tetrahedron decompositions is to decompose all hexahedrons with a non-empty intersection with the DEM surface into six tetrahedrons, where the decomposition is done along the diagonal that already defines the DEM surface. It is proven that the relative interior of the edges of these tetrahedrons has at most one intersection point with the triangles of the DEM surface. This means that

they can be subdivided into tetrahedrons using only four decomposition patterns. One of the main advantages of the proposed DEM-FCM is that DEM-fitted tetrahedron decompositions are only generated for each single hexahedron of the FEM grid during the loop of the assembling process, so that the quadrature mesh used for numerical integration does not need to be stored in its entirety. This is particularly important because of the potentially very large data volume given by the DEM. Essentially, the proposed DEM-FCM relies on an efficient reconstruction of a surface from point-based geometry data by creating a quadrature mesh that is used within the FCM framework, with the advantage that this reconstruction is exact. A direct way (without surface reconstruction) to incorporate complex point-based geometry data into the framework of the FCM or other methods is provided by point-cloud-based approaches [56,57], in which additional orientation information is required for each point.

The FEM approach underlying the DEM-FCM is based on trilinear shape functions on hexahedrons. In general, the use of hexahedrons in finite element methods is advantageous because it allows, for instance, the application of higher-order tensor product shape functions with high accuracy and other beneficial properties. The use of this type of shape function is very common in the FCM and will also be examined in a subsequent paper in connection with the DEM-FCM approach. Fictitious domain approaches (as the DEM-FCM) offer a natural concept for the use of hexahedrons. It should be noted that other approaches also allow for hexahedrons over large regions in domains with complex boundaries or transitions, for example by combining hexahedrons with tetrahedrons and pyramids in hybrid meshes (in conjunction with appropriate shape functions) [58,59].

Several numerical experiments based on benchmark problems with simple geometries demonstrate the convergence properties of the DEM-FCM and confirm convergence results similar to those obtained using a standard lowest-order finite element method (implemented in the FEM software ELMER). The use of the DEM-FCM in the geological application allows for computational simulations which are in accordance with the current state of geological knowledge and show that the DEM-FCM can be applied as an efficient tool for stress simulations based on digital elevation models.

CRediT authorship contribution statement

Viktor Haunsperger: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jörg Robl:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Resources, Project administration, Investigation, Funding acquisition, Data curation, Conceptualization. **Andreas Schröder:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Resources, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors gratefully acknowledge the support by the Earth System Sciences programme of the Austrian Academy of Sciences under the project ‘Moving mountains - landslides as geosystem services in Austrian geoparks’ (ESS22-24 - MOVEMONT).

Data availability

Data will be made available on request.

References

[1] A. Verruijt, Computational Geomechanics, Kluwer; Theory and Applications of Transport in Porous Media, vol. 7, p. 1995.
[2] O.C. Zienkiewicz, A.H.C. Chan, M. Pastor, B.A. Schrefler, T. Shiomi, Computational Geomechanics with Special Reference To Earthquake Engineering, Wiley, Chichester, 1999.
[3] A.H.C. Chan, M. Pastor, B.A. Schrefler, T. Shiomi, O. Zienkiewicz, Computational Geomechanics Theory and Applications, Wiley, 2020.
[4] S. Moon, J. Perron, S. Martel, W. Holbrook, J. St. Clair, A model of three-dimensional topographic stresses with implications for bedrock fractures, surface processes, and landscape evolution, J. Geophys. Res.: Earth Surf. 122 (4) (2017) 823–846.
[5] R. Verzicco, M.D. de Tullio, F. Viola, An Introduction To Immersed Boundary Methods, in: Camb. Monogr. Appl. Comput. Math., vol. 43, Cambridge University Press, Cambridge, 2026.
[6] R. Glowinski, T.-W. Pan, J. Periaux, A fictitious domain method for dirichlet problem and applications, Comput. Methods Appl. Mech. Engrg. 111 (3–4) (1994) 283–303.
[7] A. Düster, J. Parvizian, Z. Yang, E. Rank, The finite cell method for three-dimensional problems of solid mechanics, Comput. Methods Appl. Mech. Engrg. 197 (45–48) (2008) 3768–3782.
[8] J. Parvizian, A. Düster, E. Rank, Finite cell method : *h*- and *p*-extension for embedded domain problems in solid mechanics, Comput. Mech. 41 (1) (2007) 121–133.
[9] N. Zander, S. Kollmannsberger, M. Ruess, Z. Yosibash, E. Rank, The finite cell method for linear thermoelasticity, Comput. Math. Appl. 64 (11) (2012) 3527–3541.
[10] S. Duczek, S. Liefold, U. Gabbert, The finite and spectral cell methods for smart structure applications: Transient analysis, Acta Mech. 226 (3) (2015) 845–869.

[11] D. Schillinger, M. Ruess, N. Zander, Y. Bazilevs, A. Düster, E. Rank, Small and large deformation analysis with the *p*- and *B*-spline versions of the finite cell method, Comput. Mech. 50 (4) (2012) 445–478.
[12] S. Kollmannsberger, D. D'Angella, E. Rank, W. Garhuom, S. Hubrich, A. Düster, P.D. Stolfo, A. Schröder, Spline- and hp-basis functions of higher differentiability in the finite cell method, GAMM-Mitt. 43 (1) (2020).
[13] M. Ruess, D. Tal, N. Trabelsi, Z. Yosibash, E. Rank, The finite cell method for bone simulations: Verification and validation, Biomech. Model. Mechanobiol. 11 (3–4) (2012) 425–437.
[14] C. Verhoosel, G. van Zwieten, B. van Rietbergen, R. de Borst, Image-based goal-oriented adaptive isogeometric analysis with application to the micro-mechanical modeling of trabecular bone, Comput. Methods Appl. Mech. Engrg. 284 (2015) 138–164.
[15] Z. Yang, S. Kollmannsberger, A. Düster, M. Ruess, E.G. Garcia, R. Burgkart, E. Rank, Non-standard bone simulation: Interactive numerical analysis by computational steering, Comput. Vis. Sci. 14 (5) (2011) 207–216.
[16] A. Abedian, J. Parvizian, A. Düster, E. Rank, The finite cell method for the J_2 flow theory of plasticity, Finite Elem. Anal. Des. 69 (2013) 37–47.
[17] S. Heinze, T. Bleistein, A. Düster, S. Diebels, A. Jung, Experimental and numerical investigation of single pores for identification of effective metal foams properties, ZAMM Z. Angew. Math. Mech. 98 (5) (2018) 682–695.
[18] S. Heinze, M. Joulaian, A. Düster, Numerical homogenization of hybrid metal foams using the finite cell method, Comput. Math. Appl. 70 (7) (2015) 1501–1517.
[19] E. Rank, S. Kollmannsberger, C. Sorger, A. Düster, Shell finite cell method: A high order fictitious domain approach for thin-walled structures, Comput. Methods Appl. Mech. Engrg. 200 (45–46) (2011) 3200–3209.
[20] S. Duczek, M. Joulaian, A. Düster, U. Gabbert, Numerical analysis of lamb waves using the finite and spectral cell methods, Internat. J. Numer. Methods Engrg. 99 (1) (2014) 26–53.
[21] M. Elhaddad, N. Zander, S. Kollmannsberger, A. Shadavakhsh, V. Nübel, E. Rank, Finite cell method: High-order structural dynamics for complex geometries, Int. J. Struct. Stab. Dyn. 15 (7) (2015).
[22] M. Joulaian, S. Duczek, U. Gabbert, A. Düster, Finite and spectral cell method for wave propagation in heterogeneous materials, Comput. Mech. 54 (3) (2014) 661–675.
[23] J. Parvizian, A. Düster, E. Rank, Topology optimization using the finite cell method, Optim. Eng. 13 (1) (2012) 57–78.
[24] M. Dauge, A. Düster, E. Rank, Theoretical and numerical investigation of the finite cell method, J. Sci. Comput. 65 (3) (2015) 1039–1064.
[25] A. Osmers, A. Rademacher, A. Schröder, Goal-Oriented Adaptive Finite Cell Methods, in: Lecture Notes in Computational Science and Engineering 154 LNCSE, 2025, pp. 223–232.
[26] Di Stolfo P., A. Schröder, Error control and adaptivity for the finite cell method, Lect. Notes Appl. Comput. Mech. 98 (2022) 377–403.
[27] Di Stolfo P., A. Schröder, Reliable residual-based error estimation for the finite cell method, J. Sci. Comput. 87 (1) (2021).
[28] P. Di Stolfo, A. Rademacher, A. Schröder, Dual weighted residual error estimation for the finite cell method, J. Numer. Math. 27 (2) (2019) 101–122.
[29] L. Kudela, N. Zander, T. Bog, S. Kollmannsberger, E. Rank, Efficient and accurate numerical quadrature for immersed boundary methods 2 (1) (2015).
[30] L. Kudela, N. Zander, S. Kollmannsberger, E. Rank, Smart octrees: Accurately integrating discontinuous functions in 3d, Comput. Methods Appl. Mech. Engrg. 306 (2016) 406–426.
[31] M. Petö, F. Duvigneau, S. Eisenträger, Enhanced numerical integration scheme based on image-compression techniques: Application to fictitious domain methods, Adv. Model. Simul. Eng. Sci. 7 (1) (2020).
[32] M. Petö, W. Garhuom, F. Duvigneau, S. Eisenträger, A. Düster, D. Juhre, Octree-based integration scheme with merged sub-cells for the finite cell method: Application to non-linear problems in 3D, Comput. Methods Appl. Mech. Engrg. (2022).
[33] A. Abedian, A. Düster, An extension of the finite cell method using boolean operations, Comput. Mech. 59 (5) (2017) 877–886.
[34] M. Petö, S. Eisenträger, F. Duvigneau, D. Juhre, Boolean finite cell method for multi-material problems including local enrichment of the ansatz space, Comput. Mech. 72 (4) (2023) 743–764.
[35] S. Hubrich, Di Stolfo P., L. Kudela, S. Kollmannsberger, E. Rank, A. Schröder, A. Düster, Numerical integration of discontinuous functions: Moment fitting and smart octree, Comput. Mech. 60 (5) (2017) 863–881.
[36] M. Joulaian, S. Hubrich, A. Düster, Numerical integration of discontinuities on arbitrary domains based on moment fitting, Comput. Mech. 57 (6) (2016) 979–999.
[37] M. Meßmer, S. Kollmannsberger, R. Wüchner, K.-U. Bletzinger, Robust numerical integration of embedded solids described in boundary representation, Comput. Methods Appl. Mech. Engrg. 419 (2024) 116670.
[38] W. Garhuom, A. Düster, Non-negative moment fitting quadrature for cut finite elements and cells undergoing large deformations, Comput. Mech. 70 (5) (2022) 1059–1081.
[39] G. Legrain, Non-negative moment fitting quadrature rules for fictitious domain methods, Comput. Math. Appl. 99 (2021) 270–291.
[40] A. Byfut, F. Hellwig, A. Schröder, Marching volume polytopes algorithm, Internat. J. Numer. Methods Engrg. 117 (12) (2019) 1171–1204.
[41] A. Stroud, Some fourth degree integration formulas for simplexes, Math. Comp. 30 (134) (1976) 291–294.
[42] J. Ruokolainen, P. Råback, M. Malinen, T. Zwinger, Elmerfem, zenodo repository of elmer release 9.0, 2023, http://www.elmerfem.org.
[43] V. Haunsperger, J. Robl, A.-L. Argentin, S. Hergarten, M. Mergili, A. Schröder, Stress redistribution following landslides: Insights from 3D stress modeling of mountain topography, 2025, submitted for publication.
[44] inspire.gv.at, INSPIRE Geoportal Österreich, 2025, https://geoportal.inspire.gv.at. (Accessed 23 December 2025).
[45] J.R. Shewchuk, Tetrahedral mesh generation by Delaunay refinement, in: Proceedings of the Annual Symposium on Computational Geometry, 1998, pp. 86–95.
[46] J. Stoer, Einführung in die Numerische Mathematik. I. Unter Berücksichtigung von Vorlesungen von F. L. Bauer, in: Heidelb. Taschenb., Vol. 105, Springer, Berlin, 1972.
[47] SalomePlatform, SMESH — salome mesh module (github repository), 2025, https://github.com/SalomePlatform/smesh. (Accessed 16 December 2025).
[48] H.J. Melosh, Planetary surface processes, Cambridge Planetary Science, Cambridge University Press, 2011, pp. 49–103, Ch. Strength versus gravity.
[49] N. Barton, The shear strength of rock and rock joints, Int. J. Rock Mech. Min. Sci. 13 (9) (1976) 255–279.
[50] N. Gegenhuber, J. Pupos, Rock physics template from laboratory data for carbonates, J. Appl. Geophys. 114 (2015) 12–18.
[51] N. Gegenhuber, Application of Gassmann's equation for laboratory data from carbonates from Austria, Austrian J. Earth Sci. 108 (2015) 239–244.
[52] O. Molina, V. Vilarrasa, M. Zeidouni, Geologic carbon storage for shale gas recovery, Energy Procedia 114 (2017) 5748–5760.
[53] D.J. Miller, T. Dunne, Topographic perturbations of regional stresses and consequent bedrock fracturing, J. Geophys. Res.: Solid Earth 101 (B11) (1996) 25523–25536.
[54] P. Molnar, Interactions among topographically induced elastic stress, static fatigue, and valley incision, J. Geophys. Res.: Earth Surf. 109 (F2) (2004).
[55] K. Leith, J.R. Moore, F. Amann, S. Loew, In situ stress control on microcrack generation and macroscopic extensional fracture in exhuming bedrock, J. Geophys. Res.: Solid Earth 119 (1) (2014) 594–615.
[56] L. Kudela, S. Kollmannsberger, U. Almac, E. Rank, Direct structural analysis of domains defined by point clouds, Comput. Methods Appl. Mech. Engrg. 358 (2020).
[57] J. Zhang, S. Eisenträger, Y. Zhan, A. Saputra, C. Song, Direct point-cloud-based numerical analysis using octree meshes, Comput. Struct. 289 (2023).
[58] K.T. Danielson, M.D. Adley, Five node pyramid elements for explicit time integration in nonlinear solid dynamics, Finite Elem. Anal. Des. 141 (2018) 37–54.
[59] R.S. Browning, K.T. Danielson, D.L. Littlefield, Second-Order Pyramid Element Formulations Suitable for Lumped-Mass Explicit Methods in Nonlinear Solid Mechanics, vol. 405, 2023.