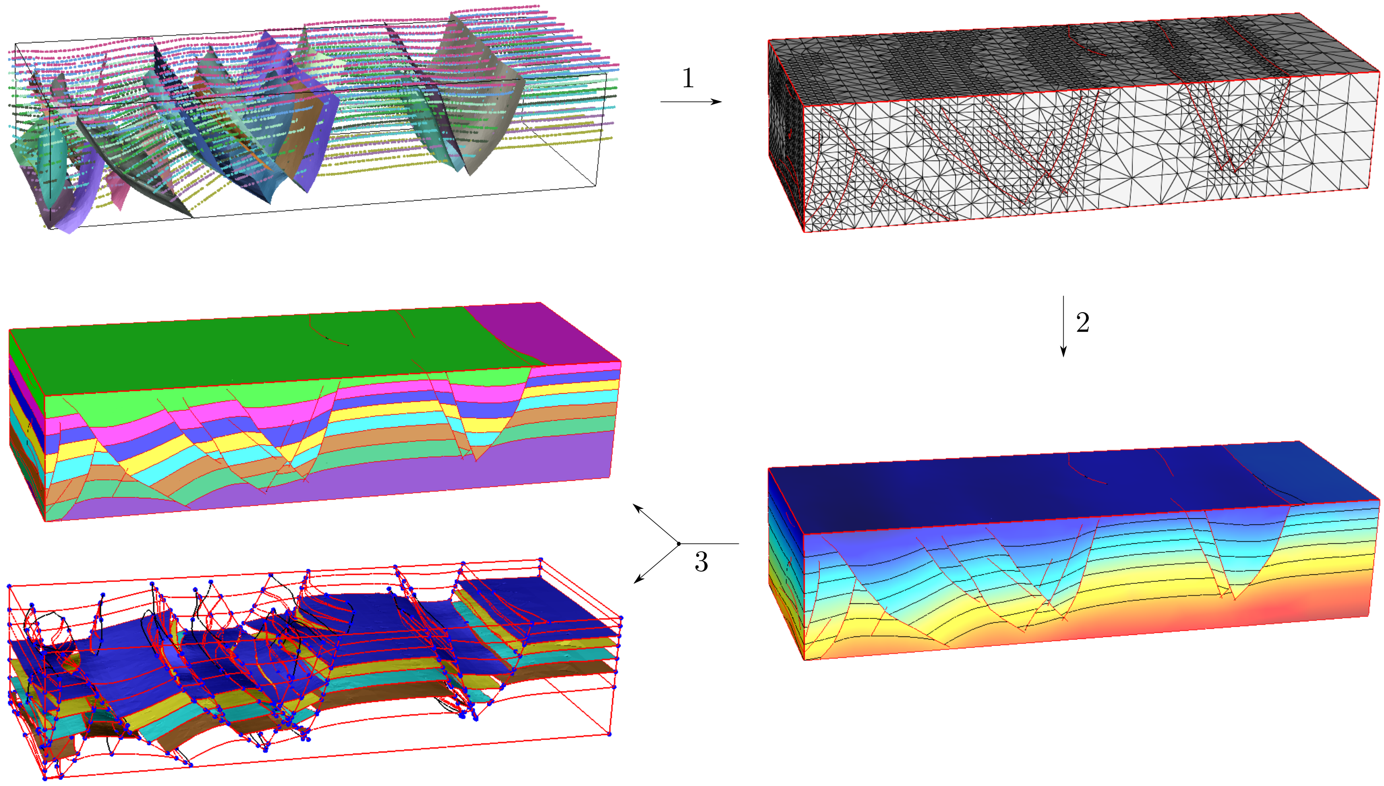
Finite Element Implicit 3D Subsurface Structural Modeling

Modeste Irakarama, Morgan Thierry-Coudon, Mustapha Zakari, Guillaume Caumon

Graphical Abstract

Finite Element Implicit 3D Subsurface Structural Modeling[⋆]

Modeste Irakarama, Morgan Thierry-Coudon, Mustapha Zakari, Guillaume Caumon



The input data (initial CT data courtesy of IFPEN and C&C Reservoirs) is used to create a mesh that is discontinuous across fault surfaces (1), the resulting mesh is then used for finite element implicit stratigraphic modeling (2), from which a sealed subsurface structural model is extracted (3) as the final output.

Highlights

Finite Element Implicit 3D Subsurface Structural Modeling

Modeste Irakarama, Morgan Thierry-Coudon, Mustapha Zakari, Guillaume Caumon

- A finite element discretization of the Laplacian without imposing any boundary condition
- A finite element discretization of the Hessian without imposing any boundary condition
- Building 3D subsurface numerical models using finite elements
- The Laplacian may not be as adequate for interpolation on tetrahedra as it is on triangles

Finite Element Implicit 3D Subsurface Structural Modeling

Modeste Irakarama[a], Morgan Thierry-Coudon[b], Mustapha Zakari[b], Guillaume Caumon[b]

[a]*Independent research, Antony, 92160, France*
[b]*RING, GeoRessources - ENSG, Universite de Lorraine / CNRS, Nancy, 54000, France*

Abstract

We introduce a method for 3D implicit geological structural modeling from sparse sample points, where several conformable geological surfaces are represented by one single scalar field. Laplacian and Hessian regularization energies are discretized on a tetrahedral mesh using finite elements. This scheme is believed to offer some geometrical flexibility as it is readily implemented on both structured and unstructured grids. While implicit modeling on unstructured grids is not new, methods based on finite elements have received little attention. The finite element method is routinely used to solve boundary value problems. However, because boundary conditions are typically unknown in implicit subsurface structural modeling, the traditional finite element method requires some adjustments. To this end, we present boundary free discretizations of the Laplacian and Hessian energies that do not assume vanishing Neumann boundary conditions, thereby eliminating the boundary artifacts usually associated with that assumption. Furthermore, we argue that while an appropriate discretization of the Laplacian can be used to minimize the curvature of a function on triangulated meshes, it may fail to do so on tetrahedral meshes.

Keywords: Subsurface modeling, implicit modeling, data interpolation, finite elements, Laplacian energy, Hessian energy

Introduction

The 3D numerical representation of subsurface geological structures is typically performed from borehole data and seismic image interpretation points sampling the interfaces between different rock materials. Geological structural modeling refers to the creation of surfaces and layers from these sparse sample points, for instance horizons (which have a constant geological age) and faults (finite surfaces which shift rock units tangentially in response to tectonic forces)[see 1, for a recent review]. In this paper, we consider the implicit geological structural modeling problem [2, 3, 4, 5, 6, 7, 8, 9, 10]. A typical workflow is illustrated in Figure 1, where the input data comes from the analogue model examined by [11, 12]. This problem is similar to implicit surface reconstruction techniques [13], but it very often represents several non-intersecting horizons with one single scalar field, and it additionally needs to consider multiple discontinuities (faults) within the modeling domain. Implicit structural modeling has traditionally been separated into two main classes [7, 1]: (1) mesh-free methods [2, 3, 14, 6, 15, 16, 17, 18] , and (2) mesh-based methods [4, 19, 5, 7, 8, 20, 21, 22]. We also acknowledge an increasing interest in methods based on machine learning [23]. The method presented here is a mesh-based method, which starts by generating a mesh conforming to discontinuities such as faults [see 24, 25, 26, 27, 28, for suitable mesh generation techniques]. The principle of mesh-based methods is to use available data to define structural constraints that are discretized on a discrete computational domain, and then assemble those constraints into a system of linear equations, which is supplemented with an additional linear system of smoothing regularization constraints [29, 30, 31]. In particular, the problem can be stated as that of finding a scalar function $\phi(\mathbf{x})$ that satisfies, in a least-squares sense, a system of linear equations of the form

$$\begin{bmatrix}\bar{\bar{\mathbf{D}}}\\ \bar{\bar{\mathbf{R}}}\end{bmatrix}\left[\bar{\Phi}\right]=\begin{bmatrix}\bar{\mathbf{f}}\\ \bar{\mathbf{0}}\end{bmatrix}, \tag{1}$$

where the *data system*

$$\bar{\bar{\mathbf{D}}}\bar{\Phi}=\bar{\mathbf{f}} \tag{2}$$

is assembled from geological interpretation data collected from the field and/or from seismic images. Geological data are often sparse; as a result, the data system is usually undetermined and has to be regularized by the *regularization system*

$$\bar{\bar{\mathbf{R}}}\bar{\Phi}=\bar{\mathbf{0}}. \tag{3}$$

We only focus on the regularization system in this paper.

We examine two regularization operators based on a boundary free finite element discretization of the Laplacian: the first one will discretize the Laplacian

$$\mathscr{R}_1\phi(\mathbf{x})=\partial_x^2\phi(\mathbf{x})+\partial_y^2\phi(\mathbf{x})+\partial_z^2\phi(\mathbf{x}), \tag{4}$$

⋆Published in Computer-Aided Design, Vol. 149 (August 2022), 103267.doi:10.1016/j.cad.2022.103267

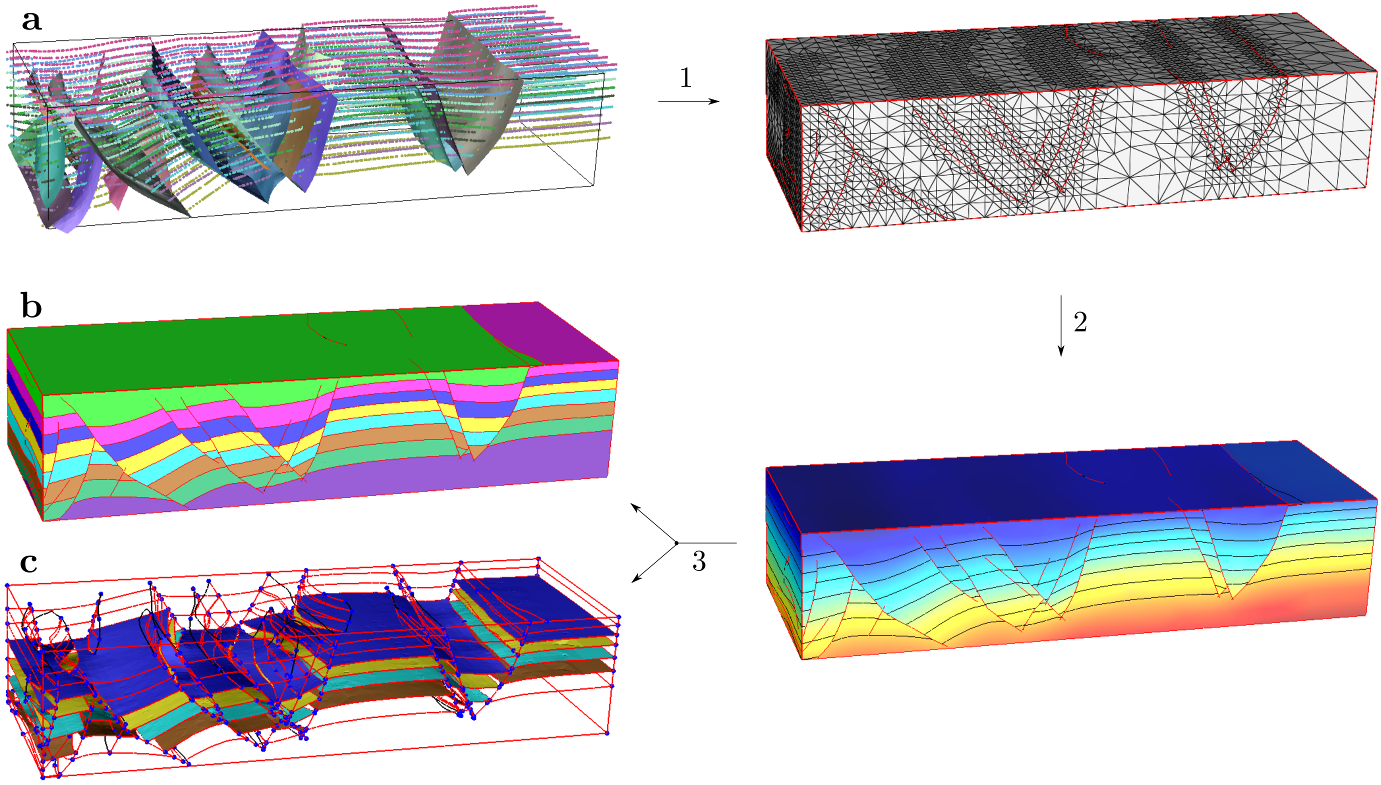


Figure 1: Implicit subsurface structural modeling. (**a**) The input data (point sets from 8 horizons and 27 fault surfaces, initial CT data courtesy of IFPEN and C&C Reservoirs) is used to create a mesh that is discontinuous across fault surfaces (1), the resulting mesh is then used for finite element implicit stratigraphic modeling (2), from which a sealed structural model is extracted (3) as the final output (**b**: layers view, **c**: surfaces view).

and the second one will discretize each component of the Laplacian independently

$$\mathscr{R}_3\phi(\mathbf{x}) = \begin{cases} \partial_x^2\phi(\mathbf{x}) \\ \partial_y^2\phi(\mathbf{x}) \\ \partial_z^2\phi(\mathbf{x}) \end{cases}. \tag{5}$$

We also examine a Hessian based regularization operator, in which we discretize each component of the Hessian matrix independently

$$\mathscr{R}_6\phi(\mathbf{x}) = \begin{cases} \partial_x^2\phi(\mathbf{x}) \\ \partial_y^2\phi(\mathbf{x}) \\ \partial_z^2\phi(\mathbf{x}) \\ \sqrt{2}\partial_{xy}^2\phi(\mathbf{x}) \\ \sqrt{2}\partial_{yz}^2\phi(\mathbf{x}) \\ \sqrt{2}\partial_{zx}^2\phi(\mathbf{x}) \end{cases}. \tag{6}$$

The index i in the notation $\mathscr{R}_i$ indicates the number of independent equations in the regularization operator at each point. In all three cases, the challenge will be to propose a discretization that does not impose any boundary constraints. These regularization operators are intended to measure the curvature of the function $\phi(\mathbf{x})$. In our application, the goal is to find a function $\phi(\mathbf{x})$ that interpolates data and yet has minimal curvature. Minimization of curvature is achieved by imposing Equation 3 where the matrix $\bar{\bar{\mathbf{R}}}$ is assembled by discretizing one of the regularization operators above. Following [20, 22], we consider the function $\phi(\mathbf{x})$ to have zero curvature at the point $\mathbf{x}$ if, by definition, the second directional derivative $\partial_k^2\phi(\mathbf{x})$ (and hence the curvature) along all directions $\mathbf{d}_k$ vanishes at that point. In particular, given the Hessian matrix

$$\bar{\bar{\mathbf{H}}} = \begin{bmatrix} \partial_x(\partial_x\phi) & \partial_x(\partial_y\phi) & \partial_x(\partial_z\phi) \\ \partial_y(\partial_x\phi) & \partial_y(\partial_y\phi) & \partial_y(\partial_z\phi) \\ \partial_z(\partial_x\phi) & \partial_z(\partial_y\phi) & \partial_z(\partial_z\phi) \end{bmatrix},$$

the function $\phi(\mathbf{x})$ is defined to have zero curvature at $\mathbf{x}$ if

$$\mathbf{d}_k^T\bar{\bar{\mathbf{H}}}\mathbf{d}_k = 0, \tag{7}$$

for all directional vectors $\mathbf{d}_k$ at that point. Clearly a function satisfying

$$\mathscr{R}_6\phi(\mathbf{x}) = 0$$

has zero curvature as defined in Equation 7. On the other hand, functions satisfying

$$\mathscr{R}_1\phi(\mathbf{x}) = 0 \text{ or } \mathscr{R}_3\phi(\mathbf{x}) = 0$$

may not necessarily have zero curvature as those operators do not involve cross derivatives. Nevertheless, the operators $\mathscr{R}_1, \mathscr{R}_3$ are still useful in practice. In fact, we will later show that the discretization of $\mathscr{R}_1$ proposed here on tetrahedral meshes is equivalent to the constant gradient regularization operator of Frank [5] that has been the workhorse of mesh-based implicit structural modeling for over a decade [7, 32, 33, 1].

As seen above, the Hessian minimization has an interesting potential for minimizing the curvature in all directions. It has recently been implemented on Cartesian grids in 3D [22], where faults exhibit a stairstep effect. One of the goals of this paper is to consider the use of tetrahedral meshes, which can be generated conformably to faults and other geological discontinuities. In Section 1 we briefly describe the problem with a straightforward boundary free discretization of $\mathscr{R}_1$, which introduces artifacts because it assumes vanishing Neumann boundary conditions. We then propose a slight modification of the straightforward discretization that does not impose any boundary constraints for $\mathscr{R}_6$; however, the modification introduces an additional constraint on the mesh that has proven challenging to satisfy in practice for geological modeling because of discontinuities. In Section 2.1, we propose a discretization of $\mathscr{R}_1$ that does not impose any constraints on the boundary and does not impose any additional constraints on the mesh; we propose an equivalent discretization for $\mathscr{R}_3$ and $\mathscr{R}_6$ in Section 2.2. In section 3 we compare our work to existing related literature; we equate $\mathscr{R}_1$ to the regularization operator of [5] and argue that while minimizing this operator minimizes the curvature of the implicit function on triangulated meshes, it may fail to do so on tetrahedral meshes.

Our attention to details in boundary effects comes from the way faults and other discontinuities are handled in structural modeling. Consider the example in Figure 1, the grid generated from the input data is conformal to faults and vertices on faults are duplicated to introduce discontinuities in the grid; as a result, each discontinuity surface introduces an internal boundary [e.g. 34, 7, 9, 22].

The main contributions of this paper are:

1. We propose a finite element discretization of the Laplacian that does not impose any boundary condition. Related effort, such as [35, 36], that discretize the Laplacian using finite elements with Neumann and/or Dirichlet boundary conditions are inadequate for our application, where boundary conditions are unknown.
2. We reveal that the celebrated constant-gradient regularization operator of [5] is a finite element of the Laplacian in disguise; the constant-gradient operator was originally derived from geometrical arguments [see chapter 1 in 37].
3. We show that while appropriate discretizations of the Laplacian, such as those in [38, 39], are sufficient to minimize the curvature of a function on triangulated meshes, they may not be sufficient on tetrahedral meshes.

4. We propose a finite element discretization of the Hessian that does not impose any boundary condition.

1. Problem formulation

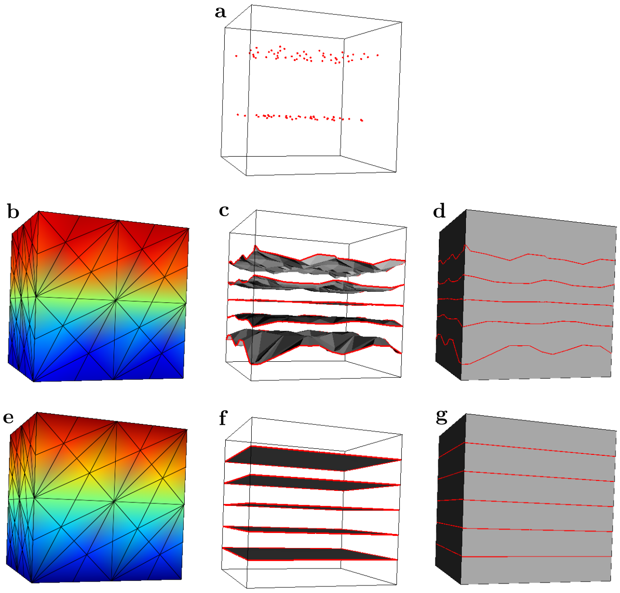


Figure 2: An illustrative example of boundary artifacts introduced by inappropriate boundary conditions. **a**) Input data, point sets at two isovalues. **b-d**) Implicit function, extracted surfaces, and isolines obtained using $\mathscr{R}_1$ (based on Equation 17), assuming vanishing Neumann boundary conditions. **e-g**) Implicit function, extracted surfaces, and isolines obtained using $\mathscr{R}_6$ (based on Equation 23).

Let us consider a straightforward finite element discretization of $\mathscr{R}_1$. This will serve to illustrate the problem at hand, but also to introduce the notation used throughout the paper and to review the finite element method. The short review of finite elements presented here follows the text of [40], to which the reader is referred to for more details. We denote the domain of interest as Ω, its interior and its boundary are denoted by Ω_I and Ω_B respectively such that

$$\Omega = \Omega_I \cup \Omega_B \,, \Omega_I \cap \Omega_B = \varnothing .$$

In practice, the domain of interest Ω is discretized into a set of vertices, edges, triangles, and tetrahedra. We denote the set of triangles by

$$\mathbb{T} = \mathbb{T}_I \cup \mathbb{T}_B \,, \mathbb{T}_I \cap \mathbb{T}_B = \varnothing , \tag{8}$$

where $\mathbb{T}_I$ and $\mathbb{T}_B$ are the set of internal and boundary triangles respectively; a triangle is an internal triangle if it is shared by two tetrahedra, otherwise it is a boundary triangle. Similarly, we denote the set of vertices by

$$\mathbb{V} = \mathbb{V}_I \cup \mathbb{V}_B \,, \mathbb{V}_I \cap \mathbb{V}_B = \varnothing , \tag{9}$$

where $\mathbb{V}_I$ and $\mathbb{V}_B$ are the set of internal and boundary vertices respectively; a vertex is a boundary vertex if it belongs to at least one boundary triangle, otherwise it is an internal vertex.

To discretize

$$\mathscr{R}_1 \phi = \Delta \phi = 0 \tag{10}$$

on Ω using linear finite elements, one reduces the order of derivation on ϕ in Equation 10 using integration by parts and the divergence theorem. In particular, given a *test function* w, Equation 10 can be transformed to

$$\begin{aligned} \int_\Omega w(\mathbf{x}) \Delta \phi(\mathbf{x}) = - \int_\Omega \nabla w(\mathbf{x}) \nabla \phi(\mathbf{x}) \\ + \int_{\Omega_B} w(\mathbf{x}) \nabla \phi(\mathbf{x}) \cdot \mathbf{n} = 0 \end{aligned} \tag{11}$$

$$= - \int_\Omega \nabla w(\mathbf{x}) \nabla \phi(\mathbf{x}) = 0, \tag{12}$$

where $\mathbf{n}$ is the normal on Ω_B. The integral over the boundary Ω_B in Equation 11 is undesirable as we do not a priori know the value of $\nabla \phi(\mathbf{x})$ at the boundary; our goal is to get rid of that integral. For now, we are simply going to ignore that integral, as often done in the related literature [see for example Section 2 in 41, and references therein], so that we end up with Equation 12. Let $V(\mathbf{x}, \mathbf{x}_i)$ denote the standard piecewise linear Lagrange basis function defined on the mesh such that

$$V(\mathbf{x}_j, \mathbf{x}_i) = V_i(\mathbf{x}_j) = \delta_{ji} \text{ for vertices } \mathbf{x}_j, \mathbf{x}_i \in \mathbb{V}; \tag{13}$$

we use V_i for Lagrange basis functions to remind that these functions are attached to vertices v_i (later, we will introduce basis functions not attached to vertices). Using the basis functions V_i, we approximate $w(\mathbf{x})$ and $\phi(\mathbf{x})$ on the mesh as

$$w(\mathbf{x}) = \sum_{\mathbf{x}_i \in \mathbb{V}} V_i(\mathbf{x}) w_i, \tag{14}$$

$$\phi(\mathbf{x}) = \sum_{\mathbf{x}_j \in \mathbb{V}} V_j(\mathbf{x}) \phi_j, \tag{15}$$

transforming Equation 12 to

$$\begin{aligned} 0 &= \sum_{\mathbf{x}_i \in \mathbb{V}} w_i \sum_{\mathbf{x}_j \in \mathbb{V}} \int_\Omega \nabla V_i(\mathbf{x}) \nabla V_j(\mathbf{x}) \phi_j \\ &= \sum_{\mathbf{x}_i \in \mathbb{V}} w_i \sum_{\mathbf{x}_j \in \mathbb{V}} K_{ij} \phi_j = \sum_{\mathbf{x}_i \in \mathbb{V}} w_i G_i . \end{aligned} \tag{16}$$

Equation 16 should be valid for all w_i; because the coefficients w_i are arbitrary, it follows that

$$\begin{aligned} G_i &= \sum_{\mathbf{x}_j \in \mathbb{V}} K_{ij} \phi_j \\ &= \sum_{\mathbf{x}_j \in \mathbb{V}} \int_\Omega \nabla V_i(\mathbf{x}) \nabla V_j(\mathbf{x}) \phi_j = 0, \forall \mathbf{x}_i \in \mathbb{V}. \end{aligned} \tag{17}$$

The matrix K_{ij} in Equation 17 can then be assembled by iterating over each tetrahedron independently as is customary in finite elements [40].

By ignoring the boundary integral in Equation 11, we implicitly imposed

$$\nabla\phi(\mathbf{x})\cdot\mathbf{n} = 0, \text{ for } \mathbf{x}\in\Omega_B, \tag{18}$$

since we know from Equation 14 that $w(\mathbf{x})$ may not necessarily vanish on Ω_B. To illustrate an example of artifacts introduced by this assumption, we interpolate the two point sets in Figure 2a by imposing Equation 17, the resulting implicit function is shown in Figure 2b. Figures 2c-d show surfaces and isolines extracted at five equal intervals. We clearly see that curvature of ϕ is not minimized: those isolines tend to concentrate in the middle of the model and they are not flat at the boundary. For reference, Figures 2e-g show the results obtained using the method presented in the next paragraph.

1.1. Towards a boundary free interpolation

Our strategy for developing a boundary free interpolation will be to look for a test function $w(\mathbf{x})$ that has a vanishing integral at the boundary Ω_B, that is,

$$\int_{\Omega_B} w(\mathbf{x}) = 0; \tag{19}$$

this way, the boundary integral in Equation 11 can be discarded without implicitly imposing any constraint on $\phi(\mathbf{x})$ at the boundary. First, let us consider a test function $w(\mathbf{x})$ such that

$$w(\mathbf{x}) = 0 \text{ for all } \mathbf{x}\in\Omega_B \tag{20}$$

as in [42]. Such a function is easily obtained by replacing Equation 14 with

$$w(\mathbf{x}) = \sum_{\mathbf{x}_i\in\mathbb{V}_I} V_i(\mathbf{x})w_i; \tag{21}$$

the only difference between this equation and Equation 14 is the domain of summation: the test function defined in Equation 21 vanishes at the boundary, which follows from the delta property of the Lagrange basis functions V_i (Equation 13). In traditional finite elements, one uses a test function w that vanishes at Dirichlet boundaries where the values of ϕ are known [40]; here, we want a test function w that vanishes on the entire boundary and we do not know the values of ϕ at the boundaries. Inserting Equations 21 and 15 into the left hand side of Equation 11 leads to

$$\sum_{\mathbf{x}_j\in\mathbb{V}} K_{ij}\phi_j = \sum_{\mathbf{x}_j\in\mathbb{V}}\int_\Omega \nabla V_i(\mathbf{x})\nabla V_j(\mathbf{x})\phi_j = 0, \forall\mathbf{x}_i\in\mathbb{V}_I. \tag{22}$$

Unlike Equation 17, this equation does not imply

$$\nabla\phi(\mathbf{x})\cdot\mathbf{n} = 0$$

on the boundary. However, because $|\mathbb{V}_I| < |\mathbb{V}|$, the matrix form of Equation 22 has less rows than columns and therefore the problem is ill-posed.

Let us now consider the discretization of $\mathscr{R}_6$ (Equation 6). We detail only the first term, the rest of the terms are discretized in a similar way. Starting with the integral

$$\int_\Omega w(\mathbf{x})\partial_x^2\phi(\mathbf{x}) = \int_z\int_y\Big(\int_x w(\mathbf{x})\partial_x^2\phi(\mathbf{x})\Big)$$
$$= \int_z\int_y\Big(-\int_x \partial_x w(\mathbf{x})\partial_x\phi(\mathbf{x}) + \big[w(\mathbf{x})\partial_x\phi(\mathbf{x})\big]_{\mathbf{x}_1\in\mathbb{V}_B}^{\mathbf{x}_2\in\mathbb{V}_B}\Big),$$

for some two boundary points $\mathbf{x}_1, \mathbf{x}_2$, we use Equations 21 and 15 to obtain the discretization of $\partial_x^2\phi = 0$ as

$$\sum_{\mathbf{x}_j\in\mathbb{V}} K_{ij}^1\phi_j = \int_\Omega \partial_x V_i(\mathbf{x})\partial_x V_j(\mathbf{x})\phi_j = 0, \text{for all } \mathbf{x}_i\in\mathbb{V}_I, \tag{23}$$

where the superscript 1 in K_{ij}^1 refers to the 1st term of the operator $\mathscr{R}_6$. The complete regularization matrix is obtained by gathering the six matrices (one for each term)

$$\bar{\bar{\mathbf{K}}} = \begin{bmatrix}\bar{\bar{\mathbf{K}}}^1\\ \bar{\bar{\mathbf{K}}}^2\\ \bar{\bar{\mathbf{K}}}^3\\ \bar{\bar{\mathbf{K}}}^4\\ \bar{\bar{\mathbf{K}}}^5\\ \bar{\bar{\mathbf{K}}}^6\end{bmatrix}. \tag{24}$$

Like the matrix in Equation 22, this matrix does not impose any constraint on $\phi(\mathbf{x})$ on the boundary, but unlike in Equation 22, we now have more rows than columns since we have six equations at each internal point; this matrix therefore has enough equations for sparse data interpolation. Figures 2e-g show illustrative results obtained by discretizing $\mathscr{R}_6$ based on Equation 24.

While Equation 21 does not constrain $\phi(\mathbf{x})$ at the boundary, it imposes a constraint on mesh generation. Consider for example the illustrative 2D domain in Figure 3a discretized with nine vertices and eight internal edges. The mesh has only one internal vertex: v_4. Because there is no edge connecting the boundary vertices v_0 and v_8 to vertex v_4, the basis function V_4 does not overlap with the basis functions V_0, V_8; therefore, it follows from Equation 23 that the 1st and 9th columns of the matrix $\bar{\bar{\mathbf{K}}}$ (Equation 24) will contain only zeros. In order to avoid such cases, each triangle of the mesh should have at least one internal vertex as shown in Figure 3b. In practice, it has proven challenging to enforce such a mesh constraint, especially in the presence of discontinuities, making the strategy proposed in this section of limited use for subsurface modeling [42].

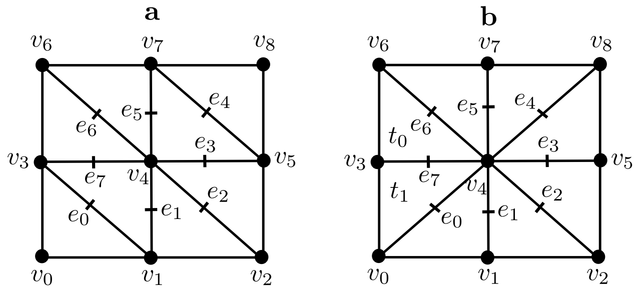


Figure 3: Meshing constraint imposed by Equation 21: in **a**, vertices 0 and 8 will not contribute to the interpolation matrix because they are not connected to the internal vertex 4; the problem can be solved by modifying the mesh to have vertices 0 and 8 connected to the internal vertex 4 as shown in **b**.

1.2. Problem statement

The problem of interest in this paper is to propose finite element discretizations of $\mathscr{R}_1$ (Equation 4), $\mathscr{R}_3$ (Equation 5), and $\mathscr{R}_6$ (Equation 6) that do not impose any constraint on the implicit function $\phi(\mathbf{x})$ at boundary points $\mathbf{x} \in \Omega_B$ (for example Equation 18), and that do not impose any constraint on the input mesh (for example Figure 3). Furthermore, we are only interested in discretizations based on linear basis functions.

2. Boundary free interpolation

Preliminary – Crouzeix-Raviart basis functions

We now look for a test function $w(\mathbf{x})$ that satisfies Equation 19 without necessarily satisfying Equation 20. For this purpose, we are going to use Crouzeix-Raviart basis functions [43], which we review shortly. Consider a triangle in 2D in the physical space (Figure 4a) and its mapping in the reference space (Figure 4b). The Lagrange basis functions V_i on vertices v_i in the reference space are given by

$$\begin{array}{ll} V_0 & = 1 - \xi - \eta \\ V_1 & = \xi \\ V_2 & = \eta \end{array} \quad , \tag{25}$$

and the scalar Crouzeix-Raviart basis functions E_i on edges e_i are given by

$$\begin{array}{lll} E_0 & = 1 - 2V0 & = -1 + 2(\xi + \eta) \\ E_1 & = 1 - 2V1 & = 1 - 2\xi \\ E_2 & = 1 - 2V2 & = 1 - 2\eta \end{array} \quad , \tag{26}$$

where the edge e_i is opposite the vertex v_i as illustrated in Figure 4. The functions in Equation 26 have the properties that

$$\sum_i E_i = 1, \tag{27}$$

$$\nabla E_i = \mathbf{n}_i k, \text{ for some } k \in \mathbb{R}\backslash\{0\} \tag{28}$$

where $\mathbf{n}_i$ is the normal vector of the edge e_i and, of particular interest to our application,

$$\int_{e_i} E_j = \delta_{ij} k, \text{ for some } k \in \mathbb{R}\backslash\{0\}\,. \tag{29}$$

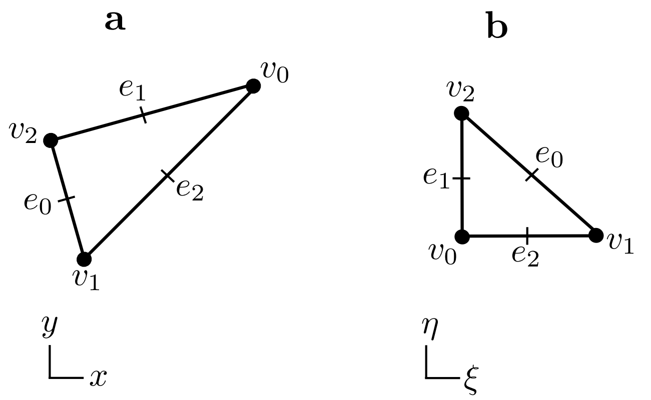


Figure 4: A triangle in the physical space (**a**) and its mapping into the reference space (**b**).

To see that the integral of the function E_j along the edge e_i vanishes when $j \neq i$, it suffices to note that E_j vanishes at the midpoint of edge e_i: because E_j is linear, a Gaussian quadrature with one integration point at the midpoint of the edge e_i would compute that integral exactly (to zero). Similarly, for a tetrahedron in 3D we have Lagrange basis functions V_i on vertices v_i

$$\begin{array}{ll} V_0 & = 1 - \xi - \eta - \gamma \\ V_1 & = \xi \\ V_2 & = \eta \\ V_3 & = \gamma \end{array} \quad , \tag{30}$$

and Crouzeix-Raviart basis functions T_i on triangles t_i

$$\begin{array}{lll} T_0 & = 1 - 3V0 & = -2 + 3(\xi + \eta + \gamma) \\ T_1 & = 1 - 3V1 & = 1 - 3\xi \\ T_2 & = 1 - 3V2 & = 1 - 3\eta \\ T_3 & = 1 - 3V2 & = 1 - 3\gamma \end{array} \tag{31}$$

which satisfy

$$\begin{array}{rll} \sum_i T_i & = 1 & (a) \\ \nabla T_i & = \mathbf{n}_i k, \text{ for some } k \in \mathbb{R}\backslash\{0\} & (b) \\ \displaystyle\int_{t_i} T_j & = \delta_{ij} k, \text{ for some } k \in \mathbb{R}\backslash\{0\} & (c) \end{array} \tag{32}$$

where $\mathbf{n}_i$ is the normal vector of the triangle t_i.

2.1. Interpolation with the Laplacian

We now have all the information needed to propose a boundary free discretization of $\mathscr{R}_1$ (Equation 4). Let us start by transforming Equation 11 into a sum of integrals on individual tetrahedra tet_i of the mesh

$$\begin{array}{rll} 0 & = \displaystyle\int_\Omega w(\mathbf{x}) \Delta\phi(\mathbf{x}) & \\ & = -\displaystyle\sum_i \int_{tet_i} \nabla w(\mathbf{x}) \nabla\phi(\mathbf{x}) + \sum_{t_j \in \mathbb{T}_B} \int_{t_j} w(\mathbf{x}) \nabla\phi(\mathbf{x}) \cdot \mathbf{n} & (a) \\ & = -\displaystyle\sum_i \int_{tet_i} \nabla w(\mathbf{x}) \nabla\phi(\mathbf{x}) + \sum_{t_j \in \mathbb{T}_B} \nabla\phi_j \cdot \mathbf{n}_j \int_{t_j} w(\mathbf{x}) & (b) \end{array} , \tag{33}$$

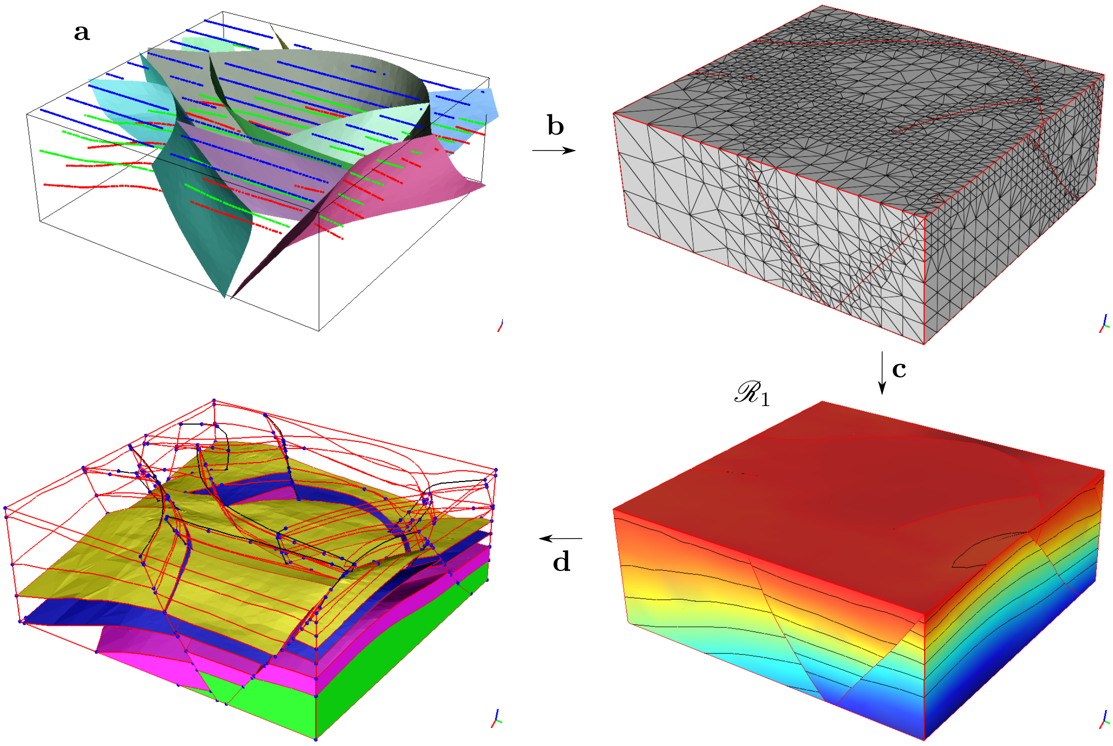


Figure 5: (**a**) Input data (courtesy of TotalEnergies): point sets from three horizons and 10 fault surfaces. (**b**) Generation of a grid that is discontinuous across input faults. (**c**) Interpolation of horizon data using $\mathscr{R}_1$ (Equation 38). (**d**) Extraction of a sealed structural model.

where $\nabla\phi_j$ is the gradient of the tetrahedron that contains the boundary triangle $t_j \in \mathbb{T}_B$; we have pulled out the gradient from the integral as it is constant for each tetrahedron. We see that approximating (compare this with Equation 21)

$$w(\mathbf{x}) = \sum_{t_i \in \mathbb{T}_I} T_i(\mathbf{x}) w_i, \tag{34}$$

where the basis functions T_i corresponding to triangles t_i are those defined in Equation 31, eliminates the boundary integral

$$\sum_{t_j \in \mathbb{T}_B} \nabla\phi_j \cdot \mathbf{n}_j \int_{t_j} w(\mathbf{x}) = \sum_{t_j \in \mathbb{T}_B} \nabla\phi_j \cdot \mathbf{n}_j \int_{t_j} \sum_{t_k \in \mathbb{T}_I} T_k w_k \tag{35}$$

$$= \sum_{t_j \in \mathbb{T}_B} \nabla\phi_j \cdot \mathbf{n}_j \sum_{t_k \in \mathbb{T}_I} w_k \int_{t_j} T_k \tag{36}$$

$$= \sum_{t_j \in \mathbb{T}_B} \nabla\phi_j \cdot \mathbf{n}_j \sum_{t_k \in \mathbb{T}_I} w_k \delta_{jk} = 0. \tag{37}$$

The last equality follows from Equations 32c and 8 ($t_j \in \mathbb{T}_B, t_k \in \mathbb{T}_I \rightarrow j \neq k$). Finally, approximating $\phi(\mathbf{x})$ with Lagrange basis functions V_j (Equation 15) leads to the following matrix representation of Equation 33

$$\begin{aligned} 0 &= \sum_{\mathbf{x}_j \in \mathbb{V}} K_{ij}\phi_j \\ &= \sum_{\mathbf{x}_j \in \mathbb{V}} \sum_k \int_{tet_k} \nabla T_i(\mathbf{x}) \nabla V_j(\mathbf{x}) \phi_j, \ \forall i \text{ s. t. } t_i \in \mathbb{T}_I \end{aligned}, \tag{38}$$

which is the proposed discretization of $\mathscr{R}_1$. It should be noted that the row index i runs over internal triangles, resulting in more rows than columns when $|\mathbb{T}_I| > |\mathbb{V}|$.

Going back to the 2D model in Figure 3a, we see that the boundary vertex v_8 would now contribute to the matrix K_{ij} in Equation 38 as its basis function V_8 overlaps the basis function E_4 of the internal edge e_4; a similar argument holds for the boundary vertex v_0. Figure 5 shows a sealed structural model extracted from a stratigraphic function based on Equation 38.

2.2. Interpolation with the Hessian

Numerical evidence has shown that the discretization presented in Section 2.1 is not convenient to discretize the individual components of the Hessian matrix, which are needed for the operators $\mathscr{R}_3$ (Equation 5) and $\mathscr{R}_6$ (Equation 6). To gain some intuition on its limitations, consider the discretization of

$$-\partial_y^2\phi(x,y) = 0 \tag{39}$$

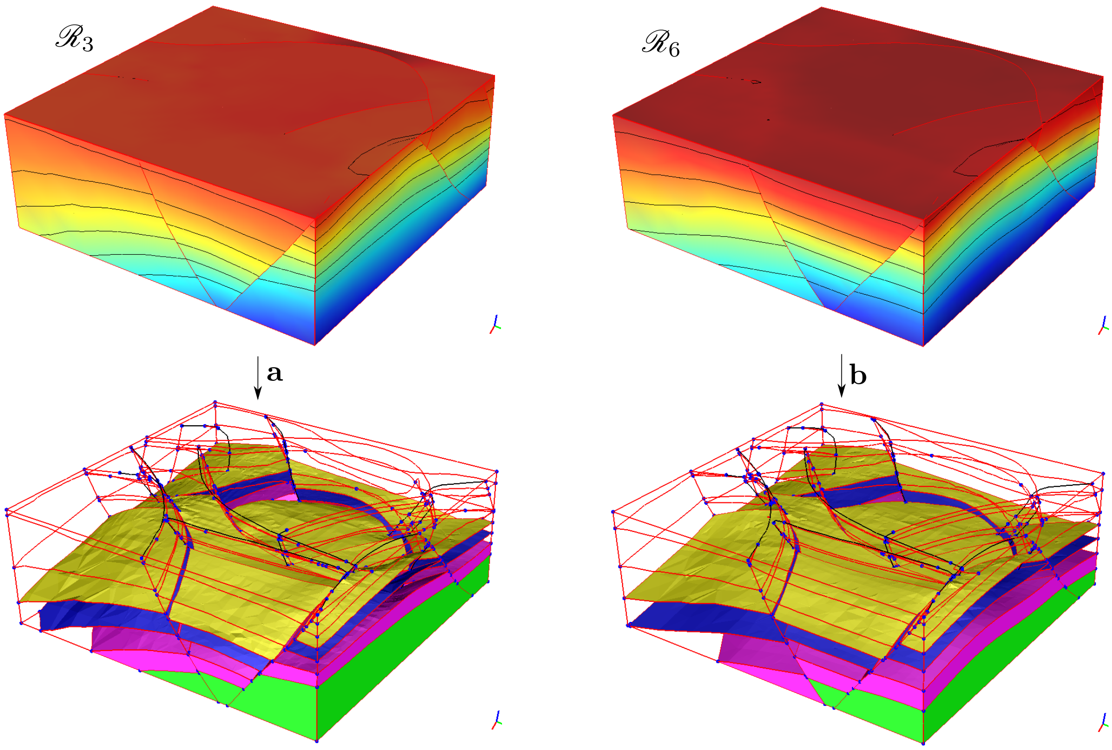


Figure 6: Sealed structural models extracted from stratigraphic functions based on $\mathscr{R}_3$ (**a**) and $\mathscr{R}_6$ (**b**). The stratigraphic functions were computed in the grid from Figure 5b using the discretization in Equation 46.

on the 2D model in Figure 3b using Equation 38. From finite differences, we expect the following contributions from vertices $v3, v4, v5$

$$\begin{array}{lll} \partial_y^2\phi|_{v_3} & = -\phi_0 + 2\phi_3 - \phi_6 & (a) \\ \partial_y^2\phi|_{v_4} & = -\phi_1 + 2\phi_4 - \phi_7 & (b) \\ \partial_y^2\phi|_{v_5} & = -\phi_2 + 2\phi_5 - \phi_8 & (c) \end{array} . \tag{40}$$

Looking at the triangle in Figure 4b and the basis functions in Equations 25 and 26, it can be shown that the contribution of edge e_7 from triangle t_0 in Figure 3b using Equation 38 is

$$\partial_y^2\phi|_{e_7}^{t_0} = \phi_3 - \phi_6,$$

and, by symmetry, the contribution of edge e_7 from triangle t_1 is

$$\partial_y^2\phi|_{e_7}^{t_1} = \phi_3 - \phi_0,$$

bringing the total contribution of e_7 to

$$\partial_y^2\phi|_{e_7} = \partial_y^2\phi|_{e_7}^{t_0} + \partial_y^2\phi|_{e_7}^{t_1} = -\phi_0 + 2\phi_3 - \phi_6.$$

For the entire model, we find that the non zero contributions will come from edges $e_0, e_2, e_3, e_4, e_6, e_7$ as given by

$$\begin{array}{lll} \partial_y^2\phi|_{e_0} & = \phi_0 - \phi_1 - \phi_3 + \phi_4 & (a) \\ \partial_y^2\phi|_{e_2} & = -\phi_1 + \phi_2 + \phi_4 - \phi_5 & (b) \\ \partial_y^2\phi|_{e_3} & = -\phi_2 + 2\phi_5 - \phi_8 & (c) \\ \partial_y^2\phi|_{e_4} & = \phi_4 - \phi_5 - \phi_7 + \phi_8 & (d) \\ \partial_y^2\phi|_{e_6} & = -\phi_3 + \phi_4 + \phi_6 - \phi_7 & (e) \\ \partial_y^2\phi|_{e_7} & = -\phi_0 + 2\phi_3 - \phi_6 & (f) \end{array} . \tag{41}$$

We see that this discretization has captured the derivative $\partial_y^2\phi$ at vertices v_3, v_5 (see Equations 40a,c and 41c,f) but has ignored the derivative at vertex v_4 (Equation 40b). On the other hand, consider accumulating the contributions from all internal edges that are connected to vertex v_4, we find from Equation 41 that

$$\begin{aligned} \partial_y^2\phi|_{v_4} = {} & \partial_y^2\phi|_{e_0} + \partial_y^2\phi|_{e_2} + \partial_y^2\phi|_{e_3} \\ & + \partial_y^2\phi|_{e_4} + \partial_y^2\phi|_{e_6} + \partial_y^2\phi|_{e_7} \\ = {} & -2\phi_1 + 4\phi_4 - 2\phi_7, \end{aligned} \tag{42}$$

which is the equivalent to that expected (Equation 40b) up to a scaling factor. This observation suggests that a more appropriate discretization for the individual components of the Hessian matrix in 3D at a vertex v_i should accumulate the contribution from each internal triangle t_j that is connected to v_i. In particular, let

$$\mathbb{T}_I(v_i)$$

denote the set of internal triangles connected to a vertex $v_i \in \mathbb{V}$, and let

$$\mathbb{V} \cap \mathbb{T}_I$$

denote the set of vertices $v_i \in \mathbb{V}$ such that

$$|\mathbb{T}_I(v_i)| > 0.$$

Define the basis functions

$$N_i(x) = \sum_{t_j \in \mathbb{T}_I(v_i)} T_j(x), \tag{43}$$

which, by Equations 31 and 37, have the properties

$$\begin{aligned} N_i(x) &= V_i(x), \text{ for all } x_i \in \mathbb{V}_I \\ \int_{t_j} N_i(x) &= 0, \text{ for all } t_j \in \mathbb{T}_B \end{aligned} \quad . \tag{44}$$

The approximation (compare with Equations 21 and 34)

$$w(\mathbf{x}) = \sum_{v_i \in \mathbb{V} \cap \mathbb{T}_I} N_i(\mathbf{x}) w_i \tag{45}$$

satisfies Equation 19, and leads to the following discretization of $\partial_x^2 \phi = 0$ (compare with Equation 23)

$$\begin{aligned} 0 &= \sum_{\mathbf{x}_j \in \mathbb{V}} K_{ij}^1 \phi_j \\ &= \sum_{\mathbf{x}_j \in \mathbb{V}} \sum_k \int_{tet_k} \partial_x N_i(\mathbf{x}) \partial_x V_j(\mathbf{x}) \phi_j, \forall v_i \in \mathbb{V} \cap \mathbb{T}_I \\ &= \sum_{\mathbf{x}_j \in \mathbb{V}} \sum_k \sum_{t_m \in \mathbb{T}_I(v_i)} \int_{tet_k} \partial_x T_m(\mathbf{x}) \partial_x V_j(\mathbf{x}) \phi_j, \\ &\qquad \forall v_i \in \mathbb{V} \cap \mathbb{T}_I, \end{aligned} \tag{46}$$

as a more appropriate discretization for $\mathscr{R}_3$ (Equation 5) and $\mathscr{R}_6$ (Equation 6). In Equation 38, each row of the matrix K_{ij} is an equation imposed on an internal triangle $t_i \in \mathbb{T}_I$, while in Equation 46 each row is an equation imposed on a vertex v_i such that there exist an internal triangle $t_m \in \mathbb{T}_I$ that contains v_i; v_i may be an internal or a boundary vertex. Figure 6 shows two sealed structural models extracted from stratigraphic functions based on Equation 46.

3. Discussion

3.1. Related work

The computer graphics community has extensively explored smoothing and regularization operators based on the Laplacian and Hessian. We relate our work to some of that already available in the literature. First, we start with the work of [5] who also performed sparse data interpolation on tetrahedral meshes. They proposed the regularization operator

$$\mathbf{n}_i \cdot \nabla\phi_0(\mathbf{x}) - \mathbf{n}_i \cdot \nabla\phi_1(\mathbf{x}) = 0, \forall i \text{ such that } t_i \in \mathbb{T}_I, \tag{47}$$

where $\mathbf{n}_i$ is the normal vector of the triangle t_i and $\nabla\phi_0, \nabla\phi_1$ are the gradients in the two tetrahedra that share t_i. This operator is equivalent to the discretization of $\mathscr{R}_1$ proposed in Equation 38: the i^{th} row of the matrix in Equation 38 implements

$$\begin{aligned} &\sum_k \int_{tet_k} \nabla T_i(\mathbf{x}) \nabla\phi_k(\mathbf{x}) \\ &= \int_{tet_0} \nabla T_i(\mathbf{x}) \nabla\phi_0(\mathbf{x}) + \int_{tet_1} \nabla T_i(\mathbf{x}) \nabla\phi_1(\mathbf{x}) \quad , \\ &= \int_{tet_0} \mathbf{n}_i \cdot \nabla\phi_0(\mathbf{x}) - \int_{tet_1} \mathbf{n}_i \cdot \nabla\phi_1(\mathbf{x}) \end{aligned} \tag{48}$$

where $\mathbf{n}_i$ is the normal vector of the triangle t_i and $\nabla\phi_0, \nabla\phi_1$ are the gradients in the two tetrahedra tet_0, tet_1 that share t_i; the first equality follows for the local support of the basis functions T_i (they are non zero only inside the respective tetrahedra), and the second equality follows from Equation 32b. While we have omitted examples from this paper, stratigraphic functions computed using Equation 47 are similar to those obtained using Equation 38.

Most of the related work in computer graphics seeks to discretize the Laplacian or the Hessian energies on surfaces in 3D, where those operators are discretized on triangles as opposed to tetrahedra in our case; we will argue in Section 3.2 that a successful discretization of a smoothing operator on triangles may not necessarily extend to tetrahedra. Closely related to our work is that of [41, 44] who also explore a boundary distortion free finite element discretization of the Laplacian and Hessian smoothing energies, but on triangulated surfaces. To show how their approach differs from ours, consider the problem of minimizing the Lagrangian corresponding to Equation 1

$$\mathscr{L} = \int_\Omega [\Delta\phi(\mathbf{x})]^2 + \sum_l \int_\Omega [\phi(\mathbf{x}) - f(\mathbf{x}^l)]^2; \tag{49}$$

$\mathbf{x}^l$ denotes a point inside the l^{th} cell of the mesh, $\mathbf{x}^l$ may or may not belong to the set of mesh points $\mathbb{V}$. Using the mixed finite element formulation of [35]

$$\Delta\phi = 0 \rightarrow \left| \begin{array}{ccc} \psi & = & 0 \\ \Delta\phi & = & \psi \end{array} \right. , \tag{50}$$

the Lagrangian 49 becomes

$$\begin{aligned} &\mathscr{L} \\ &= \frac{1}{2}\int_\Omega \psi\psi + \int_\Omega w(\Delta\phi - \psi) + \frac{1}{2}\sum_l \int_\Omega (\phi^l - f^l)^2 \qquad (a) \\ &= \frac{1}{2}\int_\Omega \psi\psi + \int_\Omega w\Delta\phi - \int_\Omega w\psi + \frac{1}{2}\sum_l \int_\Omega (\phi - f^l)^2 \quad (b) \\ &= \frac{1}{2}\int_\Omega \psi\psi + \int_{\Omega_B} w(\nabla\phi \cdot \mathbf{n}) - \int_\Omega \nabla w \nabla\phi - \int_\Omega w\psi \\ &\qquad + \frac{1}{2}\sum_l \int_\Omega (\phi - f^l)^2 \qquad (c), \end{aligned} \tag{51}$$

where the weight function, w, now plays the role of a Lagrange multiplier. [41] proposes to get rid of the boundary integral in Equation 51c by requiring w to satisfy Equation 20; in particular, the authors discretize the Lagrangian by using piecewise linear approximation of ψ, ϕ as in Equation 15, and Equation 21 to approximate w. This leads to the discrete Lagrangian

$$
\begin{aligned}
\mathscr{L} &= \frac{1}{2}\sum_{x_i\in\mathbb{V}}\sum_{x_j\in\mathbb{V}}\int_\Omega V_iV_j\psi_i\psi_j \\
&- \sum_{x_i\in\mathbb{V}_I}\sum_{x_j\in\mathbb{V}}\int_\Omega \nabla V_i\nabla V_j w_i\phi_j - \sum_{x_i\in\mathbb{V}_I}\sum_{x_j\in\mathbb{V}}\int_\Omega V_iV_jw_i\psi_j \\
&+ \frac{1}{2}\sum_l\int_\Omega \Big(\sum_{x_j\in\mathbb{V}}\sum_{x_i\in\mathbb{V}} V_jV_i\phi_j\phi_i - 2\sum_{x_i\in\mathbb{V}} V_i\phi_i f^l + f^lf^l\Big),
\end{aligned}
\tag{52}
$$

with the corresponding system of equations

$$
\partial_{\psi_k}\mathscr{L} = 0 = \sum_{x_i\in\mathbb{V}}\int_\Omega V_kV_i\psi_i - \sum_{x_j\in\mathbb{V}_I}\int_\Omega V_kV_jw_j, \forall x_k \in \mathbb{V}
$$

$$
\partial_{\phi_k}\mathscr{L} = 0 = -\sum_{x_i\in\mathbb{V}_I}\int_\Omega \nabla V_k\nabla V_i w_i + \sum_l\int_\Omega\Big(\sum_{x_i\in\mathbb{V}} V_kV_i\phi_i - V_kf^l\Big), \forall x_k\in\mathbb{V}
$$

$$
\partial_{w_k}\mathscr{L} = 0 = -\sum_{x_i\in\mathbb{V}}\int_\Omega \nabla V_k\nabla V_i\phi_i - \sum_{x_i\in\mathbb{V}}\int_\Omega V_kV_i\psi_i, \forall x_k\in\mathbb{V}_I. \tag{53}
$$

Equation 53 implies

$$
w = \psi = \Delta\phi, \tag{54}
$$

thereby simplifying the system of equations to

$$
\begin{array}{lll}
\sum_k V_jf^k & = -\sum_{x_i\in\mathbb{V}_I} K_{ij}w_i + \sum_{x_i\in\mathbb{V}} M_{ji}\phi_i, \forall x_j\in\mathbb{V} & (a) \\
0 & = -\sum_{x_j\in\mathbb{V}} K_{ij}\phi_j - \sum_{x_j\in\mathbb{V}_I} M_{ji}w_j, \forall x_i\in\mathbb{V}_I & (b)
\end{array},
\tag{55}
$$

more compactly written as

$$
\begin{bmatrix} -\bar{\bar{K}}^t_{\mathbb{V}_I,\mathbb{V}} & \bar{\bar{M}}_{\mathbb{V},\mathbb{V}} \\ \bar{\bar{M}}^t_{\mathbb{V}_I,\mathbb{V}_I} & \bar{\bar{K}}_{\mathbb{V}_I,\mathbb{V}} \end{bmatrix}
\begin{bmatrix} \bar{w}_{\mathbb{V}_I} \\ \bar{\phi}_{\mathbb{V}} \end{bmatrix}
=
\begin{bmatrix} \bar{f}_{\mathbb{V}} \\ \bar{0}_{\mathbb{V}_I} \end{bmatrix}. \tag{56}
$$

Using Equation 55b

$$
\bar{w}_{\mathbb{V}_I} = -\bar{\bar{M}}^{-t}_{\mathbb{V}_I,\mathbb{V}_I}\bar{\bar{K}}_{\mathbb{V}_I,\mathbb{V}}\bar{\phi}_{\mathbb{V}}
$$

finally leads to

$$
(\bar{\bar{K}}^t_{\mathbb{V}_I,\mathbb{V}}\bar{\bar{M}}^{-t}_{\mathbb{V}_I,\mathbb{V}_I}\bar{\bar{K}}_{\mathbb{V}_I,\mathbb{V}} + \bar{\bar{M}}_{\mathbb{V},\mathbb{V}})\bar{\phi}_{\mathbb{V}} = \bar{f}_{\mathbb{V}}; \tag{57}
$$

this equation does not impose vanishing Neumann boundary condition since Equation 54 implies

$$
\Delta\phi(\mathbf{x}_i) = 0 \to \nabla\phi(\mathbf{x}_i) = \text{constant}, \forall\mathbf{x}_i\in\mathbb{V}_B.
$$

Equation 57 is comparable to our least squares formulation in Section 1.1, particularly Equation 22, which would lead to

$$
(\bar{\bar{K}}^t_{\mathbb{V}_I,\mathbb{V}}\bar{\bar{K}}_{\mathbb{V}_I,\mathbb{V}} + \bar{\bar{M}}^t_{\mathbb{V},\mathbb{V}}\bar{\bar{M}}_{\mathbb{V},\mathbb{V}})\bar{\phi}_{\mathbb{V}} = \bar{\bar{M}}^t_{\mathbb{V},\mathbb{V}}\bar{f}_{\mathbb{V}};
$$

the main difference is that we do not approximate the mass matrix with a diagonal matrix as done in [35, 41] since our formulation does not require inverting the mass matrix.

According to [39], most commonly used smoothing operators on triangulated surfaces are either equivalent to our $\mathscr{R}_1$ discretization in Equation 17 [e.g. 45], or

$$
\sum_j K_{ij}\phi_j = \sum_j\int_\Omega \nabla E_i(\mathbf{x})\nabla E_j(\mathbf{x})\phi_j, \text{ for internal edges } e_i, e_j,
$$

[e.g. 38], which is comparable to our $\mathscr{R}_1$ discretization in Equation 38 since, by Equation 26,

$$
\nabla E_i(\mathbf{x})\nabla E_j(\mathbf{x}) = -2\nabla E_i(\mathbf{x})\nabla V_j(\mathbf{x}).
$$

3.2. Interpolation on triangles vs tetrahedra

[41] observe qualitatively that their Hessian energy gives similar results to the smoothing energy proposed by [38] (comparable to our $\mathscr{R}_1$ discretization in Equation 38), which is derived from the Laplacian. We now give a more quantitative argument that minimizing $\mathscr{R}_1$ as discretized in Equation 38 can indeed be interpreted as a minimization of the Hessian on triangulated meshes; we show, however, that this interpretation does not carry over to tetrahedral meshes.

Let $\mathbf{n}_i = (n_{ix}, n_{iy})$ be the unit normal vector of the edge e_i shared by two equal area triangles t_0 and t_1 in 2D such that each row of $\mathscr{R}_1$ as given by Equation 48 reads

$$
\begin{array}{rll}
0 & = \int_{t_0}\mathbf{n}^i\cdot\nabla\phi_0(\mathbf{x}) - \int_{t_1}\mathbf{n}^i\cdot\nabla\phi_1(\mathbf{x}) & (a) \\
 & = \mathbf{n}^i\cdot\nabla\phi_0(\mathbf{x}) - \mathbf{n}^i\cdot\nabla\phi_1(\mathbf{x}) & (b) \\
 & = \partial_{ni}\phi_0(\mathbf{x}) - \partial_{ni}\phi_1(\mathbf{x}) & (c) \\
 & \Rightarrow \partial^2_{ni}\phi(\mathbf{x}) = 0 & (d)
\end{array},
\tag{58}
$$

where ∂_{ni} and ∂^2_{ni} denote the first and second directional derivatives along the normal $\mathbf{n}_i$. Now consider a triangle t with internal edges e_0, e_1, e_2 and corresponding normals $\mathbf{n}_0, \mathbf{n}_1, \mathbf{n}_2$; it follows from Equation 58 that this particular discretization of $\mathscr{R}_1$ imposes the following constraints at the triangle t

$$
\begin{array}{llll}
\partial^2_{n0}\phi & = \mathbf{n}_0^t\bar{\bar{\mathbf{H}}}\mathbf{n}_0 & = n^2_{0x}H_{xx} + n^2_{0y}H_{yy} + 2n_{0x}n_{0y}H_{yx} & = 0 \\
\partial^2_{n1}\phi & = \mathbf{n}_1^t\bar{\bar{\mathbf{H}}}\mathbf{n}_1 & = n^2_{1x}H_{xx} + n^2_{1y}H_{yy} + 2n_{1x}n_{1y}H_{yx} & = 0 \\
\partial^2_{n2}\phi & = \mathbf{n}_2^t\bar{\bar{\mathbf{H}}}\mathbf{n}_2 & = n^2_{2x}H_{xx} + n^2_{2y}H_{yy} + 2n_{2x}n_{2y}H_{yx} & = 0
\end{array},
\tag{59}
$$

where $\bar{\bar{\mathbf{H}}}$ is the 2D Hessian matrix. Because the normals $\mathbf{n}_0, \mathbf{n}_1, \mathbf{n}_2$ are linearly independent (assuming the triangle

t is not degenerate), the matrix

$$\begin{bmatrix} n_{0x}^2 & n_{0y}^2 & \sqrt{2}n_{0x}n_{0y} \\ n_{1x}^2 & n_{1y}^2 & \sqrt{2}n_{1x}n_{1y} \\ n_{2x}^2 & n_{2y}^2 & \sqrt{2}n_{2x}n_{2y} \end{bmatrix} \tag{60}$$

appearing in Equation 59 has linearly independent rows (see Appendix A) and therefore Equation 59 has the unique solution

$$\begin{array}{rcll} H_{xx} & = & \partial_x(\partial_x\phi) & = 0 \\ H_{yy} & = & \partial_y(\partial_y\phi) & = 0 \\ \sqrt{2}H_{xy} & = & \sqrt{2}\partial_x(\partial_y\phi) & = 0 \end{array},$$

which is the equivalent of

$$\mathscr{R}_6\phi(\mathbf{x}) = \mathbf{0}$$

in 2D. For triangulated surfaces in 3D, one would locally map the triangle t and its neighbors to the tangent plane at t and then apply the same argument. It is in this sense that the minimization of $\mathscr{R}_1$ discretized using Equation 38/47/48 can be interpreted to minimize the Hessian for triangulated surfaces. For tetrahedral meshes, the equivalent of Equation 59 has four equations (one for each facet of the tetrahedron) and six unknowns (one for each independent component of the 3D Hessian matrix) and therefore allows solutions that do not imply a vanishing Hessian. This means that minimizing $\mathscr{R}_1$ as discretized in Equation 38/47/48 on tetrahedral meshes may not necessarily minimize the curvature of ϕ as defined by Equation 7. We conclude from a similar argument that minimizing $\mathscr{R}_3$ may not necessarily minimize the curvature either.

3.3. Interpolation on tetrahedra: Laplacian vs Hessian

We argued in Section 3.2 that, in theory, minimizing $\mathscr{R}_1$ (or $\mathscr{R}_3$) is not sufficient to minimize the curvature of a function on tetrahedral meshes. In practice, $\mathscr{R}_1$ tends be as useful as $\mathscr{R}_6$ for most cases. The different regularization operators do give different results, especially in regions where data is sparse, but it is often difficult to decide which one is better because of the non uniqueness of the problem and the lack of a reference solution. An example of this can be seen in Figure 7: the three regularization operators give (slightly) different results but are all consistent with input data. We have however observed some cases where $\mathscr{R}_1$ seemed to have objectively underperformed compared to $\mathscr{R}_6$. One such example is shown in the highlighted region in Figure 8b: the region illustrates an instance where $\mathscr{R}_6$ has arguably done a better job at minimizing the curvature of ϕ. A closer look shows that this region is in a isolated fault block with insufficient constraining data: all the constraining data in that fault block lie on a vertical plane. The second instance is shown in Figure **??** where we compare a fault-horizon contact line from $\mathscr{R}_1$ (Figure **??**b) with one from $\mathscr{R}_6$ (Figure **??**d), we note that the contact line from $\mathscr{R}_1$ is more sinuous; the contact line from $\mathscr{R}_3$ (Figure **??**d) is even more sinuous. In general, we found $\mathscr{R}_3$ to be the least effective of the three regularizations operators in terms of curvature minimization.

3.4. A comment on fault modeling

Because, as mentioned in the introduction, our primary focus is the numerical and mathematical analysis of the regularization system 3, we have used the bare minimum of information needed to build our data system 2. As a result, some models may present a few geological inconsistencies. This is the case, for example, in the model in Figure 6b where some layers have different thickness' across some faults, or in models in Figure 7 where there is a reversal of fault throw on some faults. These fault inconsistencies, which are likely to occur across a fault that completely separates a region into two independent fault-blocks, can be resolved by estimating fault displacement profiles and then integrating that information in the data system 2 to explicitly constrain fault throws. For more details on this subject, the reader is referred to chapter 4 of [19] and chapter 7 of [9]. Alternative approaches for constraining fault throws are explored in [46, 47].

4. Conclusion

We have presented finite element discretizations for three regularization operators that are suitable for implicit subsurface structural modeling, where boundary conditions are rarely specified. The regularizations are based on the Laplacian and the Hessian of the implicit function; the applicability of the proposed discretizations to structural modeling is illustrated on one highly faulted analogue structural model (Figure 1) and on two field data sets (Figures 5 and 7). We show that our discretization of the Laplacian is equivalent to the regularization operator of [5] that is routinely used for subsurface modeling. We argue (Section 3.2) that the Laplacian may not be as adequate for interpolation on tetrahedral meshes as it is for triangular meshes; this claim is further supported by illustrative examples where the Hessian performs better than the Laplacian, in terms of curvature minimization (Section 3.3).

Acknowledgements

The authors would like to thank IFPEN and C&C Reservoirs for the data used in Figure 1, TotalEnergies for the data used in Figure 5, and an anonymous source for the data used in Figure 7. We are thankful to two anonymous reviewers for improving this paper. Modeste Irakarama would also like to thank Jean-Luc Rudkiewicz (IFPEN), Pierre Thore (TotalEnergies) for discussions that initiated this work, Pierre Anquez (formerly RING) for discussions on mesh generation, and Mark Verschuren for an early review of this paper. Part of this work was done in the frame of the RING project at Université de Lorraine, we would

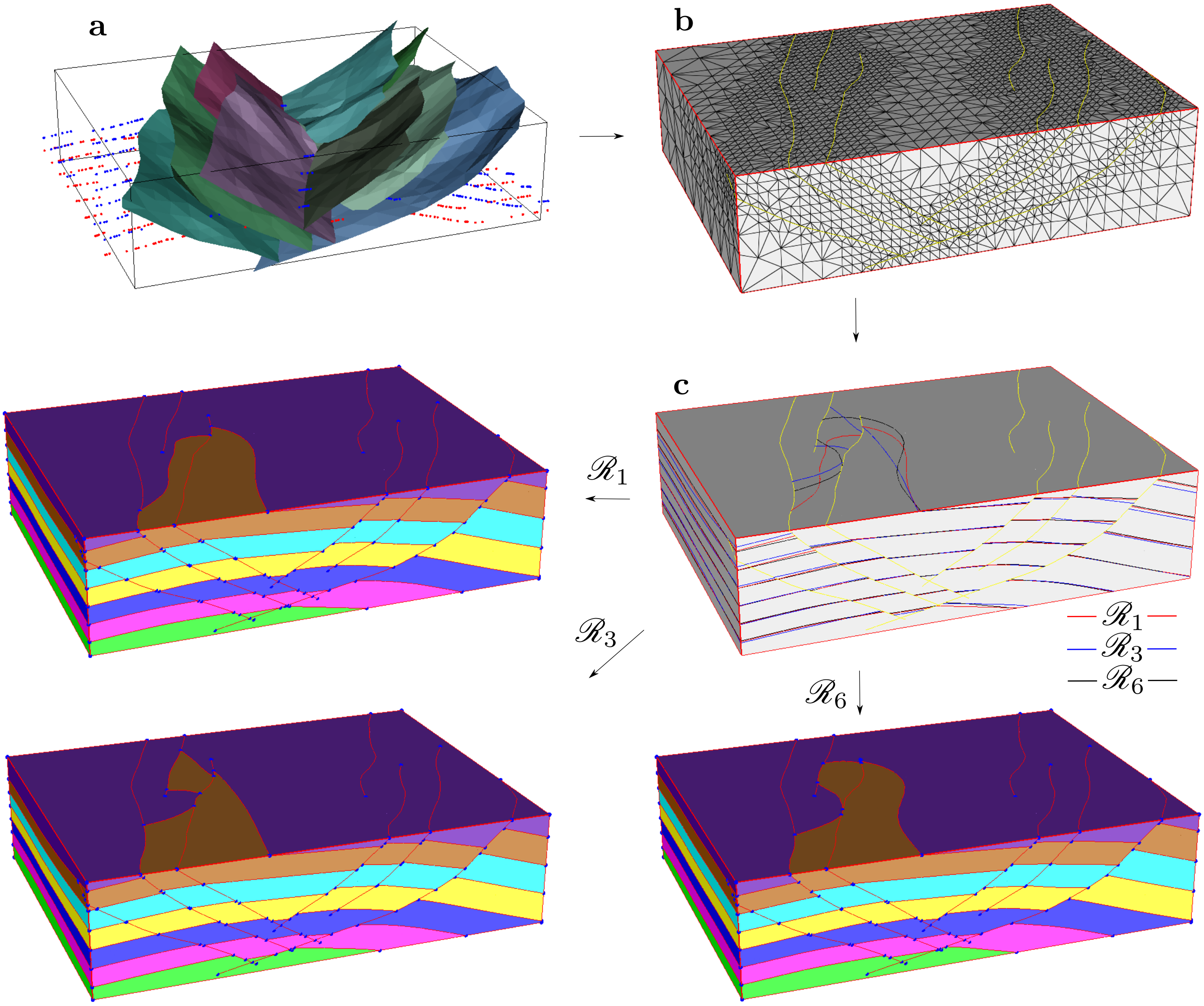


Figure 7: **a**) Input data (horizon point sets from two horizons and 9 fault surfaces, courtesy of an anonymous source). **b**) Grid that is discontinuous across input faults. **c**) Comparison between $\mathscr{R}_1$ (red), $\mathscr{R}_3$ (blue) and $\mathscr{R}_6$ (black).

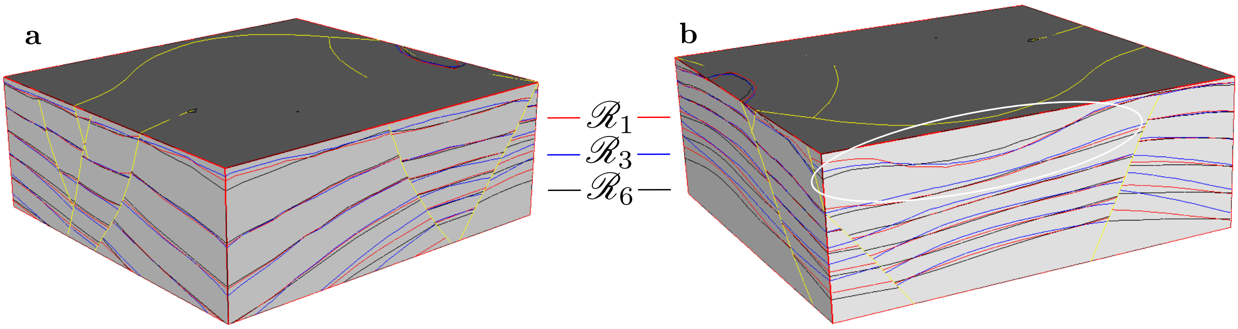


Figure 8: Comparison between $\mathscr{R}_1$ (red), $\mathscr{R}_3$ (blue) and $\mathscr{R}_6$ (black) for models from the data in Figure 5a.

therefore like to thank the sponsors of the RING-GOCAD Consortium managed by ASGA for their support.

Appendix A. Linear independence of Equation 59

We now argue that if the triangle t has edges e_0, e_1, e_2 and corresponding normals $\mathbf{n}_0, \mathbf{n}_1, \mathbf{n}_2$, the matrix in Equation 60 has linearly independent rows if the normals $\mathbf{n}_0, \mathbf{n}_1, \mathbf{n}_2$ are linearly independent. Linear independence of the unit normals $\mathbf{n}_0, \mathbf{n}_1$ implies

$$\mathbf{n}_0 \cdot \mathbf{n}_1 = n_{0x} n_{1x} + n_{0y} n_{1y} \neq \pm 1,$$

which implies the linear dependence of the first two rows

$$\begin{aligned} \mathbf{r}_0 &= (n_{0x}^2, n_{0y}^2, \sqrt{2} n_{0x} n_{0y}) \\ \mathbf{r}_1 &= (n_{1x}^2, n_{1y}^2, \sqrt{2} n_{1x} n_{1y}) \end{aligned}$$

of the matrix in Equation 60 by

$$\begin{aligned} \tfrac{\mathbf{r}_0 \cdot \mathbf{r}_1}{|\mathbf{r}_0||\mathbf{r}_1|} &= \mathbf{r}_0 \cdot \mathbf{r}_1 \\ &= (\mathbf{n}_0 \cdot \mathbf{n}_1)^2 \neq \pm 1 \end{aligned},$$

where we have used

$$|\mathbf{r}_i| = \sqrt{n_{ix}^4 + n_{iy}^4 + 2 n_{ix}^2 n_{iy}^2} = \sqrt{(\mathbf{n}_i \cdot \mathbf{n}_i)^2} = \mathbf{n}_i \cdot \mathbf{n}_i = 1.$$

References

[1] F. Wellmann, G. Caumon, 3-D Structural geological models: Concepts, methods, and uncertainties, Advances in Geophysics 59 (2018) 1–121.

[2] C. Lajaunie, G. Courrioux, L. Manuel, Foliation fields and 3D cartography in geology: Principles of a method based on potential interpolation, Mathematical Geology 29 (1997).

[3] E. Cowan, R. Beatson, W. Fright, T. McLennan, T. Mitchell, Rapid geological modeling, in: Applied Structural Geology for Mineral Exploration and Mining, Kalgoorlie, Australia, 2002, pp. 1–9.

[4] J.-L. Mallet, Space-time mathematical framework for sedimentary geology, Mathematical Geology 36 (2004).

[5] T. Frank, 3D reconstruction of complex geological interfaces from irregularly distributed and noisy point data, Computers and Geosciences 33 (2007) 932–943.

[6] P. Calcagno, J. Chilès, G. Courrioux, A. Guillen, Geological modelling from the field data and geological knowledge Part I. Modeling method coupling 3D potential-field interpolation and geological rules, Physics of the Earth and Planetary Interiors 17 (2008) 147–157.

[7] G. Caumon, G. Gray, C. Antoine, M.-O. Titeux, Three-dimensional implicit stratigraphic model building from remote sensing data on tetrahedral meshes: theory and application to regional model of La Popa Basin, NE Mexico, IEEE Transactions on Geoscience and Remote Sensing 51 (2013).

[8] L. Souche, F. Lepage, G. Iskenova, Volume Based Modeling - Automated construction of complex structural models, in: 75th EAGE Conference and Exhibition, 2013.

[9] J.-L. Mallet, Elements of mathematical sedimentary geology: The GeoChron model, EAGE Publications, 2014.

[10] M. Irakarama, M. Thierry-Coudon, M. Zakari, P. Anquez, G. Caumon, Implicit 3D Subsurface Structural Modeling by Finite Elements, in: 82nd EAGE Conference and Exhibition, Amsterdam, The Netherlands, 2021.

[11] B. Colletta, J. Letouzey, R. Pinedo, J. F. Ballard, P. Balé, Computerized X-ray tomography analysis of sandbox models: Examples of thin-skinned thrust systems, Geology 19 (1991) 1063–1067.

[12] B. Chauvin, P. Lovely, J. Stockmeyer, A. Plesch, G. Caumon, J. Shaw, Validating novel boundary conditions for three-dimensional mechanics-based restoration: An extensional sandbox model example, AAPG Bulletin 102 (2018) 245–266.

[13] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, C. T. Silva, A survey of surface seconstruction from point clouds, Computer Graphics Forum 36 (2017) 301–329.

[14] J.-P. Chilès, C. Aug, A. Guillen, T. Less, Modeling the geometry of geological units and its uncertainty in 3d form structural data: the potential-field method, in: Orebody Modeling and Strategic Mine Planning, volume Spectrum 14, Perth, WA, 2004.

[15] R. Martin, J. Boisvert, Iterative refinement of implicit boundary models for improved geological feature reproduction, Computers and Geosciences 109 (2017) 1–15.

[16] J. Renaudeau, E. Malvesin, F. Maerten, G. Caumon, Implicit Structural Modeling by Minimization of the Bending Energy with Moving Least Squares Functions, Mathematical Geosciences 51 (2019) 693–724.

[17] L. Yang, P. Achtziger-Zupančič, J. K. Caers, 3d modeling of large-scale geological structures by linear combinations of implicit functions: Application to a large banded iron formation, Natural Resources Research 30 (2021) 3139 – 3163.

[18] L. Pizzella, R. Alais, S. Lopez, X. Freulon, J. Rivoirard, Taking better advantage of fold axis data to characterize anisotropy of complex folded structures in the implicit modeling framework, Mathematical Geosciences 54 (2021) 95–130.

[19] R. Moyen, "Paramétrisation 3D de l'espace en géologie sédimentaire: le modèle GeoChron", Ph.D. thesis, Université de Lorraine-ENSG, Nancy, France, 2005.

[20] M. Irakarama, G. Laurent, J. Renaudeau, G. Caumon, Finite Difference Implicit Modeling of Geological Structures, in: 80th EAGE Conference and Exhibition, Copenhagen, Denmark, 2018.

[21] J. Renaudeau, M. Irakarama, G. Laurent, F. Maerten, G. Caumon, Implicit modeling of geological structures: A Cartesian grid method handling discontinuities with ghost points, in: 41st International Conference on Boundary Elements and other Mesh Reduction Methods, New Forest, UK, 2018.

[22] M. Irakarama, G. Laurent, J. Renaudeau, G. Caumon, Finite Difference Implicit Structural Modeling of Geological Structures, Mathematical Geosciences 53 (2020) 785–808.

[23] Z. Bi, X. Wu, Z. Geng, H. Li, Deep relative geologic time: a deep learning method for simultaneously interpreting 3-d seismic horizons and faults, Journal of Geophysical Research 126 (2021).

[24] G. Caumon, F. Lepage, C. H. Sword, J.-L. Mallet, Building and editing a sealed geological model, Mathematical Geology 36 (2004) 405–424.

[25] H. Si, TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator, ACM Transactions on Mathematical Software 41 (2015) 1–36.

[26] B. Zehner, J. H. Börner, I. Görz, K. Spitzer, Workflows for generating tetrahedral meshes for finite element simulations on complex geological structures, Computers & Geosciences 79 (2015) 105–117.

[27] Y. Hu, Q. Zhou, X. Gao, A. Jacobson, D. Zorin, D. Panozzo, Tetrahedral meshing in the wild, ACM Transactions on Graphics 37 (2018) 1–14.

[28] M. Irakarama, Subsurface tetrahedral mesh generation without a sealed surface model, in: 83rd EAGE Conference and Exhibition, Madrid, Spain, 2022.

[29] J.-L. Mallet, Discrete smooth interpolation, ACM Transactions on Graphics 8 (1989) 121–144.

[30] J.-L. Mallet, Discrete smooth interpolation in geometric modeling, Computer-Aided Design 24 (1992) 178–191.

[31] J.-L. Mallet, Discrete Modeling for Natural Objects, Mathe-

matical Geology 29 (1997).

[32] G. Laurent, Iterative thickness regularization of stratigraphic layers in discrete implicit modeling, Mathematical Geosciences 48 (2016) 811–833.

[33] G. Laurent, L. Ailleres, L. Grose, G. Caumon, M. Jessell, R. Armit, Implicit modeling of folds and overprinting deformation, Earth and Planetary Science Letters 456 (2016) 26–38.

[34] J. B. Cavalcante Neto, P. A. Wawrzynek, M. T. M. Carvalho, L. F. Martha, A. R. Ingraffea, An algorithm for three-dimensional mesh generation for arbitrary regions with cracks, Engineering with Computers 17 (2001) 75–91.

[35] A. Jacobson, E. Tosun, O. Sorkine-Hornung, D. Zorin, Mixed finite elements for variational surface modeling, Computer Graphics Forum 29 (2010).

[36] M. Kazhdan, H. Hoppe, Screened Poisson surface reconstruction, ACM Transactions on Graphics 32 (2013) 1–13.

[37] T. Frank, Advanced Visualization and Modeling of Tetrahedral Meshes, Ph.D. thesis, Université de Lorraine-ENSG, Nancy, France, 2006.

[38] M. Bergou, M. Wardetzky, D. Harmon, D. Zorin, E. Grinspun, A quadratic bending model for inextensible surfaces, in: Symposium on Geometry Processing, Eurographics Association, Goslar, DEU, 2006, p. 227–230.

[39] M. Wardetzky, M. Bergou, D. Harmon, D. Zorin, E. Grinspun, Discrete quadratic curvature energies, Computer Aided Geometric Design 24 (2007) 499–518.

[40] T. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Dover Publications, 2000.

[41] O. Stein, E. Grinspun, M. Wardetzky, A. Jacobson, Natural Boundary Conditions for Smoothing in Geometry Processing, ACM 37 (2018).

[42] M. Thierry-Coudon, M. Irakarama, M. Zakari, P. Anquez, G. Caumon, Finite element implementation of second order directional derivatives for regularization of implicit modeling of geological structures, in: 2019 RING meeting, ASGA, Université de Lorraine-ENSG, Nancy, France, 2019.

[43] M. Crouzeix, P.-A. Raviart, Conforming and nonconforming finite element methods for solving the stationary stokes equations i, ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique 7 (1973) 33–75.

[44] O. Stein, A. Jacobson, M. Wardetzky, E. Grinspun, A smoothness energy without boundary distortion for curved surfaces, ACM 39 (2020).

[45] M. Meyer, M. Desbrun, P. Schröder, A. H. Barr, Discrete differential-geometry operators for triangulated 2-manifolds, in: H.-C. Hege, K. Polthier (Eds.), Visualization and Mathematics III, Springer, Berlin, Heidelberg, 2003.

[46] G. Laurent, G. Caumon, A. Bouziat, M. W. Jessell, A parametric method to model 3d displacements around faults with volumetric vector fields, Tectonophysics 590 (2013) 83–93.

[47] G. Godefroy, G. Caumon, M. Ford, G. Laurent, C. A.-L. Jackson, A parametric fault displacement model to introduce kinematic control into modeling faults from sparse data, Interpretation 6 (2018).