



Implicit Structural Modeling by Minimization of the Bending Energy with Moving Least Squares Functions

Julien Renaudeau^{1,2}  · Emmanuel Malvesin¹ ·
Frantz Maerten¹ · Guillaume Caumon²

Received: 18 June 2018 / Accepted: 19 February 2019 / Published online: 7 March 2019
© International Association for Mathematical Geosciences 2019

Abstract In this paper, an implicit structural modeling method using locally defined moving least squares shape functions is proposed. The continuous bending energy is minimized to interpolate between data points and approximate geological structures. This method solves a sparse problem without relying on a complex mesh. Discontinuities such as faults and unconformities are handled with minor modifications of the method using meshless optic principles. The method is illustrated on a two-dimensional model with folds, faults and an unconformity. This model is then modified to show the ability of the method to handle sparsity, noise and different reliabilities in the data. Key parameters of the shape functions and the pertinence of the bending energy for structural modeling applications are discussed. The predefined values deduced from these studies for each parameter of the method can also be used to construct other models.

Keywords 2D–3D structural modeling · Meshless · Local shape functions · Energy penalization · Structural discontinuities

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s11004-019-09789-6>) contains supplementary material, which is available to authorized users.

✉ Julien Renaudeau
julien.renaudeau4@gmail.com; julien.renaudeau@univ-lorraine.fr

Emmanuel Malvesin
emalvesin@slb.com

Frantz Maerten
fmaerten@gmail.com

Guillaume Caumon
guillaume.caumon@univ-lorraine.fr

¹ Schlumberger Petroleum Services, 34000 Montpellier, France

² GeoRessources, UMR 7359, Université de Lorraine, ENSG, 54518 Vandoeuvre-lès-Nancy, France

1 Introduction

Structural geology is partly concerned with the characterization of geological bodies and structures. From the millimetric to the kilometeric scale, these descriptions help in understanding the local and regional geological settings (Fossen 2016). The geometries of folds and geological discontinuities, such as faults and unconformities, are key for developing an understanding of the subsurface geology, which is essential to a range of applications such as the estimation of natural resources.

Structural modeling intends to accurately reproduce the geometry of geological structures with a numerical model (Caumon et al. 2009). The quality of the data acquired on the field depends on acquisition tools, operator skills and rock exposure. Field observations are then interpreted by geologists and geophysicists to obtain the spatial points, lines and vectors used as input data in structural modeling. Depending on both acquisition and interpretation, these numerical inputs may be noisy, sparse and unevenly sampled (both scattered and clustered) (Carmichael and Ailleres 2016; Houlding 1994; Mallet 1992, 1997, 2002). Reproducing complex geological structures from such data requires simplifications and empirical rules based on analog structures.

Implicit structural modeling algorithms have drawn significant attention during the past 30 years (Caumon et al. 2013; Hillier et al. 2014; Lajaunie et al. 1997; Mallet 1988, 2014). They represent a model by an implicit function, also called a stratigraphic function, defined on the entire volume of interest. A horizon, which is an interface between two stratigraphic layers, is given by a single iso-value of the stratigraphic function, and a structural discontinuity, such as a fault or an unconformity, corresponds to a discontinuous jump in the function. The advantage of algorithms building implicit functions is that they take all data constraining the geological structures into account at once, and without involving projections. Geological structures can, therefore, be interpolated and extrapolated away from the data everywhere in the studied area.

The discrete smooth interpolation (DSI) is a class of explicit (Mallet 1992, 1997, 2002) and implicit (Frank et al. 2007; Mallet 1988, 2014; Souche et al. 2014) methods to construct structural models, which is well known in the oil and gas industry [software: SKUA-Gocad by Paradigm (2018) and volume based modeling by Schlumberger (2018)]. The implicit DSI variant discretizes the stratigraphic function on a volumetric mesh. The function's coefficients are centered on the mesh vertices and are linearly interpolated within the mesh elements.

By assuming that the expected model should be as smooth as possible, DSI introduces a global roughness factor minimized in the least squares sense. By locally changing the weight of the roughness, it is possible to control the model features away from the data. Caumon (2009) and Caumon et al. (2013) generate kink folds with this principle. This roughness factor has been formulated with constant gradient equations (Frank et al. 2007) or smooth gradient equations (Souche et al. 2014). As this roughness factor has only been described discretely, its relationship with continuous physical principles remains unclear.

In DSI, the implicit function is continuous on the mesh elements. Therefore, the mesh elements should not intersect structural discontinuities (i.e., the triangles of the discontinuities should be faces of the mesh elements). While unstructured meshes

can efficiently handle discontinuities, their construction represents a challenge in some geological settings (Karimi-Fard and Durlofsky 2016; Pellerin et al. 2014). For instance, mesh algorithms cannot always ensure creating a mesh in which the gradient of the implicit function can be correctly computed (Shewchuk 2002), especially when dealing with fault networks of complex geometries and intersections. It may also be computationally and memory expensive and may require significant user interactions. Moreover, DSI results depend on the mesh and its quality (Laurent 2016).

The potential field method (PFM) (Calcagno et al. 2008; Chilès et al. 2004; Cowan et al. 2003; Lajaunie et al. 1997) is another class of numerical methods that can be used to create implicit structural models and which is well known in the mining industry [software: Geomodeller by Intrepid-Geophysics (2018) and LeapFrog by ARANZ Geo (2018)]. PFM can be formulated as a dual cokriging interpolation or as a radial basis functions interpolation [Hillier et al. (2014), based on Matheron (1981)'s proof of equivalence between kriging and splines]. Some radial basis functions link PFM to physical principles. For instance, the thin plate splines (Duchon 1977) are the Green's functions of the bending energy, which means that an implicit function defined in a PFM scheme by a sum of thin plate splines [as in Jessell et al. (2014)] intrinsically minimizes the bending energy (also called thin plate energy) (Dubrule 1984; Wahba 1990).

In PFM, no mesh is involved in the computation of the implicit function, although the results are generally evaluated on a grid for visualization. Instead, the interpolation is supported by the data and their position: each data point is associated with an interpolant and a coefficient. The implicit function is thus dependent on the data distribution and the range of influence of the chosen interpolants. With global interpolants, such as thin plate splines, the data coefficients have an influence everywhere in space. With local interpolants, such as compactly supported radial basis functions (Wendland 1995; Wu 1995), the data coefficients have a restricted influence centered on their position. In structural modeling, defining the implicit function everywhere in the domain of study is a requirement, as horizons are supposed to be infinite surfaces also existing in wide areas where data are missing. This can be ensured even with local interpolants in PFM by a global polynomial drift added to the solution. However, the transition between sampled areas, influenced by local interpolants, and empty areas, defined by the drift alone, often leads to high curvature artifacts in the solution. Therefore, although local interpolants are common in PFM, such as the cubic covariance (Aug 2004; Calcagno et al. 2008; De la Varga et al. 2019), they are used like global interpolants in structural modeling applications, with a range of influence scaled on the domain's dimensions. This creates a dense system whose size increases with the number of data and becomes unsolvable for more than a few thousand data without optimization techniques (Cowan et al. 2003; Cuomo et al. 2013; Yokota et al. 2010).

Specific treatments as introduced in Marechal (1984) are given to the discontinuities in PFM (Calcagno et al. 2008; Chilès et al. 2004). The faults are handled by enriching the implicit function with polynomial drifts weighted by jump functions. This implies the definition of a fault zone or a fault influence radius for the jump functions, which is difficult to determine (Godefroy et al. 2018) and may require many manual interactions. The stratigraphic unconformities are handled by computing an implicit function for each conformable stratigraphic series, and by using Boolean operations to reconstruct a unique model.

In this article, an algorithm is suggested to build implicit functions using locally defined moving least squares interpolants. Those interpolants are centered on regularly sampled nodes and ensure the definition of the implicit function everywhere in the domain. The discontinuities are handled with meshless techniques, limiting user interactions even in complex cases. The proposed framework also explicitly minimizes the bending energy, as a continuous regularization to solve the structural modeling problem. Everything is described in two dimensions for simplicity, but the formalism is adaptable in three dimensions.

The proposed method is described in Sect. 2. Section 3 emphasizes the ability of the method to handle some well-known issues with data inputs in structural modeling applications. Section 4 discusses the limits of the method, the proposed related solutions, and some perspectives.

2 Locally Based Structural Modeling with Meshless Concepts

2.1 Construction of the Implicit Function

In two dimensions, the implicit function $u(\mathbf{x})$ is a function in the vector space V defined as

$$V = \left\{ u(\mathbf{x}) = \sum_{l=1}^N \Phi_l(\mathbf{x}) u_l = \Phi(\mathbf{x})^T \cdot \mathbf{U} \mid \mathbf{x} \in \Omega \right\}, \quad (1)$$

where $\mathbf{x}(x, y)$ is a position in \mathbb{R}^2 , Ω is the domain of study, $\Phi^T = [\Phi_1, \dots, \Phi_N]$ is a basis of linearly independent functions, $\mathbf{U}^T = [u_1, \dots, u_N]$ is a set of scalar coefficients, and N is the number of terms in Φ and \mathbf{U} . This section explains how the N shape functions Φ_l and coefficients u_l are defined.

2.1.1 Implicit Function Coefficients: The Discretization of the Domain

For simplicity, the domain of study Ω is discretized regularly. The grid is not stored, but its N cell corner points are key to the method and are referred to as interpolation nodes in the remainder of the paper. The implicit function is constructed on these nodes: an interpolant Φ_l and a coefficient u_l are centered on each interpolation node \mathbf{x}_l . The number of interpolants and coefficients is, therefore, equal to the number of interpolation nodes N . The shape functions Φ_l are linearly independent as long as the interpolation nodes have distinct coordinates, so the implicit function is uniquely defined by the coefficients u_l . In this paper, the moving least squares are used as interpolation functions and are described below.

2.1.2 Interpolation Functions: The Moving Least Squares

Weight functions The weight functions w_l are key to define the moving least squares functions Φ_l . They are continuous functions centered on the interpolation nodes

$\mathbf{x}_l(x_l, y_l)$ and defined in a local area S_l called support. In the presented method, they are defined with a rectangular support as

$$w_l(\mathbf{x}) = w(q_{l_x}) w(q_{l_y}) = w\left(\frac{|x - x_l|}{\rho_{l_x}}\right) w\left(\frac{|y - y_l|}{\rho_{l_y}}\right). \tag{2}$$

The dilatation parameters $\rho_l(\rho_{l_x}, \rho_{l_y})$ control the size of the supports S_l and the normalized distances $q_l(q_{l_x}, q_{l_y})$. Here, a global and adimensional dilatation parameter ρ is used to define constant dilatation parameters scaled on the nodal spacing in each axis as

$$\begin{cases} \rho_{l_x} = \rho \frac{L_x(\Omega)}{(N_x - 1)} \\ \rho_{l_y} = \rho \frac{L_y(\Omega)}{(N_y - 1)} \end{cases}, \quad l = \{1, \dots, N\}, \tag{3}$$

where $L_x(\Omega)$ and $L_y(\Omega)$ are the lengths of the domain Ω and N_x and N_y are the number of interpolation nodes for each axis.

A wide range of weight functions have been defined in the literature (Fries and Matthias 2004). In this paper, the fourth order splines are used; they are highly derivable (C^2) and polynomial-based as

$$w(q_l) \begin{cases} = 1 - 6q_l^2 + 8q_l^3 - 3q_l^4 & q_l \leq 1 \\ = 0 & q_l > 1 \end{cases}. \tag{4}$$

Moving least squares (MLS) functions The MLS functions (Lancaster and Salkauskas 1981; McLain 1976) are locally defined meshless functions. They are constructed by approximating the implicit function u as a polynomial function of degree d with spatially varying coefficients. The V space from Eq. (1) is redefined as

$$V = \left\{ u(\mathbf{x}) = \sum_{l=1}^N \Phi_l(\mathbf{x}) u_l = \sum_{j=1}^m p_j(\mathbf{x}) a_j = \mathbf{p}^T(\mathbf{x}) \cdot \mathbf{a}(\mathbf{x}) \mid \mathbf{x} \in \Omega \right\}, \tag{5}$$

with m the number of monomials p_j and coefficients a_j .

When performing a local approximation around \mathbf{x} , in a least squares sense, the MLS shape functions Φ_l are defined as

$$\Phi_l(\mathbf{x}) = w_l(\mathbf{x}) \mathbf{p}^T(\mathbf{x}) \cdot [\mathbf{A}(\mathbf{x})]^{-1} \cdot \mathbf{p}(\mathbf{x}_l) = \mathbf{\Gamma}(\mathbf{x}) \cdot \mathbf{B}_l(\mathbf{x}), \tag{6}$$

where

$$\mathbf{A}(\mathbf{x}) = \sum_{j=1}^N w_j(\mathbf{x}) \mathbf{p}(\mathbf{x}_j) \cdot \mathbf{p}^T(\mathbf{x}_j), \tag{7}$$

and with $\mathbf{A} \cdot \boldsymbol{\Gamma} = \mathbf{p}$ and $\mathbf{B}_1 = w_1(\mathbf{x})\mathbf{p}(\mathbf{x}_1)$. The matrix \mathbf{A} is called the moment matrix. Appendix A gives the details of the construction of MLS functions, and Appendix B gives a geometrical illustration of a one-dimensional MLS interpolation. Some plots of these functions in one and two dimensions can be found in Nguyen et al. (2008) and Liu and Gu (2005). Other useful details on their construction can be found in Fries and Matthias (2004), and Nguyen et al. (2008), and useful empirical values for their parameters can be found in Liu and Gu (2005).

The MLS functions form a partition of unity (PU), which means that a function u defined as in Eq. (1) with MLS functions can exactly fit a constant field. This is a necessary property in the presented approach; without a PU, the solution would bend abnormally between the sample nodes. For instance, the compactly supported radial basis functions do not form a PU. They are thus not adapted to the method without further treatments [e.g., radial polynomial interpolation method, Liu and Gu (2005)].

Moving least squares (MLS) derivatives Solving a modeling problem requires an adapted degree of derivability of the implicit function and, therefore, of its linearly independent interpolation functions. The MLS functions have the same degree of continuity and derivability as the chosen weight functions. With fourth order splines (Eq. 4), they are C^2 .

The exact equations of MLS derivatives are complex and can be found in Fries and Matthias (2004). We use the approximation forms given in Belytschko et al. (1996) and Liu and Gu (2005) where the first order derivative is

$$\frac{\partial}{\partial i} \Phi_1(\mathbf{x}) = \frac{\partial}{\partial i} \boldsymbol{\Gamma}^T \cdot \mathbf{B}_1 + \boldsymbol{\Gamma}^T \cdot \frac{\partial}{\partial i} \mathbf{B}_1, \tag{8}$$

with $\frac{\partial}{\partial i} \boldsymbol{\Gamma}$ the solution of

$$\mathbf{A} \cdot \frac{\partial}{\partial i} \boldsymbol{\Gamma} = \frac{\partial}{\partial i} \mathbf{p} - \frac{\partial}{\partial i} \mathbf{A} \cdot \boldsymbol{\Gamma}, \tag{9}$$

and where the second order derivative is written as

$$\frac{\partial^2}{\partial i j} \Phi_1(\mathbf{x}) = \frac{\partial^2}{\partial i j} \boldsymbol{\Gamma}^T \cdot \mathbf{B}_1 + \frac{\partial}{\partial i} \boldsymbol{\Gamma}^T \cdot \frac{\partial}{\partial j} \mathbf{B}_1 + \frac{\partial}{\partial j} \boldsymbol{\Gamma}^T \cdot \frac{\partial}{\partial i} \mathbf{B}_1 + \boldsymbol{\Gamma}^T \cdot \frac{\partial^2}{\partial i j} \mathbf{B}_1, \tag{10}$$

with $\frac{\partial^2}{\partial i j} \boldsymbol{\Gamma}$ the solution of

$$\mathbf{A} \cdot \frac{\partial^2}{\partial i j} \boldsymbol{\Gamma} = \frac{\partial^2}{\partial i j} \mathbf{p} - \left(\frac{\partial}{\partial i} \mathbf{A} \cdot \frac{\partial}{\partial j} \boldsymbol{\Gamma} + \frac{\partial}{\partial j} \mathbf{A} \cdot \frac{\partial}{\partial i} \boldsymbol{\Gamma} + \frac{\partial^2}{\partial i j} \mathbf{A} \cdot \boldsymbol{\Gamma} \right), \tag{11}$$

where i and j represent the x or the y axis. With these equations, only the matrix \mathbf{A} needs to be inverted to obtain all the derivatives, which thus does not require more computational effort than constructing the shape functions.

2.1.3 Visibility Criterion: A Modification of Interpolation Functions Handling Discontinuities

The discontinuities represent one of the main challenging problems in structural modeling. The discrete smooth interpolation creates meshes conformal to the discontinuities, and the potential field method defines fault zones with polynomials and jump functions, but both approaches become challenging when dealing with complex fault networks and may require heavy user interactions. Stratigraphic unconformities (discontinuities in the sedimentary record) may be addressed by several implicit functions (Calcagno et al. 2008; Chilès et al. 2004), which also requires properly editing the sequences beforehand. Therefore, both types of discontinuities are here addressed as input lines, and they are treated with the visibility criterion (Belytschko et al. 1994).

The implicit function u is defined by Eq. (1) as a sum of weighted continuous functions. The visibility criterion introduces discontinuities in the interpolation functions by truncating their local supports (Belytschko et al. 1994). The principle is as follows: each interpolation node emits light and the discontinuities are opaque to this light. For any point \mathbf{x} , if a discontinuity happens to intersect a ray of light coming from a node \mathbf{x}_i , then \mathbf{x}_i is not considered as a neighbor to \mathbf{x} .

Discontinuous jumps in the implicit function are thus introduced by local intersection tests between segments and discontinuity objects. This approach has the potential to drastically reduce the user interactions to handle structural discontinuities compared to other modeling methods. There are two limitations of the visibility criterion: (I) it performs badly at discontinuity tips, and (II) it completely isolates fault blocks one from another. To solve the tip issue (I), Organ et al. (1996) introduce the diffraction and the transparency criteria. Such criteria can also be employed in the proposed framework, but the tip issue is generally negligible with a sufficient number of interpolation nodes, which is why the visibility criterion is used. Also, if necessary, these criteria may be adapted to solve the isolation issue (II) as shown in Appendix C. Illustrations of the effects of the different optic criteria on the shapes of the local supports S_i can be found in Fries and Matthias (2004).

2.2 Solving the Structural Modeling Problem

Implicit structural modeling consists of finding the unknown values u_i (Eq. 1) constructing an implicit function that represents the domain of study while honoring the input data. For this, the proposed method performs a spatial regression of N_D data points (i.e., values α_k at positions \mathbf{x}_k , $k = 1, \dots, N_D$) with the bending energy penalization (or thin plate energy) (Dubrule 1984; Wahba 1990) by minimizing

$$J(u) = \frac{1}{2} \int_{\Omega} \lambda_{\epsilon}^2 \left(\left(\frac{\partial^2}{\partial x x} u \right)^2 + \left(\frac{\partial^2}{\partial y y} u \right)^2 + 2 \left(\frac{\partial^2}{\partial x y} u \right)^2 \right) d\Omega + \frac{1}{2} \sum_{k=1}^{N_D} \lambda_k^2 (u(\mathbf{x}_k) - \alpha_k)^2, \quad (12)$$

with λ_ϵ the bending penalization weight and λ_k the weight at a data point \mathbf{x}_k . The use of other types of information and the choice of data values α_k are further discussed in Appendix D.

To solve this problem, the domain Ω is regularly divided into N subdomains Ω_l (i.e., $\Omega \subset \cup \Omega_l, \forall l \in N$, Sect. 2.1.1). For simplicity, a regular sampling is used so that all the subdomains Ω_l have the same volume value v (i.e., the volume of a cell of the regular sampling) and are centered on the nodes \mathbf{x}_l . The integration term is approximated as constant in each subdomain Ω_l and evaluated at the center node \mathbf{x}_l (i.e., Gauss quadrature with one point).

The least squares system corresponding to the minimization of Eq. (12) is

$$\begin{bmatrix}
 \lambda_\epsilon \sqrt{v} \partial_{xx}^2 \Phi_1^1 & \dots & \lambda_\epsilon \sqrt{v} \partial_{xx}^2 \Phi_N^1 \\
 \lambda_\epsilon \sqrt{v} \partial_{yy}^2 \Phi_1^1 & \dots & \lambda_\epsilon \sqrt{v} \partial_{yy}^2 \Phi_N^1 \\
 \lambda_\epsilon \sqrt{2v} \partial_{xy}^2 \Phi_1^1 & \dots & \lambda_\epsilon \sqrt{2v} \partial_{xy}^2 \Phi_N^1 \\
 \vdots & & \vdots \\
 \lambda_\epsilon \sqrt{v} \partial_{xx}^2 \Phi_1^N & \dots & \lambda_\epsilon \sqrt{v} \partial_{xx}^2 \Phi_N^N \\
 \lambda_\epsilon \sqrt{v} \partial_{yy}^2 \Phi_1^N & \dots & \lambda_\epsilon \sqrt{v} \partial_{yy}^2 \Phi_N^N \\
 \lambda_\epsilon \sqrt{2v} \partial_{xy}^2 \Phi_1^N & \dots & \lambda_\epsilon \sqrt{2v} \partial_{xy}^2 \Phi_N^N \\
 \\
 \lambda_1 \Phi_1^1 & \dots & \lambda_1 \Phi_N^1 \\
 \vdots & & \vdots \\
 \lambda_{N_D} \Phi_1^{N_D} & \dots & \lambda_{N_D} \Phi_N^{N_D}
 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \\ \lambda_1 \alpha_1 \\ \vdots \\ \lambda_{N_D} \alpha_{N_D} \end{bmatrix}, \tag{13}$$

where $\Phi_l^k = \Phi_l(\mathbf{x}_k)$ are given by Eq. (6) and $\partial_{ij}^2 \Phi_l^k = \frac{\partial^2}{\partial ij} \Phi_l(\mathbf{x}_k)$ are given by Eq. (10). After solving this system, the obtained coefficients u_l are used to evaluate the implicit function. For visualization, the domain Ω is discretized into a grid with N_V points. After evaluating the implicit function on each visualization point of the grid, the horizon lines are extracted bilinearly on each grid element.

2.3 Reference Example of the Presented Method

Figure 1 is a summary of the presented workflow. It is applied on a synthetic cross section of an eroded, faulted and folded domain in two dimensions. Figure 1a shows the reference input data and illustrates the numerical supports of the interpolation. Figure 1b shows the output implicit function with extracted horizons. Equations (1) and (12) are also recalled as they are key to the method.

All the parameters used to create Fig. 1b and their values are given in Table 1. The interpolation nodes used in the computation are actually more numerous than illustrated in Fig. 1a and were generated by using (50×50) nodes in the x and y axes. The spacing between the nodes in the y axis is, therefore, smaller than in the x axis,

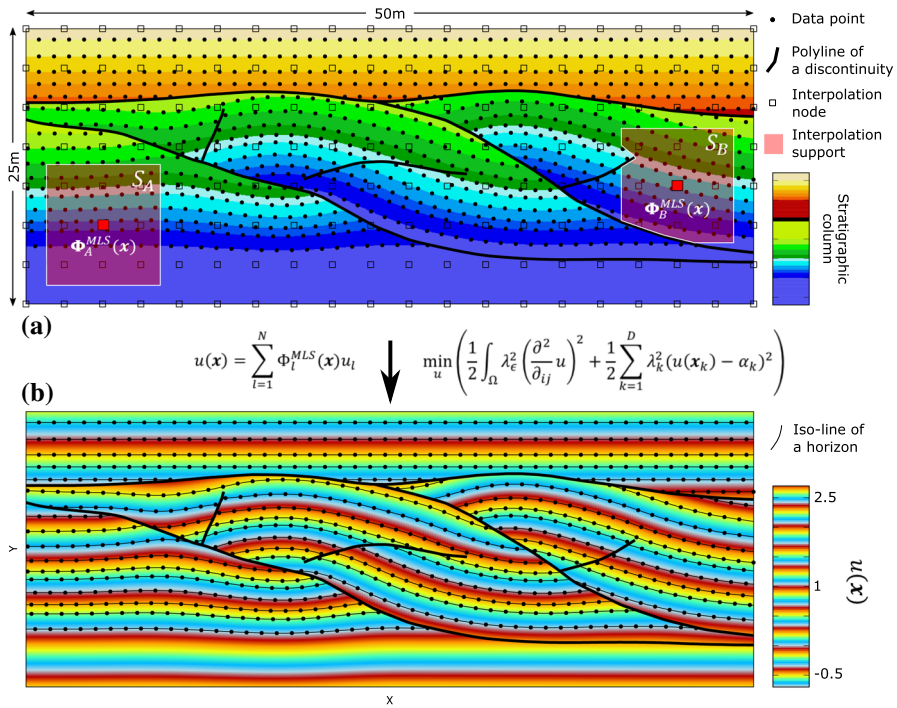


Fig. 1 Schematic representation of the proposed workflow. **a** An input cross section densely and regularly interpreted with data points and fault and unconformity segments in two dimensions. An example of sampling and two local supports S_A and S_B are illustrated. **b** Computed implicit function u obtained with the proposed method using the parameter values of Table 1. Between **a**, **b** The two main equations of the method: the implicit function definition and the problem to solve

which reflects the anisotropy of the studied structures. The support S_B is an example of a support affected by the visibility criterion. The number of visualization points is purposely bigger than the number of nodes and data points [i.e., a grid of (100 × 100) is used] to observe the behavior of the implicit function close to the discontinuities and away from the data points. This is also why the banded color template is used, giving an idea on what other iso-values than the expected horizons would look like if extracted.

The implicit function of Fig. 1b is used as reference model for sensitivity analysis. The parameters and their values are discussed and tested separately. The values given in Table 1 are systematically used as default values for all parameters; only the tested parameter values are changed in each sensitivity test. All the models are run on a laptop with Intel Core i7-4940 3 GHz with 32 Gb of RAM, and running Windows 7 Enterprise 64 bits.

Table 1 Parameters of the proposed method and their default values

Symbol	Meaning	Default value
N	Total number of interpolation nodes	2500
N_V	Total number of visualization points	10,000
N_D	Total number of data points	989
ρ	Dilatation parameter (adimensional)	1.99
d	Polynomial order	1
m	Number of monomials in the polynomial basis	3
α_h	Expected implicit value per horizon	Bottom to top [0, 0.13, 0.21, 0.32, 0.44, 0.51, 0.6, 0.73, 0.86, 1, 2, 2.09, 2.18, 2.27, 2.38, 2.5]
λ_ϵ	Constant weight on energy equations	1
λ_k	Constant weight on data equations	1

3 Sensitivity to Data Quality

3.1 Model Distance and Data Distance

In this section, the method is tested on typical issues with geological data. It focuses on the way data points constrain the modeling problem, considering the impact of their availability, quality, and reliability on the results. Two types of distances are suggested to compare the results: the distance to the reference model D_{model} and the distance of a model to data points D_{data} . The distance D_{model} is only applicable in a synthetic example, whereas the distance D_{data} is applicable in real settings where no reference model is available.

In D_{model} , the tested values are the implicit function values evaluated at the visualization points positions $u(\mathbf{x}_{\text{visu}_i})$. The reference values are the reference model's implicit function values (Fig. 1b) evaluated at the corresponding visualization points: $u_{\text{ref}_i} = u_{\text{ref}}(\mathbf{x}_{\text{visu}_i})$. The distance D_{model} is thus evaluated as

$$D_{\text{model}} = \frac{1}{N_V} \sum_{i=1}^{N_V} \frac{|u(\mathbf{x}_{\text{visu}_i}) - u_{\text{ref}}(\mathbf{x}_{\text{visu}_i})|}{\|\bar{\mathbf{g}}\|_{\text{ref}}}. \quad (14)$$

with $\|\bar{\mathbf{g}}\|_{\text{ref}}$ the average gradient norm of the reference model.

In D_{data} , the tested values are the implicit function values evaluated at the data points positions $u(\mathbf{x}_{\text{data}_i})$ of the currently tested model [e.g., sparse, noisy, depending on the application]. The reference values are the expected data values: $u_{\text{ref}_i} = \alpha_i$ (Sect. 2.2 and Appendix D). The distance D_{data} is thus evaluated as

$$D_{\text{data}} = \frac{1}{N_D} \sum_{i=1}^{N_D} \frac{|u(\mathbf{x}_{\text{data}_i}) - \alpha_i|}{\|\bar{\mathbf{g}}\|_{\text{ref}}}. \quad (15)$$

As it is normalized by the reference gradient norm $\|\bar{g}\|_{ref}$, the evaluated errors are actual distances in meters comparable to the domain’s dimensions [i.e., (50 m × 25 m)], and they are independent of the nodal spacing and the number of evaluated points. They are also independent of the implicit function’s trend, although this is an approximation as local variations of the gradient’s norm in the tested model may overestimate or underestimate the error. As an example, a D_{model} of 1 m indicates that a value evaluated at a visualization point exists, on average, at 1 m in the reference model; a D_{data} of 1 m indicates that the average distance between each data point and the corresponding iso-line is equal to 1 m.

3.2 Data Sparsity

Depending on the types of field samples [e.g., seismic and wells] and the quality of the rock exposure, the interpreted input points for structural modeling may be more or less clustered and sparse. Figure 2a shows the accuracy of the method for different degrees of random decimation in the reference data points (Fig. 1a).

As the decimation is performed randomly, a unique simulation for each data decimation percentage is not enough to understand the dependency of the method to the degree of data sparsity. In this article, when a random parameter is involved, 100 sim-

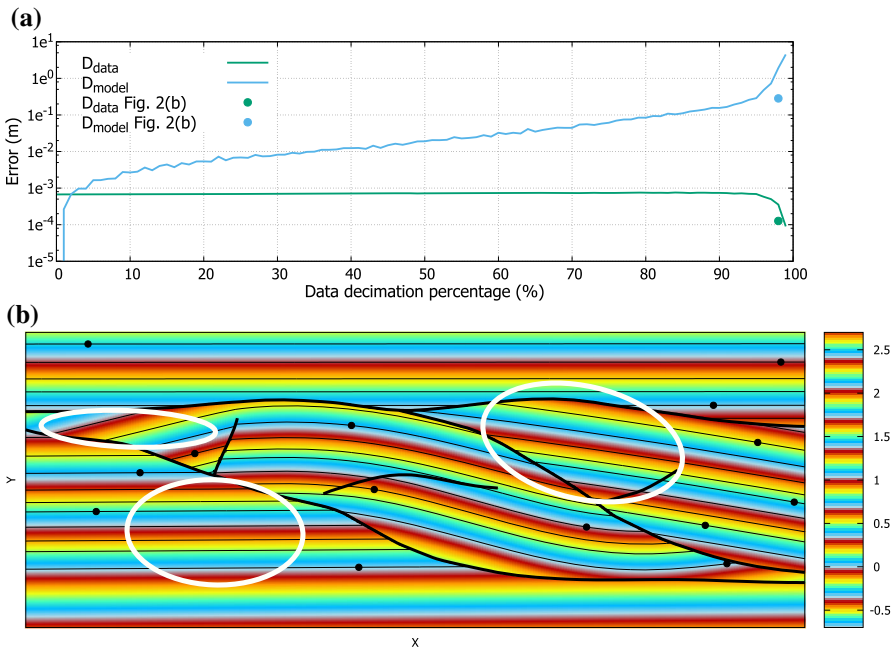


Fig. 2 Sensitivity of the proposed method to irregular data points. **a** Distances to the reference model and the decimated data with a varying random decimation in the data points. **b** A resulting implicit function obtained with a decimation value of 98% [i.e., 14 data points]. The white circles indicate where geological structures were lost as compared to Fig. 1b

ulations are computed for each given set of parameter values. To ease interpretation, only the average distances of the 100 simulations are represented in the graphs. The minimum, maximum, percentiles and standard deviation numbers were also computed and given in ONLINE RESOURCE 1, but not represented for visibility reasons.

In Fig. 2a, it can be observed that the fewer the data points, the larger the distance to the reference model D_{model} . On the contrary, D_{data} is the same for all data decimation values. The method thus fits the decimated data but fails to recover the features of the reference model. The drastic change in the distances for a decimation above 98% [i.e., using 29 data points on average] comes from the emergence of models with less than three data points in one or more fault blocks, which creates unstable results. Figure 2b shows a result with a data decimation value of 98%, which gave, in this random case, 14 input data points. In the circled areas of missing data, the folds are smoothed, but the remainder of the model is well reconstructed. The distances D_{data} and D_{model} of this model are reported in Fig. 2a.

These results show how the proposed method behaves with irregularly and sparsely sampled data. The structures are well represented if the data points sample the non-redundant parts of the geometry. Otherwise, the solution is smoothed where data are missing. Consequently, the proposed method performs best when data points sample high curvature areas and areas of thickness variation.

3.3 Noisy Data

The quality of the field measurements, the processing errors, and interpretation errors can lead to noise in the data. To test the proposed method on this aspect, perturbed data points are created by adding different levels of noise to the reference data (Fig. 1a). The intensity of noise indicates the maximum displacement a point can have during the perturbation, as a radial Euclidian distance in meters around the point. The displacement of each point is sampled from a uniform distribution between zero and this maximum displacement value. Data points having crossed a fault between their initial position and their perturbed position are deleted to avoid stratigraphic inconsistencies.

The noise in the data is handled by the smoothing ability of the bending energy penalization. Figure 3 shows the resulting models for three different values of the smoothness parameter λ_ϵ (Sect. 2.2). It is difficult to have a priori knowledge of a proper λ_ϵ value to use. In addition, the number of equations also have an influence on the results. The terms of volumes v in Eq. (13) obtained during the discretization stage normalize the influence of the number of smoothness equations, but an equivalent principle should also be applied on the N_D data equations. The relation between the smoothness and λ_ϵ is thus independent of N , but dependent to N_D .

Figure 4 shows how the error evolves when the intensity of noise and the smoothing level change. For an intensity of noise fixed to 1 m (Fig. 4a), the best D_{model} value is obtained for a λ_ϵ around 30 (illustrated by Fig. 3b). Below this range, the noisy data points are better represented [i.e., D_{data} decreases], which drives the results away from the reference model [i.e., D_{model} increases, illustrated by Fig. 3a]. Above this range, the structures start to be smoothed, which deteriorates the fitting [i.e., D_{data} and D_{model} slowly increase, illustrated by Fig. 3c]. For a λ_ϵ fixed to 30 (Fig. 4b), both

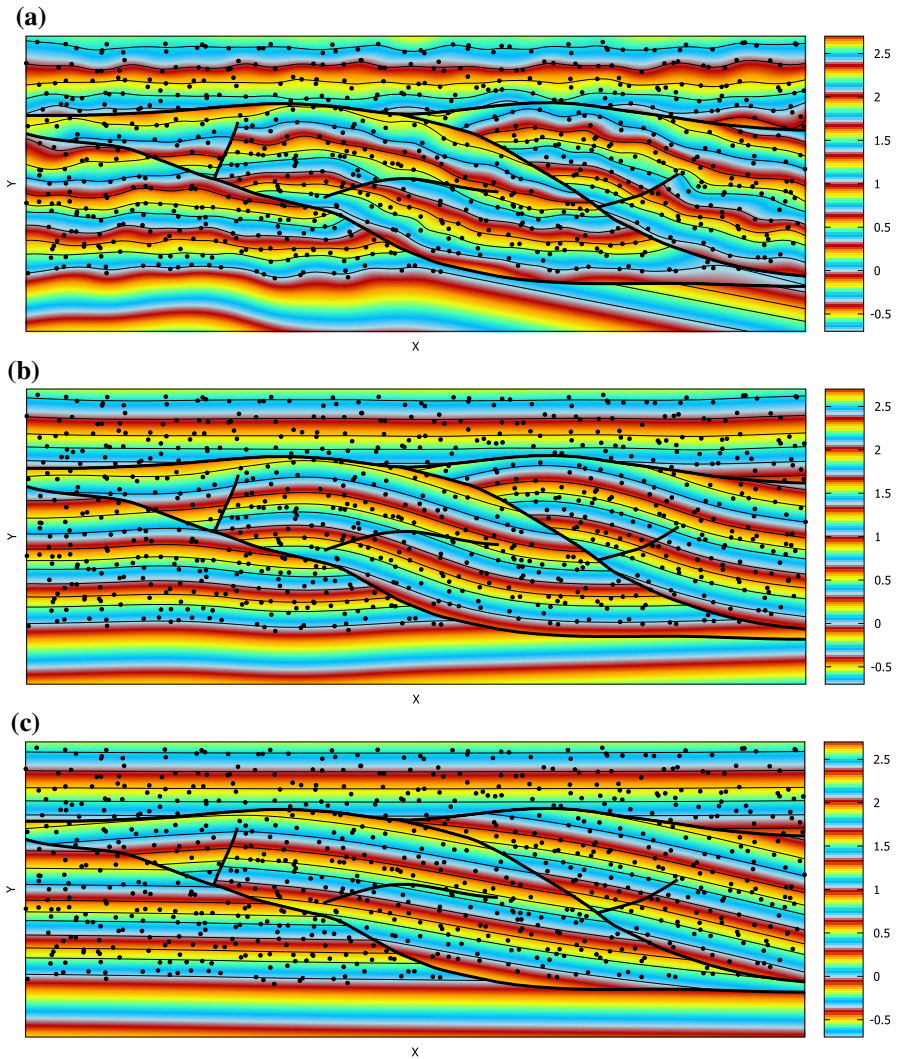


Fig. 3 Resulting implicit function when using the same set of perturbed data as input (intensity of noise is 1 m), but with different energy weights λ_ϵ . **a** $\lambda_\epsilon = 1$. **b** $\lambda_\epsilon = 30$. **c** $\lambda_\epsilon = 400$

D_{data} and D_{model} increase together with the intensity of noise. As the noise increases, the data points represent the geological structures less and less. Although the misfit trend depends on the data distribution, satisfactory results are generally obtained with a λ_ϵ between 1 and 100 for most datasets with a λ_k between 1 and 10.

3.4 Data Reliability

Data points may originate from different types of sources, hence having a varying reliability [e.g., a well datum is generally considered as more reliable than a seismic

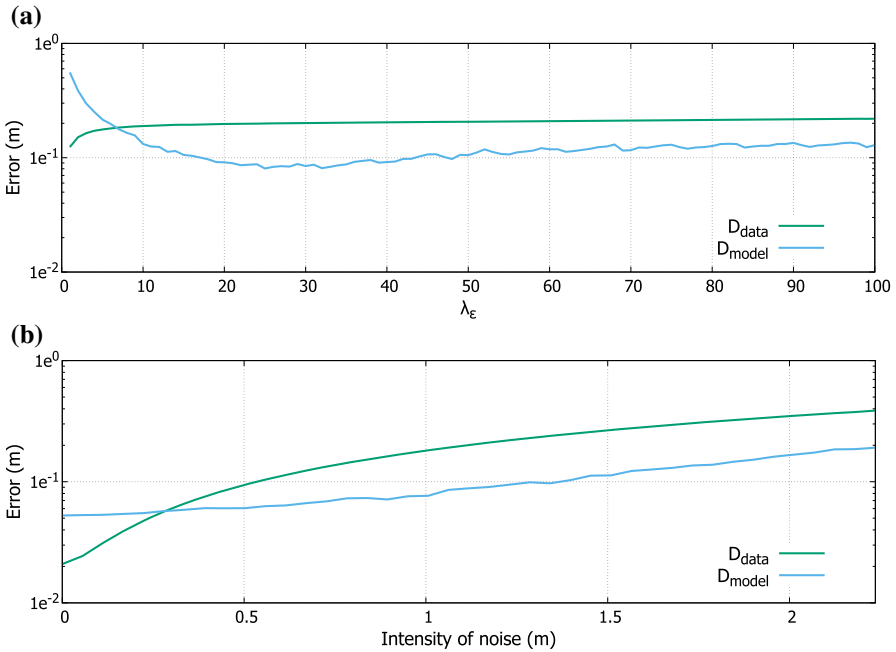


Fig. 4 Sensitivity of the proposed method to the noise and the energy weight λ_ϵ . **a** The intensity of noise is fixed to 1 m, and λ_ϵ varies. **b** The weight λ_ϵ is fixed to 30 and the intensity of noise varies

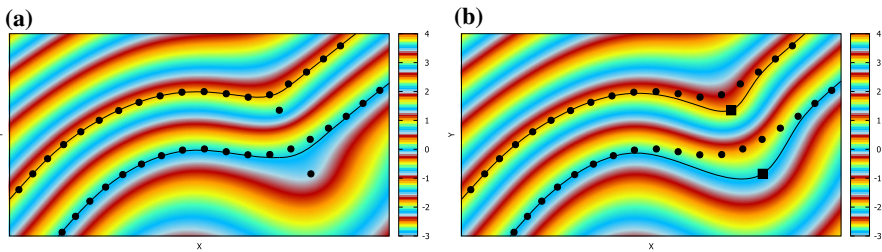


Fig. 5 Example of the influence of data constraints on the implicit function when changing their weights. (Filled square) Hard data: $\lambda_k^{hard} = 10$, (filled circle) Soft data: $\lambda_k^{soft} = 1$

pick]. The data weight λ_k (Sect. 2.2) can be taken differently between data points. As stated in Sect. 3.3, the higher the value of the penalization weight λ_ϵ , the smoother the results. The same principle applies for the data weight λ_k : the higher its value, the better the fit to the corresponding data point. Thus, data reliability can be expressed by varying each data weight λ_k . Figure 5 shows how the relative fit to data can be controlled with λ_k values.

Unfortunately, the weights λ_k have no physical meaning and cannot be associated simply with data geometric errors. In Mallet (2002), the problem is simplified by only distinguishing two types of data: (i) the hard data that must be honored (usually

borehole data), and (ii) the soft data that should be honored as much as possible under the smoothness criterion (usually seismic data).

In the proposed method, an average weight λ_k^{soft} is given to all soft data, and a greater weight λ_k^{hard} is given to all hard data. Deciding the value of λ_k^{hard} , compared to the two weights λ_k^{soft} and λ_ϵ , is not trivial: if taken too small, hard data may not be honored, and if taken too large, System (13) may become badly conditioned. Also, hard data can be honored with a negligible error, but not exactly.

Mallet (2002) avoids these issues by adding a node in the mesh at each hard data position and fixing its nodal coefficient u_1 at the hard data value. This operation may decrease the mesh quality.

With MLS functions, due to the local least squares approximation (Appendix A), imposing a nodal coefficient u_1 is not equivalent to imposing the implicit function value at this interpolation node position \mathbf{x}_1 . MLS functions are said to lack the Kronecker delta property

$$\Phi_1(\mathbf{x}_i) \neq \delta(\mathbf{x}_1 - \mathbf{x}_i) = \begin{cases} 1 & \text{if } l = i \\ 0 & \text{if } l \neq i \end{cases} \Leftrightarrow u(\mathbf{x}_1) \neq u_1. \tag{16}$$

Adding nodes on hard data positions is thus irrelevant to enforce the corresponding constraint. A possible solution would be to use Lagrange multipliers to automatically determine appropriate weights for hard data, but it would change the problem into a saddle-point one, thus modifying its complexity (Brezzi 1974). Though strongly dependent on the case study, a hard data weight λ_k^{hard} ten times greater than the soft data weight λ_k^{soft} is sufficient in practice and avoids a badly conditioned system (Fig. 5).

Figure 6 summarizes the presented data constraints and their influence on the implicit function. The soft data were obtained using a decimation of 98% and an intensity of noise of 1 m, generating thirteen noisy points. The hard data were obtained using a decimation of 99%, generating five points positioned as in the reference data. Additionally, five gradient data vectors (constraint presented in Appendix D) were extracted on the implicit function from the reference model to control the structures

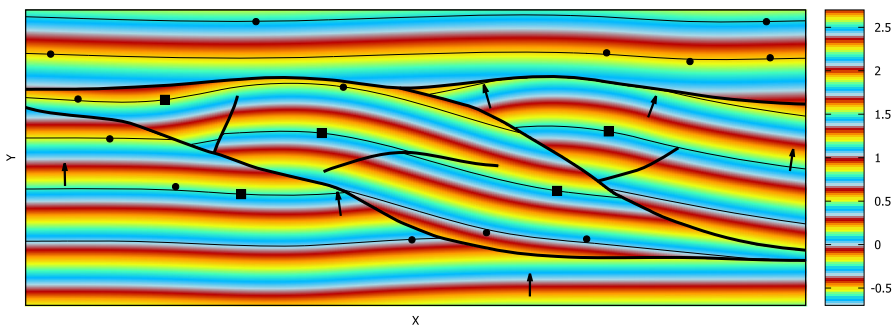


Fig. 6 A resulting implicit function using sparse and heterogeneous data with noise in the soft data (intensity of noise: 1 m) and a few hard and gradient data points. (Filled square) Hard data: $\lambda_k^{\text{hard}} = 10$, (filled circle) Soft data: $\lambda_k^{\text{soft}} = 1$, \rightarrow Gradient data: $\lambda_k^{\text{grad}} = 10$, $\lambda_\epsilon = 30$. $D_{\text{data}} = 8.3e^{-3}$ m and $D_{\text{model}} = 2.9e^{-2}$ m

in areas with missing data. With a total of 23 constraints, this model is closer to the reference model than those generated with a decimation of 60% on the reference data and above [i.e., less than about 395 data points] from Fig. 2a.

4 Discussions and Perspectives

4.1 Complexity and Stability of the Moving Least Squares Functions

4.1.1 Polynomial Order of the MLS Functions

The polynomial order d defines, together with the dimension of space (e.g., two here), the number of monomials m in the polynomial basis. This number m describes the dimensions of the moment matrix \mathbf{A} [i.e., $\dim(\mathbf{A}) = (m \times m)$]. It thus has an influence on the complexity of the method and must be chosen small enough to avoid unnecessary computational costs. In practice, Fig. 7a shows that MLS functions with an order of 0 are not enough to reproduce a geological model: the solution tends to be perpendicular to the discontinuities and the domain's borders. MLS functions with an order of 2 give results similar to an order of 1 (Fig. 7b), with a higher computational complexity. The

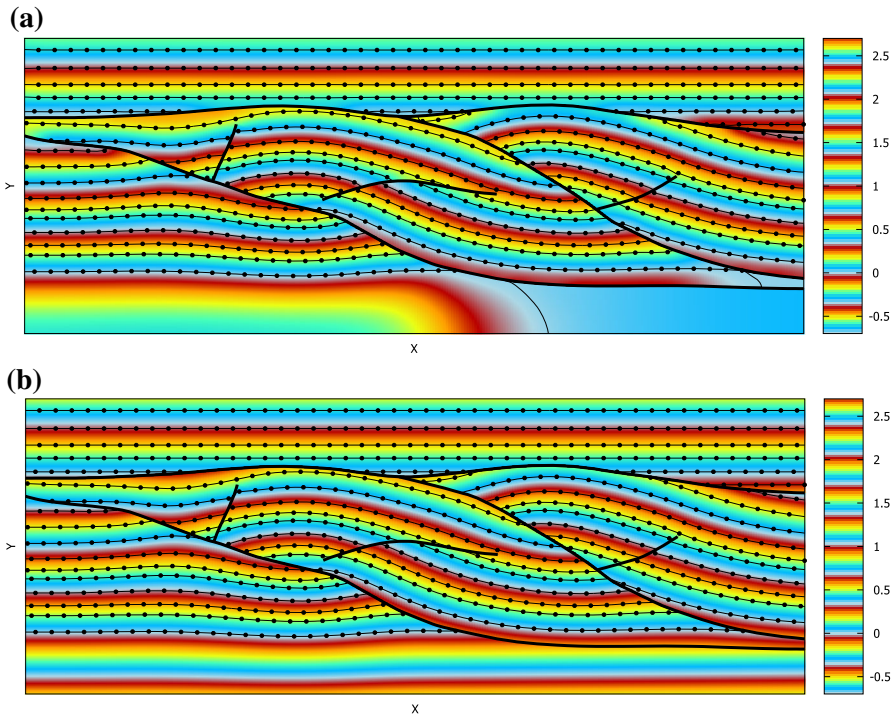


Fig. 7 Resulting implicit function with different polynomial orders d for the MLS shape functions and adapted dilatation values ρ . **a** $d = 0$ and $\rho = 1.99$. **b** $d = 2$ and $\rho = 2.99$

order d is, therefore, fixed to 1 in the presented method. Although purely empirical, first-order MLS functions seem to reproduce any complex geological structures.

4.1.2 Local Supports and Domain Coverage

An MLS shape function $\Phi_1(\mathbf{x})$ is only defined within the support S_1 of its weight function w_1 (Eq. 6). By definition, S_1 and the support of $\Phi_1(\mathbf{x})$ are the same. The node \mathbf{x}_1 has an influence on the points existing within the support S_1 . If a point \mathbf{x} is influenced by a node \mathbf{x}_1 , then \mathbf{x}_1 is said to be a neighbor of \mathbf{x} .

This restricted influence of the nodes \mathbf{x}_1 in space represents the main advantage of the MLS functions as each constraint involves a small number of neighbors [i.e., System (13) is sparse]. Unfortunately, it may also lead to singularities. Let the intersection of all supports S_1 be denoted as the cover. When the entire domain Ω is included in the cover, it is said to be complete. If the cover is not complete, then the implicit function $u(\mathbf{x})$ is undefined in the uncovered areas. In addition, the number of linearly independent neighboring nodes required around a position \mathbf{x} should be equal at least to the number of monomials m in the MLS polynomial basis to have a non-singular matrix \mathbf{A} (Appendix A). Therefore, the distribution of the nodes \mathbf{x}_1 and the size of the supports S_1 , controlled by the dilatation parameters ρ , must be defined carefully to avoid singularities. This situation is analogous to ordinary and universal kriging when the neighborhood size is too small.

4.1.3 Theoretical Relationship Between the Dilatation Parameter and the Number of Neighbors

The cover problem (Sect. 4.1.2) is tackled with squared supports (Eq. 2) and by making the dilatation parameter ρ proportional to the regular spacing of interpolation nodes. Figure 8a illustrates the relationship between ρ and the resulting support of an MLS function centered on a given interpolation node \mathbf{x}_1 . If close to a discontinuity with the visibility criterion (Sect. 2.1.3) or close to a border, the support may cover fewer neighboring nodes than illustrated. The dilatation parameter ρ has thus a direct impact on the maximum number of neighboring interpolation nodes n_{node}^{\max} around a given node \mathbf{x}_1 , following

$$n_{\text{node}}^{\max} = (2\lfloor \rho \rfloor + 1)^2, \quad (17)$$

with $\lfloor \cdot \rfloor$ the integer part operator.

As ρ is constant for all nodes, if an MLS support centered on a node \mathbf{x}_1 covers a point \mathbf{x} , then an imaginary MLS support centered on \mathbf{x} covers \mathbf{x}_1 . Figure 8b illustrates the relationship between ρ and the support of an MLS function centered on a data point \mathbf{x} , defining the neighboring nodes influencing this data. Following the comments on discontinuities and borders, the relationship between ρ and the maximum number of neighboring interpolation nodes n_{data}^{\max} around a given data point \mathbf{x} is

$$n_{\text{data}}^{\max} = (2\lfloor \rho + 0.5 \rfloor)^2. \quad (18)$$

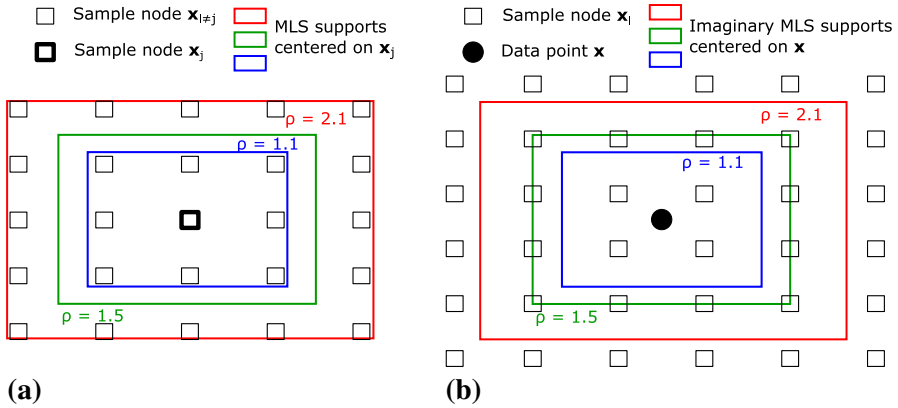


Fig. 8 Illustration of the relationship between the dilatation parameter ρ , proportional to the regular spacing, and the MLS functions supports. **a** The supports for three different ρ values centered on a node x_j . **b** The imaginary supports for the three same ρ values centered on a data or visualization point x

Equation (18) is only true if the data point is not exactly located on a node position or not colinear with two nodes in the x or y axis. The maximum number of neighboring nodes of a data point is therefore between n_{data}^{max} and n_{node}^{max} depending on its location.

The numbers n_{node}^{max} and n_{data}^{max} must be considered when defining ρ , as they give an idea of the sparsity of System (13). In addition, both must at least be greater than the number of monomials m , which is given by the dimension of space and the polynomial order d (Sect. 4.1.2).

4.1.4 Practical Influence of the Support Size on the Method

Even if the theory defines a minimum value for the dilatation parameter ρ compared to the MLS parameter m (Sect. 4.1.3), this minimum value is not necessarily reliable when discontinuities are present. This is illustrated in Fig. 9 where the influence of ρ on the method is given for a polynomial order of 1 [i.e., $m = 3$ in two dimensions].

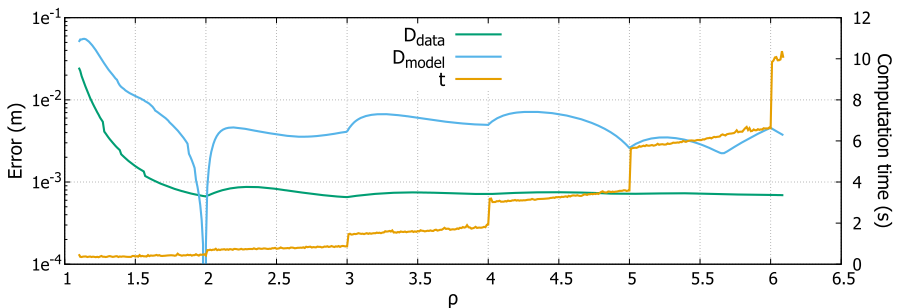


Fig. 9 Influence of the support size ρ on the method when ρ is proportional to the regular spacing and with a polynomial order d of 1

If the moment matrix \mathbf{A} is singular at a position \mathbf{x} , the implicit function u cannot be evaluated. In this case, the concerned point [e.g., data or visualization point] is avoided during the computation of distances (Sect. 2.3) and flagged as undefined. Such anomalies are not explicitly represented on Fig. 9 for visibility reasons, but their numbers per model can be found in ONLINE RESOURCE 1.

Some undefined data points (up to 14) are found in models with a ρ value between 1 and 1.5 even though these ρ values are theoretically large enough to invert the moment matrix \mathbf{A} (i.e., $n_{\text{data}}^{\text{max}} = 4 > 3$, $n_{\text{node}}^{\text{max}} = 9 > 3$). This is due to the visibility criterion [Sect. 2.1.3], which reduces the number of neighbors by cutting the supports near the discontinuities. In this case, the number $n_{\text{data}}^{\text{max}}$ is too close to m and the number of actual neighbors is very likely to drop under m if a data point is close to a discontinuity. No other undefined points are found in the other models of Fig. 9.

For ρ values greater than 1.5, D_{data} and D_{model} present an error of a few millimeters. Such differences can be considered negligible when considering that the model's folds and faults are several meters long. The method thus converges for a fixed number of nodes and an increasing support size. The perfect fit to the reference model (i.e., $D_{\text{model}} = 0$ around a ρ value of 1.99) is caused by the exact equivalence with the reference parameters (Table 1, default values).

Concerning the computational time t , it increases incrementally with ρ . The computational cost of the method is, therefore, dependent on $n_{\text{node}}^{\text{max}}$ as each unit of ρ defines a different number of neighboring nodes (Eq. 17). It is also dependent on $n_{\text{data}}^{\text{max}}$ (Eq. 18). It thus seems unnecessary to define ρ greater than 2 as the same results are obtained with a slower computation. Also, the fit to data is slightly better with a ρ around 2 than with smaller values.

In conclusion, MLS functions are stable even with an increasing number of neighbors. The dilatation parameter ρ should be taken as close as possible to 2 when using a polynomial order of 1 to avoid unnecessary computational cost while obtaining similar results. It is fixed to 1.99 in the presented method (Table 1) to avoid dealing with neighbors exactly on the edge of the supports (i.e., neighbors with no influence). When using a greater polynomial order, the ρ value must be increased accordingly, which is why a ρ equal to 2.99 was used for a polynomial order d equal to 2 in Fig. 7b.

4.2 Regular or Irregular Sampling

4.2.1 Comparison Test with Random Sampling

Distributing the interpolation nodes regularly (Sect. 2.1.1) is not a requirement of the proposed method, but has several advantages. In Fig. 10, the method is tested with a varying number of interpolation nodes, which are distributed either randomly or regularly in the domain of study. The randomly generated nodes follow a uniform distribution on the x and y axes, respectively. In this case, the dilatation parameters ρ_{1x} and ρ_{1y} cannot be specified relatively to the interpolation node spacing [e.g., using ρ as in Eq. (3)]. Therefore, they are fixed for all simulations, regardless the number of nodes N and the sampling technique. In this application, ρ_{1x} is fixed to 3.5 m, and ρ_{1y} ,

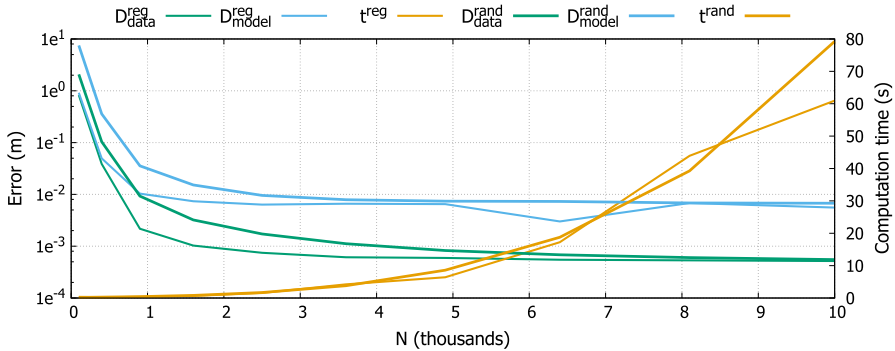


Fig. 10 Influence of the number of interpolation nodes N on the presented method with regular and random sampling. The distances and the computation time are evaluated for both random and regular sampling separately

is fixed to 1.75 m, so that the supports S_l of the interpolation functions cover $\approx 0.5\%$ of the domain Ω .

The model distance D_{model} and data distance D_{data} (Sect. 2.3) are represented respectively by $D_{\text{model}}^{\text{reg}}$ and $D_{\text{data}}^{\text{reg}}$ for regular sampling, and $D_{\text{model}}^{\text{rand}}$ and $D_{\text{data}}^{\text{rand}}$ for random sampling. The method's computation time is also represented for both techniques as t^{reg} and t^{rand} . All the simulations results together with basic statistics on the simulations are given in ONLINE RESOURCE 1.

Figure 10 emphasizes several characteristics of the method: (i) the results are dependent on the interpolation node sampling; (ii) both methods converge to the reference model when the number of interpolation nodes increases; (iii) regular sampling gets models closer to the reference model and the data set than the average random sampling for all N ; and (iv) the computational efficiencies of the two methods are equivalent. As observed in the full results (ONLINE RESOURCE 1), the two methods can generate undefined points for small numbers of nodes N (up to 900; small compared to the used dilatation parameters), but these anomalies are more represented in random sampling than in regular sampling.

This study shows that, for a given set of dilatation parameters ρ_{1x} and ρ_{1y} , both sampling techniques obtain close results as long as a minimum number of interpolation nodes N is used. The main difference is that the number of nodes N and the dilatation parameters ρ_{1x} and ρ_{1y} can be theoretically correlated to avoid singularities with regular sampling (Sect. 4.1.3), which is not the case with random sampling. In practice, this correlation can also avoid unnecessary large supports of interpolation and thus reduce drastically the computation time with regular sampling (Sect. 4.1.4). As a reference, it takes 0.9 s to generate a model with 10,000 nodes and a dilatation parameter ρ of 1.99 scaled on the nodal spacing (Eq. 3); the evaluated distances are $D_{\text{data}} = 5.14e^{-4}$ m and $D_{\text{model}} = 1.56e^{-3}$ m.

4.2.2 Possible Optimizations on Regular Sampling

Regular sampling could be accelerated by optimization techniques not implemented in this paper. Once an adapted dilatation parameter value ρ has been chosen for a given sampling resolution (Sect. 4.1.4), several repetitive calculations can be avoided. For instance, all the nodes far enough from the borders and the discontinuities (i.e., not affected by the visibility criterion, Sect. 2.1.3) have the same pattern of neighbors (Fig. 8a). The MLS second derivatives evaluated at these nodes are therefore the same. It is possible to evaluate these derivatives once, store the results, and use them for all nodes with the same pattern. The nodes close to the borders, but far from the discontinuities also follow patterns simple enough to be stored. This principle can even be extended for all possible neighboring configurations, but the number of tests to find the right pattern may then become computationally demanding.

Another possible improvement is to approximate the evaluation of the MLS functions on data and visualization points. When far from borders and discontinuities, the disposition of the neighboring nodes around a point varies continuously (Fig. 8b). This variation is restricted to the containing cell (i.e., the square area between 4 sampling nodes). Each MLS function Φ_1 could thus be approximated by studying their evolution depending on the position \mathbf{x} within a cell. The cells are implicit (i.e., not stored) as the sampling is regular. The previous comments for nodes close to the borders also apply for this suggestion on points.

Finally, the presented method is adapted for parallelization, as the equations written in System (13) are independent from one another. When considering a node or a data point, the set of neighbors is defined with the chosen support size and proximity to discontinuities; the MLS functions or their second derivatives are then evaluated; and the corresponding equation can be written in the system. Each of these steps is only dependent on the interpolation node or data point of the concerned equation. In addition, and contrarily to mesh based methods, handling the discontinuities with the visibility criterion (Sect. 2.1.3) does not require any preprocessing on the sampling, but only intersection tests between segments (and triangles in three dimensions). Therefore, each equation in System (13) can be written in parallel.

4.3 Complex Geometries of Structural Discontinuities

4.3.1 Lack of Neighboring Nodes

Although the visibility criterion (Sect. 2.1.3) is criticized for stability reasons (Belytschko et al. 1996), it shows satisfactory results in the presented application. The main issue is the modification of the set of neighboring nodes. Cutting the supports decreases the number of neighbors, which can produce areas where the estimation is impossible (Sect. 4.1.4) or with singularities.

On Fig. 11a, singularities and undefined values at the intersection of two faults are represented. This is due to a lack of neighbors on visualization points, making the moment matrix $\mathbf{A}(\mathbf{x})$ singular in the concerned area (Sect. 4.1.2). In Fig. 11b, in addition to singularities, the generated implicit function bends abnormally away from

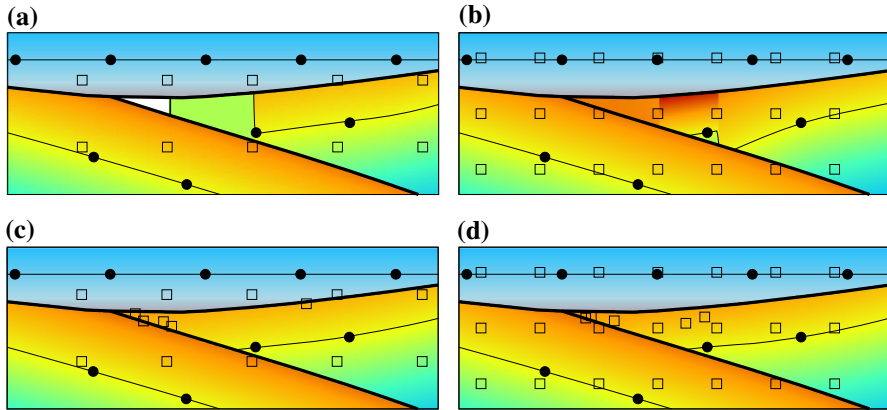


Fig. 11 Limits to the visibility criterion and a proposed correction. **a** Case of singularities and undefined values at visualization point positions when neighboring interpolation nodes are missing. **b** Case of badly evaluated nodal coefficients u_i when neighboring interpolation nodes are unevenly distributed. **c, d** Corrections to the limits observed in **(a, b)** by randomly generating new neighbor nodes

the intersection. The unevenly distributed nodes also have deteriorated the evaluation of the second derivatives, which has impacted the solution coefficients u_i attached to the concerned nodes \mathbf{x}_i .

Those results were generated with the default parameters (Table 1) and smaller numbers of interpolation nodes N . The described issues are thus related to the resolution of the sampling, but also to the discontinuities, their geometries and interactions. As undefined values are not acceptable in structural modeling, a solution is to use a finer resolution for the sampling, or a greater value for the dilatation parameter. The computing times given in Figs. 9 and 10 show that both solutions are possible but costly.

4.3.2 Changing the Polynomial Order

An alternative strategy to address the lack of neighbors could be to locally reduce the polynomial order d : if the number of interpolation neighbors of a point is smaller than the chosen number m , d can be decreased accordingly.

Unfortunately, this solution is not applicable as it does not solve situations with no neighbors (Fig. 11a) or with unevenly distributed neighbors (Fig. 11b). Also, decreasing d to the zeroth order, if necessary, is not adapted to a structural modeling application (Sect. 4.1.1, Fig. 7).

4.3.3 Generation of New Nodes

In Sect. 4.2.1, the proposed method is shown to converge to the same model whether the interpolation nodes are distributed randomly or regularly and given a sufficient number of interpolation nodes. This means that, for a position \mathbf{x} , the evaluation of the MLS functions and their derivatives should be approximately the same if there are

enough neighbors, even if randomly distributed. The lack of neighboring nodes can, therefore, be solved by randomly generating new interpolation nodes locally.

If a position \mathbf{x} (interpolation node, data point or visualization point) is close to a discontinuity, the following procedure can be employed:

1. Count the number of neighboring nodes $n_{\mathbf{x}}$.
2. Compare this number to a reference number n_{ref} for which a shape function is considered stable.
3. Add neighbors randomly until $n_{\mathbf{x}} = n_{\text{ref}}$ within the support.

Figure 11c, d show how this technique solves the two problems exposed in Sect. 4.3.1 respectively with a number of reference n_{ref} fixed to 4. Nodes generated on the other side of the faults are deleted during the generation as they are not considered neighbors (i.e., not within the support). The procedure could be improved by some node placement strategies, such as using a repulsion factor allowing to generate nodes evenly around a position \mathbf{x} . This could increase the chances to obtain a stable interpolation in those areas with a small n_{ref} number, which is not guaranteed in the present approach. Adding interpolation nodes changes the problem's dimensions and its density (Sect. 2.2), but this technique does not change drastically the computational efficiency of the algorithm as the modifications are local.

4.4 Bending Energy and Structural Modeling

Structural modeling algorithms are supposed to create models that represent geological structures. As a smooth energy, the bending energy has been presented as robust enough to deal with sparse, irregular and noisy data (Sect. 3). This ability is key for complex applications.

Unfortunately, smoothing methods perform badly on poorly sampled anisotropic, periodic and thickness variation features. A common strategy is to add artificial data for these issues: orientation and gradient data are typically used to control a fold geometry away from the data (Hillier et al. 2014; Laurent 2016) or to include a known periodicity in fold and foliation structures (Grose et al. 2017; Laurent et al. 2016). Similar constraints could also be used to impose some thickness variations. Such approaches often require expertise and manual interactions.

Other approaches intend to consider the structural anisotropy calibrated from the available data into the smoothing regularization itself. This is generally included in PFM with the experimental variogram, modifying the covariance and its range (Aug 2004). Gonçalves et al. (2017) further reduce the workload by using the maximization of the log-likelihood to automatically determine such parameters. However, the global interpolation of PFM limits the application of such techniques when assessing local anisotropy. In volume contouring, Martin and Boisvert (2017) infer the local anisotropy of the targeted geobodies by iteratively adapting partitions of the domain and local interpolations per partition.

The presented formalism introduces the concept of continuous energies in a local approach well adapted to the structural modeling application. While some versions of PFM intrinsically minimize the bending energy using Green's functions as interpolants, it is explicitly minimized in the least squares sense in the proposed approach.

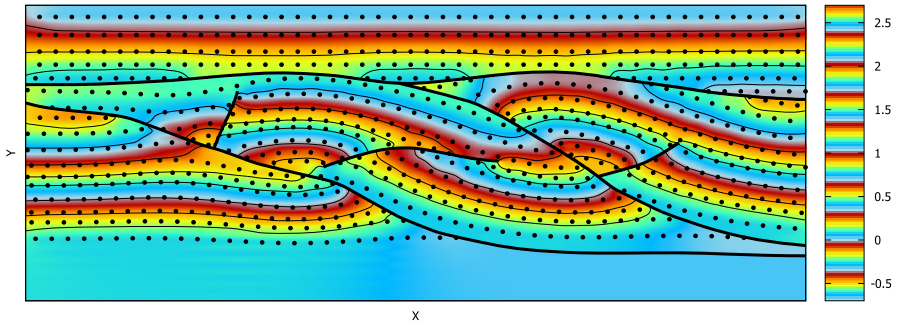


Fig. 12 Resulting implicit function using the Dirichlet energy instead of the bending energy

It is then closer to DSI, although the regularization term is posed as a continuous, and not discrete, operator. With the proposed formalism, different energies and their ability to solve specific geological features can be tested by changing the regularization term in Eq. (12) and discretizing it accordingly in System (13). It is also possible to mix several energies and even tune the weight λ_ϵ to each derivative term separately and spatially.

As an example, Fig. 12 shows how the algorithm behaves when using the Dirichlet energy (Courant 1950) as

$$J_{\text{Dirichlet}}(u) = \frac{1}{2} \int_{\Omega} \lambda_\epsilon^2 \|\nabla u\|^2 d\Omega. \quad (19)$$

The obtained implicit function tends to be constant where data are missing and perpendicular to the discontinuities. Therefore, the Dirichlet energy is not adapted to the structural modeling problem. The best continuous energy fitting geological structures is yet to be found, and the proposed framework may be used to test new ideas.

5 Conclusions

In this paper, an alternative to the existing implicit structural modeling methods is proposed. It is a locally defined method that handles the structural discontinuities with meshless concepts. The implicit function is defined as a weighted sum of moving least squares shape functions (Lancaster and Salkauskas 1981; McLain 1976). The supports of these interpolation functions are centered on regularly distributed nodes and cut by the structural discontinuities with the visibility criterion. Therefore, no mesh needs to be stored and the system to solve is sparse.

The proposed method consists of a spatial regression of data points penalized by a physical energy. It thus introduces the explicit use of continuous energy to solve the modeling problem. In this paper, the bending energy (Dubrule 1984; Wahba 1990), which has the ability to filter noise, is suggested to extrapolate between data gaps and handle clustered and sparse data by smoothing the generated structures. Based on this formalism, it is possible to test other energies and mix them to better relate geology.

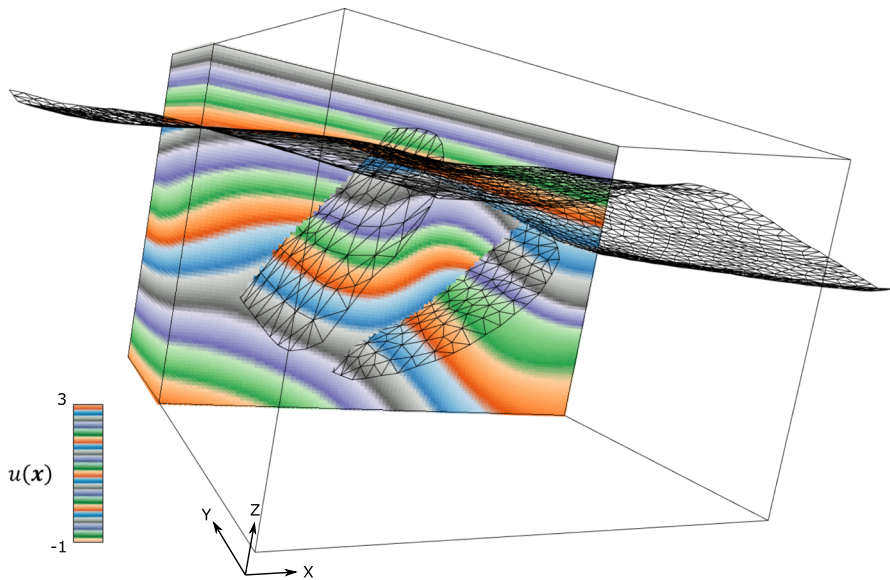


Fig. 13 Resulting implicit function on a synthetic model in three dimensions using 8815 data points with two faults, one erosion, folds, and thickness variations. The discontinuity objects are the transparent triangulated surfaces. The implicit function was computed with $(40 \times 40 \times 30)$ nodes in 25 s

In addition, the related continuous equations can be discretized in many other ways than the one presented here, centering the modeling problem on the choice of the regularization and not the discretization itself. Although only the typical smoothing assumption of structures is adopted here, the introduced concept may have the potential to enable inherent control of complex geological cases.

The sensitivity tests on data quality and the method's parameters are presented in two dimensions. This provides knowledge on the relations between results and parameters also for other models. Only two-dimensional results are shown for simplicity, but the suggested formalism is applicable to higher dimensions. Figure 13 shows a result on a synthetic three-dimensional model.

Acknowledgements The authors would like to thank Bruno Lévy for fruitful discussions on continuous equations behind implicit structural modeling. The authors would also like to thank the four anonymous reviewers whose contributions have greatly improved the paper, especially on the metric of distances between models and data. The Ph.D. thesis of Julien Renaudeau is funded by both Schlumberger and the Association nationale recherche technologique. These organizations are also gratefully acknowledged.

Appendix A: Moving Least Squares Demonstration

This section gives the construction details of the moving least squares (MLS) functions (Lancaster and Salkauskas 1981; McLain 1976) defined in Eq. (6). We use the demonstration from Nguyen et al. (2008) and clarify the approximation of the coefficients $\mathbf{a}(\bar{\mathbf{x}})$ around a fixed position $\bar{\mathbf{x}}$.

The implicit function $u(\mathbf{x})$ is redefined as a polynomial function of degree d with spatially varying coefficients $a_j(\mathbf{x})$ as

$$u(\mathbf{x}) = \sum_{l=1}^N \Phi_l(\mathbf{x}) u_l = \sum_{j=1}^m p_j(\mathbf{x}) a_j(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \cdot \mathbf{a}(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega, \quad (20)$$

with m the number of monomials $p_j(\mathbf{x})$ and coefficients $a_j(\mathbf{x})$. As all the coefficients $a_j(\mathbf{x})$ vary continuously in space, they must be determined for any position \mathbf{x} .

Let $\bar{\mathbf{x}}$ be a fixed position in Ω . If the coefficients $a_j(\bar{\mathbf{x}})$ were to be determined, they could be used to evaluate $u(\mathbf{x})$ at a position \mathbf{x} around $\bar{\mathbf{x}}$ with a non-zero error as

$$u(\mathbf{x}) \approx \mathbf{p}^T(\mathbf{x}) \cdot \mathbf{a}(\bar{\mathbf{x}}), \quad \forall (\mathbf{x}, \bar{\mathbf{x}}) \in \Omega. \quad (21)$$

Equation (21) can be written for all interpolation nodes \mathbf{x}_l as

$$u(\mathbf{x}_l) \approx \mathbf{p}^T(\mathbf{x}_l) \cdot \mathbf{a}(\bar{\mathbf{x}}), \quad \forall l \in N, \quad \forall \bar{\mathbf{x}} \in \Omega, \quad (22)$$

where the implicit function $u(\mathbf{x}_l)$ is supposed to be as close as possible to the nodal value u_l as

$$u(\mathbf{x}_l) \approx u_l. \quad (23)$$

Equations (22) and (23) are combined for each node \mathbf{x}_l , defining the system of N equations and m unknowns

$$\begin{aligned} \mathbf{p}^T(\mathbf{x}_1) \cdot \mathbf{a}(\bar{\mathbf{x}}) &\approx u_1 \\ \mathbf{p}^T(\mathbf{x}_2) \cdot \mathbf{a}(\bar{\mathbf{x}}) &\approx u_2 \\ &\vdots \\ \mathbf{p}^T(\mathbf{x}_N) \cdot \mathbf{a}(\bar{\mathbf{x}}) &\approx u_N \end{aligned} \quad (24)$$

The pertinence of each approximation in System (24) is related to the distance between each \mathbf{x}_l and $\bar{\mathbf{x}}$, respectively: as the coefficients $a_j(\mathbf{x})$ vary continuously in space, the further \mathbf{x}_l is to $\bar{\mathbf{x}}$, the less reliable is the approximation. Each approximation is thus weighted by a continuous function $w_l(\bar{\mathbf{x}})$, centered on each position \mathbf{x}_l and becoming nil when \mathbf{x}_l is far from $\bar{\mathbf{x}}$ [Sect. 2.1.2, Eqs. (2) and (4)]. Determining the coefficients $a_j(\bar{\mathbf{x}})$ is thus equivalent to minimizing, in a least squares sense, the functional $J_{\text{MLS}}(\mathbf{a}(\bar{\mathbf{x}}))$ written as

$$J_{\text{MLS}}(\mathbf{a}(\bar{\mathbf{x}})) = \sum_{l=1}^n w_l(\bar{\mathbf{x}}) \left(\mathbf{p}^T(\mathbf{x}_l) \cdot \mathbf{a}(\bar{\mathbf{x}}) - u_l \right)^2, \quad (25)$$

with n the number of nodes \mathbf{x}_l close enough to $\bar{\mathbf{x}}$ to have a non-zero weight function $w_l(\bar{\mathbf{x}})$ (i.e., $n \leq N$). If the number of neighboring nodes n is smaller than the number of unknowns m , the minimization of Eq. (25) has no solution. Minimizing J_{MLS} preserves Eq. (23) as an approximation, which is why MLS functions do not have the Kronecker delta property (Eq. 16).

An extremum of this functional is obtained by setting the derivative of J_{MLS} with respect to $\mathbf{a}(\bar{\mathbf{x}})$ equal to zero

$$\begin{aligned} \sum_{l=1}^n w_l(\bar{\mathbf{x}}) 2p_l(\mathbf{x}_l) (\mathbf{p}^T(\mathbf{x}_l) \cdot \mathbf{a}(\bar{\mathbf{x}}) - u_l) &= 0 \\ \sum_{l=1}^n w_l(\bar{\mathbf{x}}) 2p_l(\mathbf{x}_l) (\mathbf{p}^T(\mathbf{x}_l) \cdot \mathbf{a}(\bar{\mathbf{x}}) - u_l) &= 0 \\ &\vdots \\ \sum_{l=1}^n w_l(\bar{\mathbf{x}}) 2p_m(\mathbf{x}_l) (\mathbf{p}^T(\mathbf{x}_l) \cdot \mathbf{a}(\bar{\mathbf{x}}) - u_l) &= 0 \end{aligned} \tag{26}$$

which can be written in matrix form

$$\sum_{l=1}^n w_l(\bar{\mathbf{x}}) \mathbf{p}(\mathbf{x}_l) \cdot \mathbf{p}^T(\mathbf{x}_l) \cdot \mathbf{a}(\bar{\mathbf{x}}) = \sum_{l=1}^n w_l(\bar{\mathbf{x}}) \mathbf{p}(\mathbf{x}_l) u_l, \tag{27}$$

or, more compactly,

$$\mathbf{A}(\bar{\mathbf{x}}) \cdot \mathbf{a}(\bar{\mathbf{x}}) = \mathbf{B}(\bar{\mathbf{x}}) \cdot \mathbf{u}, \tag{28}$$

with $\mathbf{A}(\bar{\mathbf{x}}) = \sum_{l=1}^n w_l(\bar{\mathbf{x}}) \mathbf{p}(\mathbf{x}_l) \cdot \mathbf{p}^T(\mathbf{x}_l)$ and $\mathbf{B}(\bar{\mathbf{x}}) = [w_1(\bar{\mathbf{x}}) \mathbf{p}(\mathbf{x}_1), w_2(\bar{\mathbf{x}}) \mathbf{p}(\mathbf{x}_2), \dots, w_n(\bar{\mathbf{x}}) \mathbf{p}(\mathbf{x}_n)] = [\mathbf{B}_1(\bar{\mathbf{x}}), \mathbf{B}_2(\bar{\mathbf{x}}), \dots, \mathbf{B}_n(\bar{\mathbf{x}})]$.

Solving the least squares system defined by Eq. (28) recovers the coefficients $a_j(\bar{\mathbf{x}})$ for a given position $\bar{\mathbf{x}}$. As $\bar{\mathbf{x}}$ can be fixed anywhere in the domain Ω , the coefficients $a_j(\mathbf{x})$ can be determined at any position \mathbf{x} in Ω by

$$\mathbf{A}(\mathbf{x}) \cdot \mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x}) \cdot \mathbf{u}. \tag{29}$$

Incorporating Eq. (29) in Eq. (20) leads to

$$u(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \cdot [\mathbf{A}(\mathbf{x})]^{-1} \cdot \mathbf{B}(\mathbf{x}) \cdot \mathbf{u}, \tag{30}$$

with

$$u(\mathbf{x}) = \Phi^T(\mathbf{x}) \cdot \mathbf{u}. \tag{31}$$

Hence the Eq. (6)

$$\Phi_1(\mathbf{x}) = w_1(\mathbf{x}) \mathbf{p}^T(\mathbf{x}) \cdot [\mathbf{A}(\mathbf{x})]^{-1} \cdot \mathbf{p}(\mathbf{x}_1). \tag{32}$$

Appendix B: Moving Least Squares Interpolation

Figure 14 gives a geometrical idea of the interpolation performed with MLS functions in one dimension. When a position \mathbf{x} is within the support of two nodes, the implicit

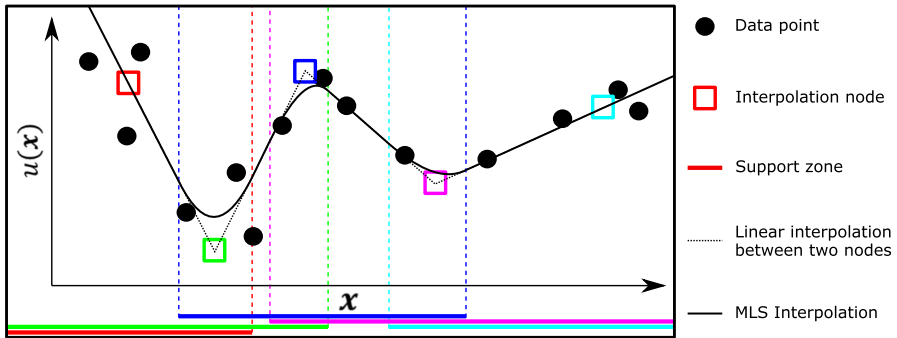


Fig. 14 Sketch of the MLS interpolation in one dimension with a polynomial order of 1 and supports only overlapping the adjacent neighboring nodes

function follows the linear interpolation between these two nodes and their coefficients. If it is within the support of three nodes, the implicit function follows a more complex trend smoothly evolving between the linear interpolations of the two pairs separately. Such interpolation can be compared to NURBS and other B-splines which use control nodes to create smooth lines and surfaces in a similar manner (Piegl and Tiller 1997).

In Fig. 14, the interpolation is strictly linear in some parts. This is due to the first order polynomial (i.e., $m = 2$ in one dimension) and the local number of neighboring nodes n (i.e., 2). Here, the derivability of the MLS functions is not controlled by the weight functions as stated in Sect. 2.1.2. It is a specific case: if the number of monomials m is equal to the number of neighbors n , System (26) is square and the weight functions have no influence when determining the coefficients. If the number of neighbors is larger than the number of monomials, the statement of Sect. 2.1.2 is true.

Appendix C: Inducing a Dependency Between Fault Blocks

Figure 15a illustrates the intrinsic assumption made when using the visibility criterion: separate fault blocks are independent from one another. In Fig. 15b, the dependency of the structures between the two fault blocks is controlled with a transparency criterion: the slope imposed by the gradient data also affects the other side of the fault. The criterion is written as a reduction of the influence of any neighboring node \mathbf{x}_1 on the other side of the fault compared to a point \mathbf{x} as

$$w_1(\mathbf{x}) = \tau w \left(\frac{|x - x_1|}{\rho_{1x}} \right) w \left(\frac{|y - y_1|}{\rho_{1y}} \right), \quad (33)$$

with $\tau \in [0, 1]$ the input transparency coefficient. This reduction is written on the weight functions in order to preserve the partition of unity formed by the MLS functions. Equation (33) is an adaptation of the classical transparency criterion (Organ et al. 1996) for a homogeneous transparency over all the discontinuity and not only

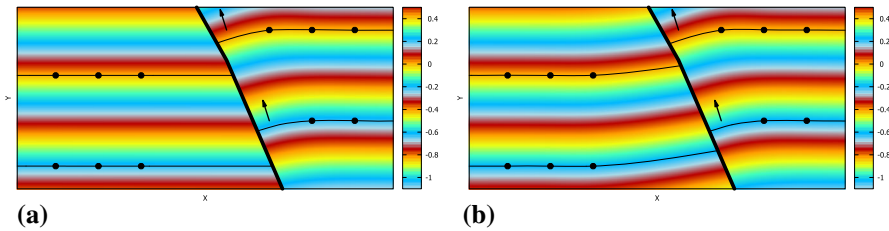


Fig. 15 Difference between two optic criteria applied homogeneously on an infinite fault. **a** The visibility criterion imposes the two fault blocks to be completely independent from one another, **b** the transparency criterion can ensure a discontinuous jump between the two fault blocks while preserving a mutual dependency on the structures

in the vicinity of its tip. The coefficient τ must be empirically tuned and the relation between its value and the results is not intuitive: a τ equal to 0.001 was used in the example. Further studies could help normalizing τ depending on the application.

Appendix D: Data Equations

Orientation Data

Dip data can be converted into normal vectors $\mathbf{g}(g_x, g_y)$, which are perpendicular to the stratigraphy, and tangential vectors $\mathbf{t}(t_x, t_y)$, which are parallel to the stratigraphy. Two choices of constraints are then possible with such vectorial information (Caumon et al. 2013; Chilès et al. 2004; Hillier et al. 2014): gradient data and orientation data.

The gradient data constraint imposes both the direction and the norm of the gradient of the solution as

$$\nabla u(\mathbf{x}_g) = \mathbf{g}_i, \tag{34}$$

with ∇ the gradient operator and \mathbf{x}_g the gradient measurement location. Equation (34) represents two equations in two dimensions. In general, gradient data are normalized as unit vectors and used with increment data points (Chilès et al. 2004) (presented in the following). It is possible to incorporate thickness variations with varying norms in the gradient data.

The orientation data constraint imposes the gradient of the solution to be perpendicular to the measured dip

$$\nabla u(\mathbf{x}_t) \cdot \mathbf{t} = \frac{\partial}{\partial x} u(\mathbf{x}_t)t_y + \frac{\partial}{\partial y} u(\mathbf{x}_t)t_x = 0, \tag{35}$$

with ∇ the gradient operator and \mathbf{x}_t the orientation measurement location. In three dimensions, two different vectors \mathbf{t} within the stratigraphic plane are necessary for this constraint. Equation (35) is unrelated to the norm and polarity of the input vector, only to its orientation.

As a sum of MLS interpolation functions (Eq. 1), the implicit function is derivable enough to compute first order derivatives everywhere in the domain Ω (Sect. 2.1.2). In addition, System (13) can include several equations involving a same subset of unknowns in a least squares sense. The presented method is thus well suited to both constraints. As an example, gradient data constraints are used in Fig. 6 with gradients extracted from the reference model. Their influence on the results is particularly visible in the fault block with one hard datum and three gradient data points.

Data Points Values

Data values as inputs When the expected implicit value α_h of a horizon h in the final model is known, a data equation represents a line in System (13) as

$$u(\mathbf{x}_{\text{data}}) = \alpha_h, \quad \forall \mathbf{x}_{\text{data}} \in h. \quad (36)$$

This is the most common way of imposing data terms (Frank et al. 2007; Mallet 2002) and it is the technique used in the presented method (Sect. 2.2).

A difficulty is that the geologist has a priori no possibility to know the expected α_h values. However, these input values have a strong impact on the solution quality. Input values should be chosen: (i) monotonically following the relative geological times, and (ii) adapted to the average thicknesses of the layers. The closer two horizons are one to another, the closer their associated input values should be. Even though some preprocessing on the data points exist to determine such values, it remains a difficult task, especially in real case studies (Collon et al. 2015).

Data values as increments When no a priori knowledge of the expected implicit values in the final model is available, it is possible to use increments (Chilès et al. 2004). As two data points $\mathbf{x}_{\text{data}_1}$ and $\mathbf{x}_{\text{data}_2}$ belonging to the same horizon h are supposed to have the same unknown iso-value, the increment constraint forces the implicit function u to have the same value at those two positions as

$$u(\mathbf{x}_{\text{data}_1}) - u(\mathbf{x}_{\text{data}_2}) = 0, \quad \forall (\mathbf{x}_{\text{data}_1}, \mathbf{x}_{\text{data}_2}) \in h, \quad \mathbf{x}_{\text{data}_1} \neq \mathbf{x}_{\text{data}_2}. \quad (37)$$

In practice, a model cannot be described only with increment constraints: it needs a reference value and a reference trend. In most applications, the reference value [e.g., 0] is given to an arbitrary point in the model, and the reference trend is given by normalized gradient data constraints (i.e., with a norm of 1) (Chilès et al. 2004).

Gradient data are not supposedly necessary to the proposed method. Therefore, in the input data points of Fig. 1a, extremum horizons per stratigraphic sequences were used as a reference (i.e., reference values of 0, 1, 2 and 2.5 in Table 1) and increment constraints were used for the other horizons. This created the reference model Fig. 1b and the remaining default iso-values of Table 1 were deduced from it. These deduced values were used with Eq. (36) for all the other tests and models.

References

- ARANZ Geo (2018) Leapfrog geo. <http://www.leapfrog3d.com/products/leapfrog-geo>. Accessed 24 Oct 2018
- Aug C (2004) Modélisation géologique 3D et caractérisation des incertitudes par la méthode du champ de potentiel. PhD thesis
- Belytschko T, Lu YY, Gu L (1994) Element-free Galerkin methods. *Int J Numer Methods Eng* 37(2):229–256
- Belytschko T, Krongauz Y, Fleming M, Organ D, Liu WKS (1996) Smoothing and accelerated computations in the element free Galerkin method. *J Comput Appl Math* 74(1):111–126
- Brezzi F (1974) On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. *Rev Française d'autom Inf Rech Opér Anal Numér* 8(R2):129–151
- Calcagno P, Chilès JP, Courrioux G, Guillen A (2008) Geological modelling from field data and geological knowledge. Part I. Modelling method coupling 3D potential-field interpolation and geological rules. *Phys Earth Planet Inter* 171(1–4):147–157
- Carmichael T, Ailleres L (2016) Method and analysis for the upscaling of structural data. *J Struct Geol* 83:121–133
- Caumon G (2009) Vers une intégration des incertitudes et des processus en géologie numérique. PhD thesis
- Caumon G, Collon-Drouaillet P, De Veslud CLC, Viseur S, Sausse J (2009) Surface-based 3D modeling of geological structures. *Math Geosci* 41(8):927–945
- Caumon G, Gray G, Antoine C, Titeux MO (2013) Three-dimensional implicit stratigraphic model building from remote sensing data on tetrahedral meshes: theory and application to a regional model of la popa basin, ne mexico. *IEEE Trans Geosci Remote Sens* 51(3):1613–1621
- Chilès JP, Aug C, Guillen A, Lees T (2004) Modelling the geometry of geological units and its uncertainty in 3D from structural data: the potential-field method. In: *orebody modeling and strategic mine planning—spectrum* 14(July):22–24
- Collon P, Steckiewicz-Laurent W, Pellerin J, Laurent G, Caumon G, Reichart G, Vaute L (2015) 3D geomodelling combining implicit surfaces and Voronoi-based remeshing: a case study in the Lorraine coal basin (France). *Comput Geosci* 77:29–43
- Courant R (1950) Dirichlet's principle, conformal mapping, and minimal surfaces. Interscience, New York
- Cowan E, Beatson R, Ross H, Fright W, McLennan T, Evans T, Carr J, Lane R, Bright D, Gillman A, Oshust P, Titley M (2003) Practical implicit geological modelling. In: *5th international mining geology conference*, vol 8, pp 89–99
- Cuomo S, Galletti a, Giunta G, Starace A (2013) Surface reconstruction from scattered point via RBF interpolation on GPU. In: *2013 federated conference on computer science and information systems (FedCSIS)*, pp 433–440. [arXiv:1305.5179v1](https://arxiv.org/abs/1305.5179v1)
- De la Varga M, Schaaf A, Wellmann F (2019) Gempy 1.0: open-source stochastic geological modeling and inversion. *Geosci Model Dev* (**in review**)
- Dubrule O (1984) Comparing splines and kriging. *Comput Geosci* 10(2–3):327–338
- Duchon J (1977) Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In: *Constructive theory of functions of several variables*, Springer, pp 85–100
- Fossen H (2016) *Structural geology*. Cambridge University Press, Cambridge
- Frank T, Tertois AL, Mallet JL (2007) 3d-reconstruction of complex geological interfaces from irregularly distributed and noisy point data. *Comput Geosci* 33(7):932–943
- Fries TP, Matthias HG (2004) Classification and overview of meshfree methods. Department of Mathematics and Computer Science. Technical University of Braunschweig, p 64
- Godefroy G, Caumon G, Ford M, Laurent G, Jackson CAL (2018) A parametric fault displacement model to introduce kinematic control into modeling faults from sparse data. *Interpretation* 6(2):B1–B13
- Gonçalves ÍG, Kumaira S, Guadagnin F (2017) A machine learning approach to the potential-field method for implicit modeling of geological structures. *Comput Geosci* 103:173–182
- Grose L, Laurent G, Ailleres L, Armit R, Jessell M, Caumon G (2017) Structural data constraints for implicit modeling of folds. *J Struct Geol* 104:80–92
- Hillier MJ, Schetselaar EM, de Kemp EA, Perron G (2014) Three-dimensional modelling of geological surfaces using generalized interpolation with radial basis functions. *Math Geosci* 46(8):931–953
- Houlding SW (1994) *3D geoscience modelling*. Computer techniques for geological characterisation, Springer, New York

- Intrepid-Geophysics (2018) Geomodeller. <http://www.intrepid-geophysics.com/ig/index.php?page=geomodeller>. Accessed 24 Oct 2018
- Jessell M, Aillères L, Kemp ED, Lindsay M, Wellmann F, Hillier M, Laurent G, Carmichael T, Martin R (2014) Next generation three-dimensional geologic modeling and inversion. *SEG Spec Publ* 18:261–272
- Karimi-Fard M, Durlafsky LJ (2016) A general gridding, discretization, and coarsening methodology for modeling flow in porous formations with discrete geological features. *Adv Water Resour* 96:354–372
- Lajaunie C, Courrioux G, Manuel L (1997) Foliation fields and 3D cartography in geology: principles of a method based on potential interpolation. *Math Geol* 29(4):571–584
- Lancaster P, Salkauskas K (1981) Surfaces generated by moving least squares methods. *Math Comput* 37(155):141–158
- Laurent G (2016) Iterative thickness regularization of stratigraphic layers in discrete implicit modeling. *Math Geosci* 48(7):811–833
- Laurent G, Aillères L, Grose L, Caumon G, Jessell M, Armit R (2016) Implicit modeling of folds and overprinting deformation. *Earth Planet Sci Lett* 456:26–38
- Liu GR, Gu Y (2005) An introduction to meshfree method and their programming. Springer, New York
- Mallet JL (1988) Three-dimensional graphic display of disconnected bodies. *Math Geol* 20(8):977–990
- Mallet JL (1992) Discrete smooth interpolation. *Comput Aided Des* 24(4):178–191
- Mallet JL (1997) Discrete modeling for natural objects. *Math Geol* 29(2):199–219
- Mallet JL (2002) Geomodeling. Oxford University Press Inc, Oxford
- Mallet JL (2014) Elements of mathematical sedimentary geology: The GeoChron model. EAGE publications, Houten
- Marechal A (1984) Kriging seismic data in presence of faults. In: Verly G, David M, Journel AG, Marechal A (eds) Geostatistics for natural resources characterization. Springer, Dordrecht, pp 271–294
- Martin R, Boisvert JB (2017) Iterative refinement of implicit boundary models for improved geological feature reproduction. *Comput Geosci* 109:1–15
- Matheron G (1981) Splines and kriging: their formal equivalence. In: Down-to-earth statistics: solutions looking for geological problems, vol 8. Syracuse University Geological Contributions, pp 77–95
- McLain DH (1976) Two dimensional interpolation from random data. *Comput J* 19(2):178–181
- Nguyen VP, Rabczuk T, Bordas S, Duflo M (2008) Meshless methods: a review and computer implementation aspects. *Math Comput Simul* 79(3):763–813
- Organ D, Fleming M, Terry T, Belytschko T (1996) Continuous meshless approximations for nonconvex bodies by diffraction and transparency. *Comput Mech* 18(3):225–235
- Paradigm (2018) Skua-gocad. <http://www.pdgm.com/products/skua-gocad/>. Accessed 24 Oct 2018
- Pellerin J, Lévy B, Caumon G, Botella A (2014) Automatic surface remeshing of 3D structural models at specified resolution: a method based on Voronoi diagrams. *Comput Geosci* 62:103–116
- Piegl L, Tiller W (1997) The NURBS book. Monographs in visual communication. Springer, New York
- Schlumberger (2018) Petrel. <https://www.software.slb.com/products/petrel>. Accessed 24 Oct 2018
- Shewchuk J (2002) What is a good linear finite element? Interpolation, conditioning, anisotropy, and quality measures (preprint). University of California at Berkeley, vol 73, p 137
- Souche L, Iskenova G, Lepage F, Desmarest D, et al (2014) Construction of structurally and stratigraphically consistent structural models using the volume-based modelling technology: applications to an Australian dataset. In: International petroleum technology conference
- Wahba G (1990) Spline models for observational data, vol 59. Society for Industrial and Applied Mathematics, Philadelphia
- Wendland H (1995) Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv Comput Math* 4(1):389–396
- Wu Z (1995) Compactly supported positive definite radial functions. *Adv Comput Math* 4(1):283
- Yokota R, Barba LA, Knepley MG (2010) Petrbbf—a parallel $O(n)$ algorithm for radial basis function interpolation with Gaussians. *Comput Methods Appl Mech Eng* 199(25):1793–1804