Formulation continue du problème de modélisation implicite de structures géologiques discrétisée avec des méthodes de réduction de maillage

Julien Renaudeau

Continuous formulation of implicit structural modeling discretized with mesh reduction methods

THÈSE

Présentée et soutenue publiquement le 24 avril 2019
pour l'obtention du grade de

Docteur de l'Université de Lorraine

Spécialité Géosciences

par

Julien Renaudeau

Composition du jury:

Présidente du jury:	Dominique Bechmann
Rapporteurs:	Julie Digne Florian Wellmann
Examinatrices:	Dominique Bechmann Laetitia Le pourrhiet Sophie Viseur
Directeur de thèse:	Guillaume Caumon
Co-directeurs de thèse:	Frantz Maerten Bruno Lévy

UMR 7359 - GeoRessources
Université de Lorraine - CNRS - CREGU
ENSG - Campus Brabois - BP 10162
54505 Vandœuvre-lès-Nancy cedex - FRANCE

Contents

Introduction **15**

1 State of the art of implicit structural modeling **23**
1.1 Structural modeling . . . 23
1.1.1 Stratigraphic structures . . . 23
1.1.2 Motivation to model the stratigraphic structures . . . 25
1.1.3 Modeling geological surfaces . . . 25
1.1.4 Numerical data and origins . . . 26
1.1.4.1 Seismic data . . . 26
1.1.4.2 Wellbore data . . . 26
1.1.4.3 Outcrop observations . . . 26
1.1.4.4 Contextual knowledge . . . 26
1.1.4.5 Summary: The nature of the numerical data . . . 26
1.1.5 Expected models . . . 27
1.1.6 Two modeling approaches . . . 28
1.2 Implicit structural modeling . . . 29
1.2.1 Definition of implicit modeling . . . 29
1.2.1.1 Scalar field and iso-surfaces . . . 29
1.2.1.2 Sets of sub-parallel iso-surfaces . . . 29
1.2.1.3 Representing a set of points . . . 30
1.2.1.4 Extraction of iso-surfaces . . . 31
1.2.2 Application to structural modeling . . . 31
1.2.2.1 Stratigraphic function . . . 31
1.2.2.2 Stratigraphic unconformities as contact surfaces between functions . . . 32
1.2.2.3 Faults as stratigraphic jumps . . . 32
1.2.2.4 Intrusive geobodies as stratigraphic cuts . . . 32
1.2.2.5 Discussion on structural discontinuities . . . 32
1.2.3 Level-set methods in structural modeling . . . 33
1.3 Numerical methods for implicit structural modeling . . . 34
1.3.1 Discrete Smooth Interpolation (DSI) . . . 34
1.3.1.1 Origin and applications . . . 34
1.3.1.2 Constraints on numerical data . . . 35
1.3.1.3 Mesh-based interpolation . . . 35
1.3.1.4 Smooth regularization . . . 36
1.3.1.5 Handling heterogeneous data . . . 37
1.3.1.6 Handling structural discontinuities . . . 38
1.3.1.7 Example of DSI system of equations . . . 38
1.3.1.8 Visualizing the results . . . 38
1.3.1.9 Advantages and limits . . . 39
1.3.2 Potential Field Method (PFM) . . . 40
1.3.2.1 Origin and applications . . . 40
1.3.2.2 Numerical constraints . . . 41
1.3.2.3 Geostatistical formulation . . . 42
1.3.2.4 Global meshless formulation . . . 42
1.3.2.5 Global smooth interpolation . . . 43

1.3.2.6 Handling heterogeneous data . . . 44
1.3.2.7 Handling structural discontinuities . . . 46
1.3.2.8 Example of PFM system of equations . . . 49
1.3.2.9 Visualizing the results . . . 49
1.3.2.10 Advantages and limits . . . 49
1.3.3 Common modeling limits . . . 50
1.4 Improving the implicit structural methods . . . 50
1.4.1 Improving the data . . . 50
1.4.2 Improving the modeling tool . . . 51
1.4.2.1 Improving the algorithm . . . 51
1.4.2.2 Optimizing the algorithm . . . 52
1.4.3 Aims of the thesis . . . 53
1.5 Unique surface construction: A modeling problem analogous to structural modeling . 54
1.5.1 Modeling problem . . . 54
1.5.2 Analogy with structural modeling . . . 54
1.5.3 Some implicit modeling techniques . . . 55
1.5.3.1 Solving the Poisson equation . . . 55
1.5.3.2 Radial Basis Functions (RBF) surfaces . . . 55
1.5.3.3 Moving Least Squares (MLS) surfaces . . . 56
1.5.3.4 Partition of Unity Method (PUM) surfaces . . . 56

2 Continuous framework of implicit structural modeling 59
2.1 Review of numerical methods for continuous modeling problems . . . 59
2.1.1 Partial Differential Equation under constraints . . . 59
2.1.1.1 Weak formulation . . . 59
2.1.1.2 Discretization . . . 60
2.1.1.3 System of equations and solution . . . 61
2.1.1.4 Finite Element Method and meshless methods . . . 61
2.1.2 Test functions . . . 61
2.1.2.1 Collocation method . . . 61
2.1.2.2 Bubnov-Galerkin . . . 62
2.1.2.3 Petrov-Galerkin . . . 62
2.1.3 Shape functions . . . 64
2.1.3.1 A few concepts . . . 64
2.1.3.2 Using mesh elements . . . 65
2.1.3.3 Using radial functions . . . 65
2.1.3.4 Building an intrinsic Partition of Unity (PU) . . . 67
2.1.3.5 Building an extrinsic Partition of Unity (PU) . . . 69
2.1.3.6 Hermite-type shape functions . . . 70
2.1.3.7 Other shape functions . . . 71
2.1.4 Handling discontinuities . . . 72
2.1.4.1 Optic criteria on compact supports . . . 72
2.1.4.2 By enrichment of the basis functions . . . 72
2.1.5 Examples of numerical methods and applications . . . 73
2.1.6 Discussion on mesh-based and meshless methods . . . 77
2.1.7 Application to implicit structural modeling . . . 78
2.1.7.1 Positioning DSI and PFM . . . 78
2.1.7.2 Opportunities . . . 79
2.2 Application to implicit structural modeling . . . 79
2.2.1 Lack of a strong formulation in implicit structural modeling . . . 79
2.2.2 The proposed generic framework . . . 80
2.2.3 The standard framework . . . 80
2.2.3.1 A spatial regression of data points . . . 80
2.2.3.2 The bending energy . . . 81
2.2.3.3 A spatial regression of data points penalized by the bending energy . 81
2.2.4 Formulation with generic bases functions and domain discretization . . . 82

2.2.4.1 Test and shape functions . . . 82
2.2.4.2 Domain discretization . . . 82
2.2.4.3 General system . . . 82
2.2.4.4 Least squares equivalence in a Bubnov-Galerkin scheme . . . 83
2.2.5 Other data constraints . . . 83
2.2.5.1 Increment data points (IDP) . . . 83
2.2.5.2 Normal gradient data (NGD) . . . 83
2.2.5.3 Tangent data (TD) . . . 84
2.2.5.4 Polygonal lines (PL) and surfaces (PS) . . . 84
2.2.5.5 Hard data points (HDP) . . . 85
2.2.5.6 Inequality data constraints (IqD) . . . 86
2.2.5.7 Concept of minimum requirement of a data setting . . . 87
2.2.6 Unconformities and faults treated as discontinuities . . . 87
2.2.7 Relation with existing methods . . . 87
2.2.7.1 Discrete Smooth Interpolation . . . 87
2.2.7.2 Potential Field Method . . . 88

3 Moving Least Squares discretization 91
3.1 Motivations . . . 91
3.2 The proposed method . . . 91
3.2.1 The modeling problem using Moving Least Squares functions . . . 91
3.2.2 Node discretization . . . 92
3.2.3 Moving Least Squares (MLS) shape functions . . . 92
3.2.3.1 Default parameters . . . 93
3.2.3.2 Plot of the MLS functions and second order derivatives . . . 94
3.2.4 Test functions and domain discretization . . . 94
3.2.5 Handling the discontinuities . . . 95
3.2.6 Example of a system of equations . . . 95
3.2.7 Extraction of the iso-surfaces . . . 95
3.3 Some results . . . 96
3.3.1 Standard setting . . . 97
3.3.2 Other data constraints . . . 98
3.3.3 Other optic criteria . . . 101
3.4 Sensitivity tests . . . 103
3.4.1 Reference model for the sensitivity tests . . . 103
3.4.2 Model distance and data distance . . . 103
3.4.3 Sensitivity to data quality . . . 105
3.4.3.1 Data sparsity . . . 105
3.4.3.2 Noisy data . . . 105
3.4.4 Sensitivity to the method's parameters . . . 106
3.4.4.1 Normalizing the smoothing equations . . . 106
3.4.4.2 Complexity and stability of the moving least squares functions . . . 109
3.4.4.3 Regular or irregular sampling . . . 112
3.4.4.4 Complex geometries of structural discontinuities . . . 113
3.5 Perspectives and optimizations . . . 114
3.5.1 Finite difference method . . . 114
3.5.2 Other possible optimizations . . . 115
3.5.3 Adaptive sampling resolution with a varying dilatation parameter . . . 115

4 Cartesian grid discretization with ghost nodes 119
4.1 Motivations . . . 119
4.2 The proposed method . . . 120
4.2.1 The modeling problem using a Cartesian grid with ghost nodes . . . 120
4.2.2 Bilinear quadilateral shape functions . . . 120
4.2.3 Test functions and domain discretization . . . 121
4.2.4 Finite difference approximation of the continuous energy . . . 122

4.2.5 Handling the discontinuities with ghost nodes . . . 123
4.2.5.1 Definition . . . 123
4.2.5.2 Structural discontinuities with complex geometries . . . 124
4.2.5.3 Finite difference method with ghost nodes . . . 125
4.2.6 Example of a system of equations . . . 126
4.2.7 Extraction of the iso-surfaces . . . 127
4.3 Comparison with the moving least squares (MLS) discretization . . . 127
4.3.1 Comparison tests in 2D . . . 127
4.3.1.1 Visual comparisons of the interpolation . . . 127
4.3.1.2 Distance comparisons of the interpolation . . . 127
4.3.2 Benefits and limits of the Cartesian grid with ghost nodes . . . 129
4.3.2.1 Computational efficiency . . . 129
4.3.2.2 Limits of the ghost nodes . . . 130
4.3.2.3 Lack of flexibility at a given resolution . . . 131
4.4 Perspectives and optimizations . . . 131
4.4.1 Implementation in 3D . . . 131
4.4.2 Tree structures . . . 132

5 Improving the continuous problem 135
5.1 Motivation: the limits of the bending energy in implicit structural modeling . . . 135
5.2 Replacing the bending energy by other well known smoothing energies . . . 136
5.3 Modifying the bending energy . . . 137
5.3.1 Physical meaning of the bending energy . . . 137
5.3.2 Thickness variation context . . . 138
5.3.3 Correction with the Weighted Curvature Minimization criterion . . . 138
5.3.3.1 Concept . . . 138
5.3.3.2 Discretization . . . 140
5.3.3.3 Applications . . . 140
5.3.4 Limits of the Weighted Curvature Minimization criterion . . . 141
5.3.5 Relation with Discrete Smooth Interpolation on meshes with adaptive resolution 143
5.4 Enriching the continuous problem with an anisotropic direction . . . 143
5.4.1 Concerned issues and existing solutions . . . 143
5.4.2 Imposing a direction of anisotropy on the minimization problem . . . 144
5.5 Replacing the bending energy by an anisotropic energy . . . 146
5.5.1 Equivalence between isotropic smoothing and bending energy . . . 146
5.5.2 Anisotropy as an ellipse . . . 146
5.5.3 Anisotropy replacing the weighted curvature minimization . . . 147

6 Other modeling applications of the proposed workflow 151
6.1 Structural modeling of borehole image interpretations . . . 151
6.1.1 Specificities of the application . . . 151
6.1.2 Implicit modeling plugin . . . 152
6.2 Salt envelope surface modeling . . . 154
6.2.1 Specificities of the application . . . 154
6.2.2 Synthetic models . . . 154
6.2.3 Application cases . . . 156
6.2.3.1 Surface reconstruction in the Santos Basin . . . 156
6.2.3.2 Analogue model of the Nordkapp Basin . . . 157
6.3 Mechanical structural restoration . . . 158
6.3.1 Overview of the application . . . 158
6.3.1.1 Strong form . . . 158
6.3.1.2 Weak form . . . 159
6.3.2 Developed prototype . . . 159

General conclusions 161

Appendices **165**
A Moving least squares demonstration 165
B Discussion on the biharmonic equation and the bending energy in structural modeling 166
C Euler-Lagrange equations on the minimization problem 167
D Approximation with the bases functions: equations details 168
E Least squares equivalence in a Bubnov-Galerkin scheme: equations details 169
F Normalizing the influence of the number of data constraints in a least squares system . 170
G Lagrange multipliers: equations details 171
H Plots of weight and MLS functions 172
I Integration of an homogeneous directional rigidity 176
J Integration of an anisotropic directional rigidity 177

Abbreviations

1D	One-Dimensional space	MLS	Moving Least Squares
2D	Two-Dimensional space	PIM	Polynomial Interpolation Method
3D	Three-Dimensional space	PUM	Partition of Unity Method
		RBF	Radial Basis Functions
BC	Boundary Conditions	RPIM	Radial Polynomial Interpolation Method
BG	Bubnov-Galerkin	TPS	Thin Plate Splines
EBC	Essential Boundary Conditions	WLS	Weighted Least Squares
NBC	Neumann Boundary Conditions		
PDE	Partial Differential Equation	KΔD	Kronecker Delta Property
PG	Petrov-Galerkin	PU	Partition of Unity
DSI	Discrete Smooth Interpolation	DPV	Data Points Value
PFM	Potential Field Method	HDP	Hard Data Points
VBM	Volume Based Modeling	IDP	Increment Data Points
		IqD	Inequality Data
FEM	Finite Element Method	NGD	Normal Gradient Data
MM	Meshless Methods	PL/PS	Polygonal Lines and Surfaces
		TD	Tangent Data

Nomenclature

Police and symbol	**Meaning**
x	A constant
$\boldsymbol{x}$	A vertical vector: $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$
$\boldsymbol{x}^T$	A horizontal vector: $\begin{bmatrix} x & y & z \end{bmatrix}$
$[\ ]$	Delimits a vector or a matrix
$\{\ \}$	Delimits a set or a space
$\left\{ \right.$	Delimits a system of equations
$\forall x \in set, \quad equation$	*equation* is written for an arbitrary x taken in *set*
$\left\{ \begin{array}{l} equation, \quad \forall x \in set \end{array} \right.$	The system is created by writting *equation* on each x in *set* vertically
$\boldsymbol{x} = [x \ \mid \ \forall x \in set]$	The vector $\boldsymbol{x}$ is created by writting each x in *set* vetically

Table of symbols

Symbol	Meaning
$\boldsymbol{a}$	Vector of polynomial coefficients
$\boldsymbol{A}$	Moment matrix
$\boldsymbol{D}$	Global set of data
$\boldsymbol{D}(H)$	Set of data points in the horizon H
e	Mesh element
e'	Adjacent mesh element to e
F	Discontinuity or fault
$\boldsymbol{g}$	Gradient data
H	Horizon
$\boldsymbol{H}$	Hessian operator
$_{i,j}$	Indices on the coordinates of space
L	Linear differential operator
m	Monomial
$\boldsymbol{N}$	Set of interpolation nodes
$\boldsymbol{p}$	Point
$\overline{\boldsymbol{p}}$	Node
$\mathcal{P}$	Polynomial basis
r	Radial distance
S	Local support of a function
$\boldsymbol{S}$	Set of local supports
s^2	Nugget value
t	Time
^{T}	Transpose operator of a vector or a matrix
u	Implicit function
$\boldsymbol{U}$	Vector of unknown coefficients
v	Test function
$\boldsymbol{V}$	Vector of the test function's coefficients
w	Weight function
$\boldsymbol{x}$	Vector of spatial coordinates: $[x]^T$ in 1D, $[x,y]^T$ in 2D, $[x,y,z]^T$ in 3D
α	Iso-value
δ	Dirac function
Δ	Laplacian operator
κ	Jump function
λ	Penalization weight
∇	Gradient operator
Ω	Domain of study
ω	Subdomain
∂_i	First derivative in the i axis ($\frac{\partial}{\partial i}$)
∂^2_{ij}	Second derivative in the i and j axes ($\frac{\partial^2}{\partial ij}$)
Φ	Shape (trial, interpolation) function
$\boldsymbol{\Phi}$	Basis shape functions
Ψ	Test function (basis function)
$\boldsymbol{\Psi}$	Basis test functions
ρ	Length of a support
$\#$	Number of elements in a set

Remerciements

Voici venu le moment des remerciements. Sur le papier, ça vient tout de suite, c'est facile, mais l'attente a été un peu plus longue en vécut. Il faut dire que j'ai beaucoup repoussé ce moment car c'est sans doute le plus important d'une thèse : il en marque la fin, et j'aimerais que cette fin soit aussi belle que ces trois dernières années. Je vais donc mentionner, avec peu de mots, toutes ces relations indescriptibles qui m'ont apporté le soutien nécessaire à la réalisation de ces travaux.

Pour commencer, formalité oblige, j'aimerais remercier les institutions qui ont amené les pépettes. Pas d'argent, pas d'thèse, et pas d'thèse, pas de trois belles années. J'aimerais surtout remercier l'entreprise Schlumberger, en particulier le "Montpellier Technology Center", qui a financé la moitié de cette thèse. Malgré leur faible nombre de doctorants CIFRE, Schlumberger m'a fait confiance pour ce sujet porteur, et je suis bien content de voir que cette confiance perdure aujourd'hui, en Norvège. Je remercie aussi l'Association Nationale de la Recherche et de la Technologie pour l'autre moitié du financement. Ce contrat CIFRE est assez incroyable : il permet d'être en première loge pour découvrir le monde industriel tout en ayant un pied dans un laboratoire de recherche académique ; une situation de recherche et développement rêvée. Justement, merci à l'Université de Lorraine, le laboratoire GeoRessources, et l'équipe RING qui m'ont accueilli pour la partie académique. Le nombre de séminaires et conférences pour la recherche qui y sont organisés m'ont souvent fait regretter de ne pas y passer plus de temps. Bien qu'elles n'aient pas eu un impact direct sur ces travaux, je tiens à remercier les écoles qui m'ont préparé pour cette épreuve tout en me faisant grandir : La maternelle Elsa Triolet, l'école primaire Joliot Curie, le collège Éric Satie, le lycée Honoré de Balzac, le lycée Fénelon, et l'École Nationale Supérieure de Géologie.

Je remercie grandement les membres de mon jury. Merci aux rapporteurs d'avoir accepté de relire l'ensemble de mes travaux : Julie Digne qui a supporté patiemment cette malencontreuse situation téléphonique, et Florian Wellmann dont l'intérêt pour le sujet a mené à poser un (très) grand nombre de questions passionnantes. Merci à Dominique Bechmann pour avoir présidé avec efficacité et bienveillance cette soutenance. Et merci aux deux autres examinatrices : Laetitia Le Pourhiet pour ses questions pointues, et Sophie Viseur pour son petit cadeau clin d'œil et ses questions recherchées. L'ensemble de cette soutenance a été un moment inoubliable, et c'est en grande partie grâce à vous.

Un immense merci à tous mes encadrants. Tout d'abord à Frantz qui a eu l'idée de ce sujet, s'est battu pour en débloquer le budget, m'a apporté tous les outils de développement pour travailler en bonnes conditions, et a été une réelle source de motivation quotidienne. Les apports de Frantz à cette thèse sont innombrables et je réalise petit à petit combien il m'a fait gagner du temps (#Compilation, #Visualisation). Merci à Guillaume qui a accepté d'encadrer cette thèse malgré la distance et qui a pris le temps de lire plusieurs fois, et dans l'extrême détail, tous mes papiers. La qualité du travail écrit et oral de cette thèse ne serait rien sans ses bons conseils. Merci également à Bruno, qui a initié ce qui est devenu le thème central de cette thèse : les équations continues. Et enfin, je n'oublierais pas le plus époustouflant, ce véritable héros des temps modernes, Manou ! (alias le Geeeennniouuuusss !!). Emmanuel a su me faire aimer ces drôles de choses que sont les méthodes numériques grâce à sa patience, sa pédagogie, et surtout son humour. Je ne pourrais oublier toutes ces heures passées à noircir notre tableau d'équations et de points (en meshless, on dessine des points partout !).

J'ai passé beaucoup de temps en entreprise dans un centre technologique de plus de cent personnes (bon, ça a baissé. . . , mais au début c'était le cas !), et beaucoup m'ont apporté un grand soutient, même sans forcément s'en rendre compte. J'ai particulièrement envie de remercier l'équipe Petrel/VBM, à laquelle j'étais rattaché. Merci aux modeleurs avec qui j'ai eu le plus d'interactions : Azeddine, et ses discussions politiques ; Thomas V, et sa gentillesse/son humour à toute épreuve, et Thomas L, qui a déjà tout fait. Merci à mes chefs qui ont approuvé tous mes eClaims sans sourciller : Terje, weeell

you knoow ; et Giselle. Merci aux portfolios qui ont tenté de m'ouvrir les yeux sur le fonctionnement du reste de l'entreprise et pas que le côté dév : Arnaud, et son positivisme corporate légendaire ; Jean-Pierre ("gamin !") ; et Naouelle, notre respo cadeaux. Merci à Laurent pour tous les modèles géol et son calme en toute situation, et merci également au reste de l'équipe, présente comme passée... Un petit coucou aux stagiaires aussi qui s'y sont succédés : Lucille (ça marchera jamais), Hademimi (Julieeeeen), Agathe, Carole, Anne, etc... En dehors de cette équipe, je ne peux passer à côté de Philippe, qui m'a toujours beaucoup soutenu, qui a su me conseiller sur un nombre incroyable de sujets, professionnels comme personnels, et qui m'a fait confiance pour bidouiller son plugin de cœur avec Zhenhua. Bon, par contre, la chemise tous les jours, c'est pas encore ça... Je remercie aussi les deux derniers patrons de MpTC : Haitao et Stéphane; les RH qui ont supporté mon grand nombre de questions ; Jeff notre G.I Joe local ; le reste des membres du CE qui répondent toujours présents ; Damien le super IT; Vianney et Sylvains, nos copains chercheurs ; et tous les autres, que je n'arrive pas à classer.

Je suis aussi très heureux d'avoir pu partager mon temps avec l'équipe RING. Déjà parce qu'elle forme partie de mes origines, mais aussi pour son cadre chaleureux et favorable. Je remercie Pauline pour tous ses bons conseils ; Paul pour les parties de tarot et pour avoir donné un nouveau souffle à l'expression de mon père "Ça là ? Ça ?!! Ça c'est bon Ça !" ; Christine pour sa constante bonne humeur ; Christophe ; et Gautier pour les bonnes discussions au tableau. Un énorme merci à Fifi, qui m'a fait découvrir ce que c'est qu'un vrai rhum et qui m'a toujours ouvert sa porte (en échange de... travaux chez lui > _ >). Du côté thésard, ça me fait un peu bizarre de rejoindre le groupe des anciens : Guillaume, Jérémy, Arnaud, Benjamin, Charline, Jon, Antoine, Marion et Gaby. Remerciements particuliers à Arnaud, sans qui je n'aurais même pas envisagé la thèse ; Jon avec qui j'ai vécu des choses bien marrantes (ping-pong interminable à la guest, festival parc d'attractions, canoé migraine, etc...), La Maz qui m'a refilé certaines expressions dont j'aimerais réussir à me débarrasser (mais say whaaaat?!) et Marion, qui m'a toujours laissé squatter dans son (ses) bureau(x). Un énorme big up à ceux de ma génération. Modeste, ravi qu'on ait enfin réussi à te sortir (le dernier jour !!!). Margouw, merci de m'avoir souvent hébergé et d'avoir toujours été partante pour prendre un apéro au soleil et discuter de tout et de rien. J'ai commencé à écouter un peu de musique germande par ta faute, mais bon, j'ai pas continué :P. énorme ci-mer à Pierrot, mes souvenirs d'école d'inge m'ont souvent permis de tenir le cap. Je suis content d'avoir poursuivi cette amitié en thèse et d'avoir pu vivre et partager cette expérience avec toi (et souvent m'en plaindre d'ailleurs). Coucou à ceux qui nous suivent aussi. Le premier ne nous suit pas vraiment : ancien de la meilleure promo, il a su se faire attendre, et l'attente en valait la peine ! Nico barbe rousse, ce positiviste râleur, n'a jamais refusé une pinte ou une infusion. Il a bien contribué à booster mon moral tout en détruisant ma santé, et j'en redemanderais sans hésiter ! Quand j'ai vu la tête des suivants... Melchior et Yves... j'ai eu peur pour l'avenir du labo ! Le peu de temps qu'on a pu passer ensemble a (un peu) réussi à me rassurer. Mes doutes ne se sont vraiment dissipés qu'en rencontrant le grand dernier, Paul, qui semble être un chic type. Merci enfin à Sophie pour ces grosses marrades, et à Fatima. J'aimerais aussi remercier d'autres personnes importantes pour moi à Nancy. Je pense à Guillaume qui m'a hébergé au moins 6 mois mis bout à bout (en me laissant l'appart pour moi tout seul la moitié du temps !). De très nombreux top moments passés ensembles, comme une suite de vidéos joyeux anniv' au Médiéval, inclassable. Je pense aussi à Seb (alias Ternouille, je suis à peu près sûr qu'on l'a validé ce surnom) avec qui j'ai partagé tous les grands moments d'une thèse et d'une relation amoureuse (chacun d'son côté hein). Merci d'avoir toujours pris le temps de me voir, même quand t'étais super busy, genre à la fin de ta rédaction. Enfin, je pense à Boss, partenaire de soutenance, avec qui j'me tape des bonnes barres de rire.

À toutes ces personnes, j'aimerais ajouter d'autres amis grâce à qui cette période a été un bon petit bout de vie. À Montpellier déjà, avec Pascaloum, ce genre d'ours geek qui vit dans le noir et mange des trucs bizarres mais qui me met ma pâtée au lac du Crès, Claire, ma coloc' super badeuse qui a un ratio sport/bouffe proche de l'infini, Fabrice, le papa de la bande, qui a le même ratio mais proche de 0 (voir négatif !) et qui doit être la personne la plus accueillante que je connaisse ; Nico epic sax guy pour qui j'aurais une énigme : Mon coup n'est pas fatal..., merci de me faire ces top piqures de rappel sur le gaming. Merci aussi à Elsa pour faire les mêmes rappels mais avec les mangas, on aura finalement réussi à te faire voir l'appart... Coucou à JBouille l'espiègle, Læti et les matchs de hockey, Vincent qui n'a toujours pas finit notre mirabelle, et Violette et Alban. Une grosse contraction d'pèque pour Cyril Monmout' qui a été mon coach et nutritionniste radicalisé au bio local pendant presqu'un an et à cause de qui j'ai fini par bouffer d'la sardine presque tous les midis. Hola à Fernando (¡Sí

señor!) et son double kebab, Hamid, et Guillaume. Coucou à Mostfa aussi, mon prédécesseur, même si on ne sait jamais où il se cache ! Toujours à Montpellier, j'ai une petite pensée pour les membres du Peru gangg même s'il est aujourd'hui dissout ; je pense en particulier à Coralie qui m'a fait rencontrer ma moitié et qui est devenue une bonne amie depuis. Coucou à tous ceux qu'Inés m'a fait rencontrer dans le Sud : Julia, Aurore, Imène, Camille, les musiciens et profs de l'Hérault, et les survivants du DE. En dehors de Montpellier, je remercie très très évidemment Toutoune, mon deuxième frère, avec qui je passe toujours des moments où je suis heureux, mais genre, heureux. Je fais un bisou à Domi, mon médecin référent, au reste des Heude, et à Laure au passage. Idem pour Doc, (alias Coloc' !!), qui a toujours une bonne histoire de travail sans relâche à la plage à nous raconter. Nos parties de ssbb me manquent pas mal. J'ai d'ailleurs un peu honte de ne toujours pas avoir réussi à aller voir ces deux loustics, mais je compte bien régler ça. Petits bisous à Anaïs et Matthieu qui organisent nos soirées nouvel an, à Souphie, Marianne, Vixou, Mathilde, Magali et tous les autres géoliens et géoliennes (secte oblige).

Un tendre merci à ma famille. En tout premier lieu, à Maman, qui a relevé ce défi monstre de nous élever toute seule, avec mon frère, depuis le début de notre adolescence. Chapeau bas l'artiste, je suis fier de dire qu'on s'en est pas mal sorti tous les deux, et c'est grâce à toi. À mon père (alias El Padre), qui nous fait rêver avec ses photos, vidéos, randos, et qui organise des voyages au top niveau ! À mon grand frère, ce super héros imbattable avec qui je suis toujours aussi complice. À mon petit neveu, Solan, qui grandit trop vite. À mes grands-parents : pépé, mémé Ninja, et mémé Bisounours ; je les aime tout particulièrement. Merci à pépé Jacques et Eveline d'être aussi venus le jour de ma soutenance, c'était une belle surprise. Merci aussi à Philippe, bien que tardivement, je me rends compte que tu as fini par occuper une place paternelle dans ma vie. Et merci enfin à tous les tontons, tantes, cousins, et belles-familles : Gigi, Maxman, Mathis trop grand, Bé-Anne, Olivier, Fabienne, Solène, Brice, Eli, Brigitte, Cath, Matoche, Toto, Rafafou, Bruno, Castor, Nathalie, Océane.

Ahora quería agradecer a los relacionados de Inés que me han aceptado tan rápidamente y calurosamente. Un gran gracias a Teresa y Julio que me han tratado como un hijo desde el primer día y con quien me gusta mucho pasar la siesta delante de la tradicional pelí mala. Gracias a Estefa y Alvaro, pero lo siento Estefa, yo creo que va a ser difícil dar una vuelta en avión ahora... Muchas gracias a Inés y Glyn por sus historias y aventuras de pareja muy cómplice, espero que podamos, con Inés, tener una vida tan maravillosa como la que habéis tenido juntos los dos. Hola también a todos los Benis aunque sois demasiados para nombraros a todos. Gracias también a los amigos de Inés que he tenido el placer de encontrar y apreciar: Adri, Alberto, Kike, Laura, Elena, Carlos, Beita, Pilar, Sara, Juan, Carlos ...

J'aimerais enfin remercier, du plus profond de mon petit cœur qui bat pour elle, ma beauté, ma chérie, ma douce ma colombe, celle pour qui j'aime utiliser toutes les formules les plus gnangnans, Inésita mi wapa ! Tu as vécu tous les moments de cette thèse : les heureux, les surprenants, les stressants, et les plus difficiles. Tu as su partager ma pression et m'aider à la supporter à deux, et tu m'as aussi permis de profiter pleinement des meilleurs moments. Sans toi, je ne serais pas ressorti de ces trois ans dans le même état, et j'espère que l'on puisse vivre toutes nos prochaines aventures au moins aussi bien. Merci. Je t'aime.

Introduction

Geology is the science of the historical evolution of our planet and its subsurface. The methodology developed to assess this evolution considers the description of the present-day nature. It then elaborates multiple concepts and scenarios to explain the sequence of geological events (e.g., magmatism, deformation, erosion, sediment transport, deposition, compaction, diagenesis, fluid circulation) that led to the observations. The deduced empirical laws are also used to estimate missing descriptions and predict future phenomena. Geology is mandatory to deal with challenges impacting our day to day lives, such as earthquake prediction and other geo-hazards, estimation of natural resources and, more recently, the elaboration of deep geological repositories.

These challenges are difficult to assess mainly because of the lack of visibility in the subsurface: accessible data are often sparse and uncertain. Consequently, the data are generally used to first, evaluate current geological structures and physical properties, and then, perform estimations and simulations. The expert is supposed to consider all the approximations and uncertainties at each step of this process to decide.

Geomodeling brings useful tools to such an approach. For instance, it uses mathematics and informatics to apply geological concepts and expertise to a given study. It gathers in a same numerical model the information from various disciplines such as geophysics, geomechanics, petrology, sedimentology, and structural geology. It thus helps at cleaning and interpreting available data and at simulating possible scenarios with numerical methods.

Structural modeling is a key step in the construction of geomodels. It deals with the 2D and 3D descriptions of the geometry and connectivity existing in the geological units, such as folds, faults, unconformities in stratigraphic sequences, and envelope surfaces of intrusive geobodies (Mallet, 2002). The modeled structures represent the numerical support to the estimation of physical properties and related predictions. For instance, if evidences of oil and gas are found, structural modeling may help at estimating the geometry of the reservoir and therefore the volume in place. If fluid simulations are used to estimate oil recovery, these would be performed on the modeled structures. Consequently, structural modeling has a direct impact on the quality of a study.

A first approach in structural modeling, called explicit modeling, is to model the contact surfaces between geological units with parametric and/or polygonal surfaces (Fremming, 2002; Mallet, 2002). These surfaces are generally modeled separately, so post-processing steps involving projection and truncation rules are often performed to construct a model. Another approach, called implicit modeling, is to represent the subsurface with one or several volumetric functions (Calcagno *et al.*, 2008; Caumon *et al.*, 2013). A contact surface is then given either by an iso-value or by a discontinuous jump in these functions. Compared to explicit modeling, the advantage of implicit modeling is that all the data are considered jointly, so the post-processing steps are reduced or completely avoided. We choose the implicit modeling approach for this reason.

Implicit structural modeling uses numerical data (e.g., points and vectors) interpreted from field data to constrain the created functions in space and, therefore, to interpolate between the observations and extrapolate in areas where data are missing. There exists two ways to construct such functions: supporting the interpolation with the numerical data themselves (e.g., the Potential Field Method, Lajaunie *et al.* (1997); Chilès *et al.* (2004); Hillier *et al.* (2014)) or with a background mesh (e.g., the Discrete Smooth Interpolation, Mallet (1992); Frank *et al.* (2007)). In each case, the respective assumption of discretization of the function is inherent to the modeling problem.

Over the past twenty years, these two approaches have proven their reliability in many applied industrial and academic studies (Maxelon, 2004; Collon *et al.*, 2015; Philippon *et al.*, 2015). However, they both hold limits when dealing with specific geological cases, such as fault networks with

complex geometries, problems of thickness variations in the stratigraphic layers, or under-sampled structures with high curvatures. Dealing with such issues typically involves the modification and/or the addition of numerical data (Hillier *et al.*, 2014; Laurent, 2016; Laurent *et al.*, 2016; Grose *et al.*, 2017; de la Varga *et al.*, 2019). For instance, minor faults may be ignored in complex fault networks, and under-sampled areas may be re-sampled based on the expected structures. Although these solutions are efficient, they imply a non-negligible part of subjectivity from the expert. Another approach is to preserve the data and improve the algorithm instead, such as integrating new types of constraints (Chilès *et al.*, 2004; Frank *et al.*, 2007; Caumon *et al.*, 2013), or imposing anisotropy on the interpolation (Aug, 2004; Gonçalves *et al.*, 2017). As the modeling problem is dependent on the chosen discretization (i.e., on the numerical data or on a mesh), changing the discretization imposes to change the modeling problem too. This limits the opportunities for improvement of the existing algorithms. In addition, it limits the possible links between these algorithms: new functionalities generally emerge independently in the two literatures. In the same idea, the possible benefits from the existing literature may be limited when creating new algorithms with potentially attractive discretizations.

In this thesis, we suggest a continuous problem for implicit structural modeling. It allows us to pose a problem independent from any discretization. The links between this problem and the existing methods are discussed, and most of the existing functionalities are integrated. Therefore, this problem may constitute a common formulation to create new methods for implicit structural modeling. In this context, a review of numerical methods adapted to solve the proposed modeling problem is given and two examples of discretizations are developed and compared. The created methods use mesh reduction concepts which enable new techniques to manage stratigraphic discontinuities. Some possible modifications of the continuous problem to address specific geological settings are also suggested.

The contributions of the thesis are:

- A continuous framework for implicit structural modeling. The problem is posed as a minimization of a sum of functionals: discrete functionals for the data constraints and continuous functionals for the regularization. Several data constraints and their associated functionals are presented: data points value, increment data points, normal gradient data, tangent data, polygonal lines and surfaces, hard data points and inequality data. In the standard setting of this framework, the bending energy is used for regularization.

- A structural modeling method using Moving Least Squares functions. We propose to discretize this thesis's continuous framework with these local functions centered on nodes regularly sampled in the domain of study. This sampling brings stability arguments to construct the interpolation functions. Also, faults and unconformities are handled with the visibility criterion and/or with the transparency method.

- A second modeling method discretizing the framework on a Cartesian grid with the Finite Element Method functions and using the concept of ghost nodes to handle faults and unconformities. The method approximates the second order derivatives of the bending energy with the finite difference method.

- A first modification of the continuous equations to handle thickness variation issues by considering the physical meaning of the bending energy. The concept of the method is independent from the discretization.

- Two other modifications of the continuous equations to impose global and local anisotropy on the modeled structures. The proposed concepts are independent from the discretization.

- Four C++ codes for structural modeling: a 2D prototype of the Potential Field Method, a 2D and a 3D prototypes using the Moving Least Squares functions, and a 2D prototype using the Cartesian grid with ghost nodes. These are standalone codes and can compile on Linux, Mac and Windows. The 2D results are visualized with Gnuplot and the 3D results are visualized with an internal software of Schlumberger called Mere. Also, an API of the 3D prototype was developed to integrate it in a plugin to Petrel constructing structural models with borehole image interpretations.

This thesis is structured into six chapters.

In Chapter 1 (p.23), we present the general problem of implicit structural modeling and a review of its two most known numerical methods: the Discrete Smooth Interpolation and the Potential Field Method. The advantages and limits of both methods are discussed based on public articles only. The review is illustrated on synthetic examples using simplified 2D codes of each method.

In Chapter 2 (p.59), we define a continuous problem for implicit structural modeling. First, we present a review of numerical methods focused on meshless concepts with their applicability to implicit structural modeling. Then, we define the general framework of the thesis as a minimization of arbitrary functionals, and the standard version of this framework as a spatial regression of data points penalized by the bending energy. Other typical data constraints in structural modeling are presented and adapted to the general formalism. The equations are discretized in an abstract manner, although the Bubnov-Galerkin scheme is further developed as key to the thesis.

In Chapter 3 (p.91), we propose a first discretization of the continuous problem using the Moving Least Squares functions to perform the interpolation and using optic criteria to handle the structural discontinuities. The method is illustrated with several data constraints in 2D and in 3D, and sensibility tests are performed on a synthetic cross section in 2D.

In Chapter 4 (p.119), we propose a second discretization of the continuous problem using the Finite Element Method functions for the interpolation and using the concept of ghost nodes to handle the structural discontinuities. A few results are given in 2D with comparisons to the first discretization. Some 3D results computed with a similar implicit method are also presented.

In Chapter 5 (p.135), we modify the continuous problem to deal with the limits of the bending energy. Some ideas are presented to handle typical limits met with smoothing techniques, such as thickness variation and fold anisotropy issues. Other opportunities of the proposed formalism are further discussed. The use of fictional data constraints, or of better quality, is mentioned and illustrated, but not investigated as already well documented in the literature.

Finally, we present other examples of applications of the proposed formalism in Chapter 6 (p.151). This includes a prototype for structural modeling with wellbore image interpretations, salt surface modeling, and restoration using mechanical equations.

Context of the thesis

This thesis was realized under a contract called Conventions Industrielles de Formation par la Recherche (CIFRE). The CIFRE contract is a cooperation between the Association Nationale de la Recherche et de la Technologie (ANRT) and an industrial company. Both entities co-finance the thesis, and the Ph.D. student divides his working time between the participating company and an attached university. Therefore, although the subject is research oriented, it is often applied to industrial needs.

In the case of this thesis, the contract was defined between Schlumberger Petroleum Services and the University of Lorraine (RING, Georessources). Schlumberger is the world's leading oilfield services company supplying technology, information solutions and integrated project management that optimize reservoir performance for customers working in the oil and gas industry. Schlumberger Information Services is an entity of Schlumberger which partly deals with software solutions and other digital technologies for customers. This thesis took place in Montpellier Technology Center (MpTC) that specializes in wellbore and structural geology software. The student was in direct relation with the geomodeling team which is responsible of a structural modeling algorithm called Volume Based Modeling and other geomodeling tools. The RING team also specializes in geomodeling. It addresses four main subjects: stochastic structural and stratigraphic modeling, stochastic sedimentary and diagenetic objects description, adaptive gridding and scale management, and physical processes. The student spent approximately thirty percent of his time with this team and seventy percent in the company.

Publications and international conferences associated to this thesis

J. Renaudeau, F. Maerten & E. Malvesin [2018b]. Geologic structural model generation. Owner: Schlumberger, (US Patent 20180347320).

J. Renaudeau, M. Irakarama, G. Laurent, F. Maerten & G. Caumon [2019a]. Implicit modelling of geological structures: a cartesian grid method handling discontinuities with ghost points. Boundary Elements and other Mesh Reduction Methods XXXXI, 122:189.

J. Renaudeau, E. Malvesin, F. Maerten & G. Caumon [2019b]. Implicit structural modeling by minimization of the bending energy with moving least squares functions. Mathematical Geosciences. pp. 1-32.

M. Irakarama, G. Laurent, **J. Renaudeau** & G. Caumon [2018]. Finite difference implicit modeling of geological structures. In: 80th EAGE Conference and Exhibition 2018, Copenhagen.

M. Marin, C. Castagnac, O. Ferrer & **J. Renaudeau** [2018]. Physical models as analogues for basin and petroleum systems modeling: Example of temperature and pressure estimations around salt bodies. In: GeoMod2018, Barcelona.

J. Renaudeau, E. Malvesin, F. Maerten & G. Caumon [2018a]. Implicit structural modeling with local meshless functions. In: 80th EAGE Conference and Exhibition 2018, Copenhagen.

J. Renaudeau, F. Maerten, E. Malvesin & G. Caumon [2018c]. The weighted curvature minimization: a correction to thickness variations in implicit structural modeling. In: 19th IAMG Annual Conference 2018, Olomouc.

J. Renaudeau, F. Maerten [2018]. Accelerator award at FIZZ forum 2018. Schlumberger international forum of innovation. Palo Alto, California.

Introduction (Français)

La géologie est la science de l'évolution de notre planète et de sa lithosphère. La méthodologie développée pour discerner cette évolution considère la description de la nature telle qu'elle est aujourd'hui. Une multitude de concepts et de scénarios sont alors élaborés pour tenter d'expliquer la séquence des évènements géologiques (e.g., magmatisme, déformation, érosion, transport des sédiments, déposition, compaction, diagénèse, circulation de fluides) qui sont à l'origine des objets observés. Les lois empiriques qui en sont déduites sont également utilisées pour estimer les zones non observées et prédire les phénomènes à venir. La géologie est donc nécessaire pour un grand nombre de défis impactant notre vie quotidienne, comme la prédiction de tremblements de terre et d'autres géo-hasards, l'estimation des ressources naturelles et, plus récemment, l'élaboration de stockages de déchets radioactifs.

Ces défis sont difficiles à évaluer principalement à cause du manque de visibilité sous la surface : les données accessibles sont souvent éparses et incertaines. En conséquence, les données sont généralement utilisées pour, premièrement, évaluer les structures géologiques et les propriétés physiques actuelles, et ensuite, réaliser des estimations et des prédictions. L'expert est censé considérer toutes les approximations et les incertitudes à chaque étape de ce procédé pour établir ses conclusions.

La géomodélisation apporte des outils pratiques à cette approche. Par exemple, elle permet d'utiliser les mathématiques et l'informatique pour appliquer les concepts géologiques et l'expertise de l'utilisateur à une étude donnée. Elle regroupe, dans un même modèle numérique, l'information de disciplines diverses telles que la géophysique, la géomécanique, la pétrologie, la sédimentologie, et la géologie structurale. Cela aide donc à traiter et interpréter les données disponibles et à simuler plusieurs scénarios possibles avec des méthodes numériques.

La modélisation structurale représente une étape clef dans la construction de géomodèles. Cela concerne la description en 2D et en 3D de la géométrie et de la connectivité des unités géologiques, tels que les plis, les failles, les discordances stratigraphiques, et les surfaces d'enveloppe de corps intrusifs. Les structures modélisées représentent le support numérique utilisé dans l'estimation des propriétés physiques et dans la prédiction des phénomènes qui leurs sont associées. Par exemple, si des traces de pétrole et de gaz sont observées, la modélisation structurale peut aider à estimer la géométrie du réservoir et donc du volume en place. Si des simulations d'écoulement de fluide sont réalisées pour estimer le volume de pétrole qui pourra être récupéré, ces simulations seront effectuées sur les structures modélisées. En conséquence, la modélisation structurale a un impacte direct sur la qualité d'une étude en géomodélisation.

Une première approche en modélisation structurale, nommée modélisation explicite, consiste à modéliser les surfaces de contact entre unités géologiques avec des surfaces paramétriques et/ou polygonales. Ces surfaces sont généralement modélisées séparément, donc des étapes de post-traitements impliquant des règles de projection et de troncature sont souvent utilisées pour construire un modèle. Une autre approche, nommée modélisation implicite, consiste à représenter le sous-sol avec une ou plusieurs fonctions volumétriques. Une surface de contact est alors donnée soit par une iso-valeur, soit par un saut discontinu dans ces fonctions. Comparé à la modélisation explicite, l'avantage de la modélisation implicite est que toutes les données sont considérées en même temps, donc les étapes de post-traitements sont réduites ou complètement évitées. Nous choisissons la modélisation implicite pour cette raison.

La modélisation structurale implicite utilise des données numériques (e.g., des points et des vecteurs) interprétées sur les données de terrain pour contraindre, dans l'espace, les fonctions créées, et pour interpoler entre les observations et extrapoler dans les zones où il manque des données. Il existe deux façons de construire ces fonctions : en supportant l'interpolation avec les données numériques directement (e.g., "Potential Field Method") ou avec un maillage de fond (e.g., "Discrete Smooth Interpolation"). Dans chacun des cas, la discrétisation choisie est inhérente au problème de modélisation.

Au cours de ces vingt dernières années, ces deux approches ont prouvé leur efficacité dans un grand nombre d'études appliquées à l'industrie et à la recherche académique. Cependant, elles connaissent toutes deux des limites pour traiter des cas géologiques spécifiques, tels que les réseaux de faille avec des géométries complexes, les problèmes de variations d'épaisseur dans la stratigraphie, ou les structures sous-contraintes à fort plissement. Résoudre ce genre de problèmes implique généralement la modification et/ou l'ajout de données numériques. Par exemple, les failles mineures peuvent être ignorées dans les réseaux de failles complexes, et les zones sous-échantillonnées peuvent être rééchantillonnées

en rapport aux structures attendues. Bien que ces solutions soient efficaces, elles impliquent une part non négligeable de subjectivité venant de l'expert. Une autre approche consiste à préserver les données telles qu'elles sont et améliorer l'algorithme, tel qu'intégrer la possibilité d'utiliser d'autres types de contraintes, ou d'imposer une anisotropie dans l'interpolation. Comme le problème de modélisation est dépendant de la discrétisation choisie (i.e., des données numériques ou du maillage), changer la discrétisation impose de changer également le problème de modélisation. Cela limite les opportunités d'amélioration des algorithmes existants. De plus, cela complexifie les liens possibles entre les deux méthodes : les nouvelles fonctionnalités émergent souvent indépendamment dans les deux littératures. Dans la même idée, les bénéfices possibles venant de la littérature peuvent être limités lors de la création de nouveaux algorithmes avec des discrétisations potentiellement attractives.

Dans cette thèse, nous suggérons un problème continu pour la modélisation structurale implicite. Cela nous permet de poser le problème indépendamment de toute discrétisation. Les liens entre ce problème et les méthodes existantes sont discutés, et la plupart des fonctionnalités existantes sont intégrées. Ainsi, ce problème pourrait constituer une formulation commune pour créer de nouvelles méthodes pour la modélisation structurale implicite. Dans ce contexte, un état de l'art sur les méthodes numériques adaptées à la résolution du problème de modélisation proposé est donné, et deux exemples de discrétisations sont développés et comparés. Les méthodes créées utilisent des concepts de réduction de maillage qui rendent possible de nouvelles techniques pour gérer les discontinuités stratigraphiques. Quelques modifications possibles du problème continu pour traiter de certains cas géologiques spécifiques sont également suggérés.

Les contributions de la thèse sont les suivantes :

- Un cadre d'équations continues pour la modélisation structurale implicite. Le problème est posé comme une minimisation d'une somme de fonctionnelles : des fonctionnelles discrètes pour les contraintes de données et des fonctionnelles continues pour la régularisation. Plusieurs contraintes de données et leurs fonctionnelles sont présentées : données ponctuelles avec valeurs et avec incréments, gradients normaux, tangentes, lignes et surfaces polygonales, données dures et données d'inégalité. Dans la version standard du problème continu, l'énergie de courbure est utilisée pour la régularisation.

- Une méthode de modélisation structurale utilisant les fonctions des moindres carrés glissants. Nous proposons de discrétiser le problème continu de cette thèse avec ces fonctions locales centrées sur des nœuds régulièrement échantillonnés dans le domaine d'étude. Cet échantillonnage apporte des arguments de stabilité lors de la construction de ces fonctions d'interpolations. Également, les failles et les discordances sont gérées avec le critère de visibilité et/ou avec la méthode de transparence.

- Une seconde méthode de modélisation qui discrétise le problème continu sur une grille Cartésienne. Les fonctions de la méthode des éléments finis sont alors utilisées, ainsi que le concept de nœuds fantômes pour la gestion des failles et des discordances. La méthode approxime les dérivées secondes de l'énergie de courbure avec les différences finies.

- Une première modification des équations continues proposées pour gérer les cas de variation d'épaisseur en utilisant le sens physique de l'énergie de courbure. Le concept de cette méthode est indépendant de la discrétisation choisie.

- Deux autres modifications des équations continues pour imposer une anisotropie globale et locale sur les structures modélisées. Les concepts proposés sont indépendants de la discrétisation.

- Quatre codes en C++ pour la modélisation structurale : un prototype en 2D de "Potential Field Method", un prototype en 2D et un en 3D utilisant les moindres carrés glissants, et un prototype en 2D utilisant la grille Cartésienne avec des points fantômes. Ces codes sont autonomes et peuvent compiler sous Linux, Mac et Windows. Les résultats en 2D sont visualisés avec Gnuplot, et les résultats en 3D sont visualisés avec un logiciel interne à Schlumberger nommé Mere. Également, une API du prototype en 3D a été développée pour intégrer le code à un plugin pour Petrel qui permet la construction de modèles structuraux à partir d'interprétations d'imageries de puits.

Résumé du Chapitre 1

Ce chapitre introductif présente un état de l'art de la modélisation structurale utilisant des méthodes implicites.

La modélisation structurale, au sens large, tente de décrire les géométries et les contacts caractérisant les grandes unités géologiques du sous-sol. Ces unités regroupent, d'une part, les couches stratigraphiques et, d'autre part, les volumes de roches intrusifs. Les structures modélisées décrivent donc les relations existantes entre ces unités, ainsi que les figures liées aux évènements tectoniques qui les affectent, tels que les plis et les failles.

Les structures du sous-sol sont rarement modélisées dans toute leur complexité. En effet, seules les surfaces de contact entre les unités géologiques sont généralement représentées. Cela regroupe (i) les horizons, qui décrivent l'interface existante entre deux couches stratigraphiques, (ii) les discordances, qui peuvent mettre en relation plusieurs ensembles de strates, (iii) les failles, qui déplacent les unités géologiques les unes par rapport aux autres, et (iv) les enveloppes de volumes intrusifs. La modélisation de ces surfaces est contrainte dans l'espace par des données numériques, tels que des points, des vecteurs, des lignes et des surfaces. Ces données sont obtenues après interprétation de données de terrain ; elles peuvent donc être bruitées, éparses et irrégulièrement échantillonnées. De plus, les surfaces modélisées doivent former un tout géologiquement réaliste, c'est à dire que le modèle structural doit respecter les concepts géologiques préalablement définis sur des modèles analogues.

Dans cette thèse, nous utilisons l'approche implicite pour créer ces modèles surfaciques. Cela consiste à définir une fonction volumétrique, nommée fonction implicite, sur l'ensemble du domaine d'étude. Chaque horizon est représenté par une iso-valeur de cette fonction, et chaque surface de discontinuité (discordance, faille, intrusion) est représentée par un saut discontinu dans la fonction. Cette fonction est donc strictement continue dans tout le domaine d'étude sauf au niveau des surfaces de discontinuité. Le saut dans la fonction doit alors correspondre à la géométrie de la surface concernée. Ainsi, il est nécessaire que ces surfaces soient prédéfinies et données en entrées aux algorithmes implicites. L'avantage de cette approche est que la fonction crée respecte intrinsèquement certaines conditions de validité pour un modèle géologiquement acceptable.

Il existe deux méthodes numériques pour modéliser les structures géologiques avec l'approche implicite. La première se nomme "Discrete Smooth Interpolation" (DSI). Elle construit la fonction implicite à l'aide d'un maillage, i.e., une discrétisation du domaine d'étude avec des éléments finis conformes. La fonction est définie par une somme de fonctions d'interpolations, elles-mêmes définies sur les éléments du maillage. Plusieurs types de contraintes existent pour forcer la fonction implicite à respecter les données numériques. Nous ne présentons que les contraintes ponctuelles de type évidence d'horizon qui imposent une valeur fixe en chaque point. Comme le nombre de points de donnée peut être inférieur au nombre de degrés de liberté du maillage, ce problème de modélisation peut être sous-contraint. Il est donc proposé, dans DSI, d'ajouter des contraintes supplémentaires afin de résoudre un problème sur-contraint au sens des moindres carrés. Ces dernières sont définies par un critère de rugosité qui impose aux surfaces modélisées d'être les plus lisses possible. Enfin, le saut discontinu le long des failles est imposé par un maillage conforme aux failles, c'est à dire que les faces des éléments en suivent la géométrie.

La seconde méthode se nomme "Potential Field Method" (PFM). Elle construit la fonction implicite directement sur les données numériques. Pour ce faire, cette méthode utilise des concepts comme le krigeage dual ou les bases de fonctions radiales. Dans les deux cas, l'interpolation est supportée en tout point de l'espace par l'ensemble des données numériques. Les surfaces modélisées sont également supposées lisses. Nous ne présentons que les contraintes nécessaires à créer une version fonctionnelle de PFM selon la littérature. Cela comprend une contrainte ponctuelle de référence, des contraintes

ponctuelles dites d'incréments, et des contraintes de vecteur perpendiculaires aux couches géologiques. Enfin, le saut discontinu le long des failles est imposé par des fonctions polynomiales associées à des fonctions de saut, elles-mêmes définies sur des zones de faille volumétriques.

Des versions 2D de ces deux méthodes sont utilisées pour illustrer leur comportement sur des cas synthétiques. Le code de DSI est tiré du logiciel "Volume Based Modeling" (version simplifiée en 2D), et le code de PFM a été implémenté au cours de cette thèse. Les résultats présentés font donc témoins du comportement de ces deux méthodes lorsqu'elles sont uniquement fondées sur les publications.

Nous discutons également des avantages et inconvénients des deux méthodes, ce qui nous permet d'en identifier des limites communes. En particulier, la gestion de réseaux de failles complexes est possible mais les créations respectives de maillages conformes et de zones de faille peuvent s'avérer laborieuses. Également, l'hypothèse des surfaces lisses est peu efficace pour certains cas géologiques, tels que les problèmes de variations d'épaisseur dans les couches stratigraphiques, ou l'extrapolation de structures plissées dans des zones peu échantillonnées.

L'hypothèse des surfaces lisses est toujours formalisée de façon discrète : par des contraintes écrites sur les éléments du maillage dans DSI, et par des fonctions d'interpolation lisses centrées sur les données dans PFM. Ainsi, les futures améliorations de ces deux méthodes devront nécessairement être écrites selon ces critères discrets. Pour bénéficier d'un plus grand nombre de méthodes numériques pour résoudre les limites mentionnées, nous proposons dans la suite de cette thèse de poser le problème de modélisation structurale par méthodes implicites comme un problème continu.

Chapter 1

State of the art of implicit structural modeling

1.1 Structural modeling

Structural geology is partly concerned with the characterization of geological bodies and their deformation in two and three-dimensional spaces. From the millimetric to the kilometric scale, these descriptions help at understanding the local and regional tectonic contexts (Fossen, 2016). Assessing accurately all the specificities described in this field would be improbable, hence the simplification by conceptual models. Structural modeling is a generic term for numerical algorithms constructing one or several models of geological structures from a set of interpreted data. This section narrows the scope of the structures being studied in this thesis and it further describes the concept of structural modeling.

1.1.1 Stratigraphic structures

Geological structures concern two different types of geological units: stratigraphic layers and intrusive geobodies. This thesis deals mainly with the former ones even though the construction of salt envelope surfaces is also discussed.

A stratigraphic layer, or stratum, is a layer of homogeneous sedimentary or magmatic rock. Sedimentary layers are formed by a successive horizontal or sub-horizontal deposition over geological times. A stratigraphic sequence is then a set of sub-parallel layers of sedimentary rocks. The different sediment compositions and physical conditions of each separate geological age is at the origin of the various geological layers (Cross & Homewood, 1997).

The stratigraphic sequences can be affected by tectonic events which modify the geometry of the concerned layers. Their deformation may involve folding and/or shearing. When such events occur during the sedimentation, the deposition can be abnormally favorized in space creating layers with varying thickness laterally. As all these phenomena can occur coincidentally, stratigraphic structures may have complex geometries involving folds, faults and thickness variations in the layers.

The stratigraphic sequences can also be affected by intrusive geobodies such as salt and magmatic volumes of rock. When displacing in sedimentary basins, their movements induce complex deformations in the layers. There exists a wide range of unique structures resulting from those deformations, and their study represents an active branch of research (e.g., salt tectonics (Hudec & Jackson, 2007)).

Erosion events are also to be considered. They correspond to a spatial redistribution of rock particles: these are removed from one location, displaced, and then deposited in another location. A system that underwent ablation during an erosion event can, later, enter a new phase of sedimentation. The newly deposited layers may not be parallel to the formerly deposited ones. In this case, they are sorted in separate stratigraphic sequences, and the contact surface between those sequences is called an unconformity. As distinct tectonic events may affect a system during the successive erosion and sedimentation phases, many geometries in the stratigraphic structures are possible.

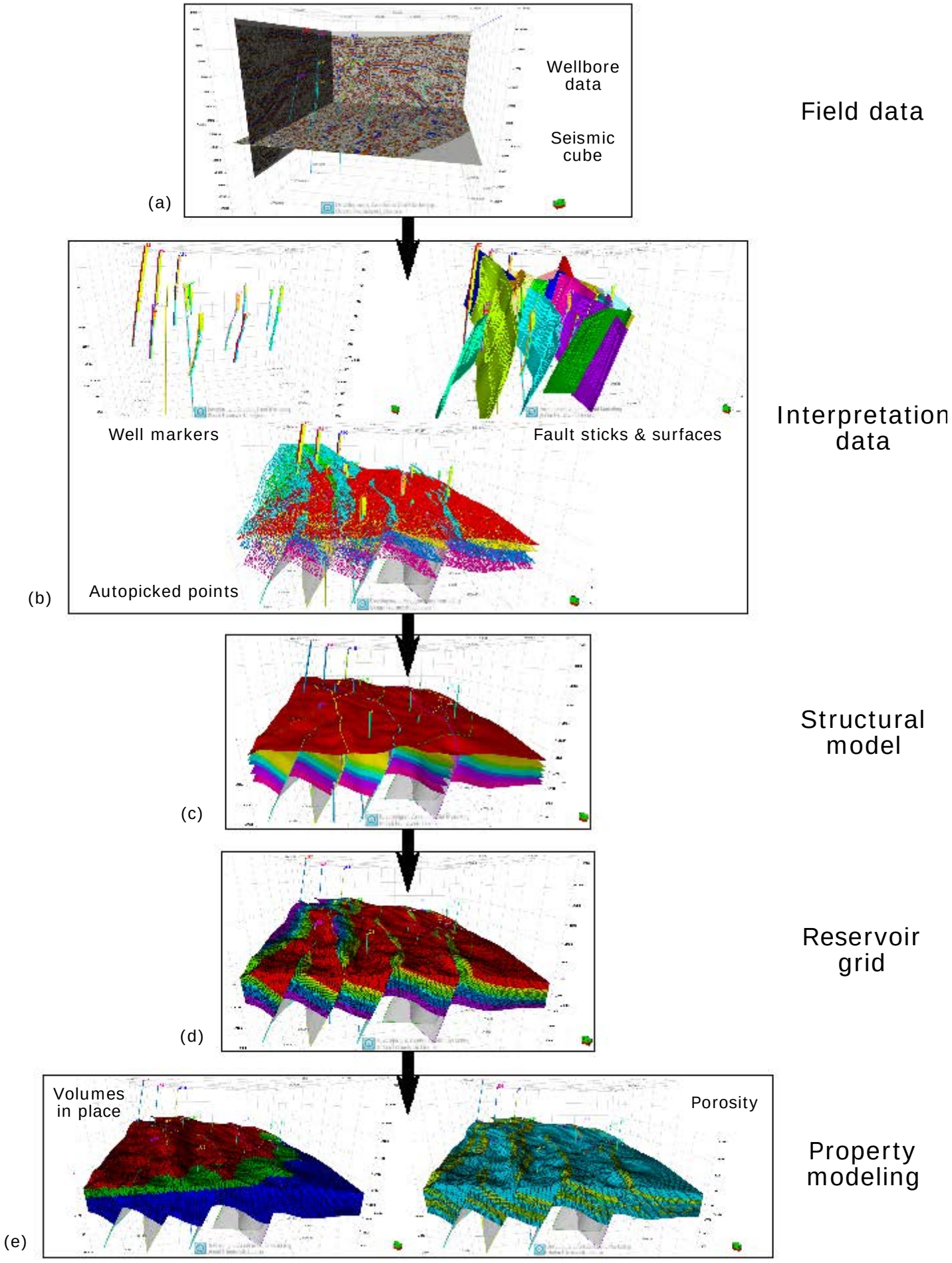


Figure 1.1: *Example of geomodeling workflow for a geoscience study in the oil and gas industry.*

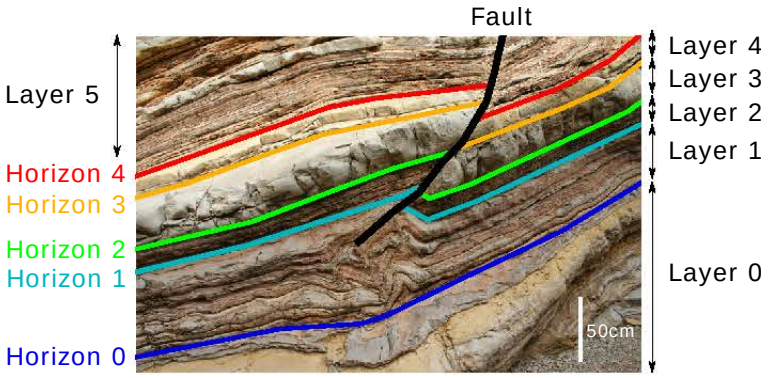


Figure 1.2: *Interpreted outcrop where thin complex structures are approximated as uniform sets, the layers, delimited by interfaces, the horizons and the fault (photography from imagenesmi.com). Horizons and faults are modeled by lines in this 2D outcrop and would be modeled by surfaces in a 3D space.*

1.1.2 Motivation to model the stratigraphic structures

In addition to better understanding our planet and its mechanism, assessing stratigraphic structures plays a key role in managing natural resources.

It is decisive for the oil and gas industry. It helps at evaluating the geometry of candidate layers for source rock, reservoir rock and cap rock. This is essential to estimate the volumes of oil and gas in place, the compartmentation of the system, and to simulate the fluid flows during extraction to optimize the production. Figure 1.1 shows a global workflow of a geoscience study estimating oil and gas resources using geomodeling. It starts with the pre-processed data acquired from the field and goes all the way through interpretation of numerical data, modeling of geological structures, gridding, property population, and estimation of the volumes of oil and gas in place. The geological structures are thus represented by a model, the structural model (Figure 1.1(c)), which represents a frame for the next steps of the workflow. It is used to create the reservoir grid (Figure 1.1(d)), in which properties are populated into the cells (Figure 1.1(e)). As an example, the volume of oil can be estimated by summing the volume of each cell labelled as containing oil. The other represented physical property, the porosity, is used in fluid flow simulations.

In addition to physical properties, representing stratigraphic structures can also lead to the population of mineral concentrations. It is thus needful for the mining industry where high concentrations of specific minerals, ore volumes, are sought in the subsurface for their economic value, such as hard metals (e.g., gold, silver, copper).

Though other scientific challenges must be dependent on geological structures, these two industrial applications have mainly carried the advances in research on structural modeling over the years. In this thesis, many reference papers and software were sponsored or promoted in either of these industries.

1.1.3 Modeling geological surfaces

The diversity and complexity of natural structures as compared to the lack of available information on the subsurface calls for simplifications. In structural modeling, only the surfaces of contact are represented. This gathers the interfaces between layers, called horizons, and all types of structural discontinuities: the faults and fractures, the stratigraphic unconformities, and the envelopes of intrusive geobodies. It is called a boundary representation model (Gjøystdal *et al.*, 1985; Caumon *et al.*, 2009).

Surface modeling is a strong simplification. For instance, all the possible horizons are rarely represented: several thin layers are often round up as one. A complex combination of different facies and small structures can thus be approximated as a uniform set (Sword Jr, 1991; Mello & Henderson, 1997) (Figure 1.2). Also, displacements in the layers may involve complex deformations in a volume called the fault zone. Nevertheless, faults are approximated as surface objects.

All these approximations are considered in the physical parameters, such as the connected porosity and the permeability. The decisions on these parameters and surface approximations thus have a direct

impact on estimations and predictions performed in geomodeling. They are generally guided by three factors: the studied phenomenon, the different scales, and the available information.

1.1.4 Numerical data and origins

A numerical algorithm relies on numerical objects. In structural modeling, we generally use points, vectors, and parametric or polygonal lines and surfaces. The data acquired on the field are thus digitalized accordingly before structural modeling (Figure 1.1(a) to (b)). As the data can have different origins and natures, the methods of interpretations of the field data and the created numerical objects are diverse.

1.1.4.1 Seismic data

Seismic campaigns evaluate the arrival time and amplitude of a seismic signal emitted from the surface and reflected in depth. After treatment, 2D and 3D seismic profiles covering entirely or partially the domain of study are produced giving a map of impedance contrasts that can be interpreted as contacts between rock units. At this stage, seismic data are represented as pixels on an image and cellular values in a seismic cube. These pixels and cells resolutions may range from the meter to the kilometer scale. Interpretation can then be performed manually by picking (e.g., the fault sticks in Figure 1.1(b)), or automatically following a reflector (e.g., the horizon points in Figure 1.1(b)) (Borgos *et al.*, 2003; Labrunye, 2004). Set of points, polylines and surfaces can be obtained with such interpretations.

1.1.4.2 Wellbore data

Exploration wellbores can produce drilling cores and borehole images (optic, acoustic and electric). Their interpretation can inform on the petrophysics of the rocks, evidences of horizons and faults, and dips and azimuths of their related surfaces. In addition, borehole images have a high resolution (up to the millimeter) and drilling cores are extracted at the surface, so they are described finely. Like seismic profiles, borehole images can be interpreted both manually and automatically. The limit to these observations resides in their restrictive lateral expansion away from the drill hole. Also, the sparsity of these lines as compared to the 3D volume of study makes the correlation between wells challenging (Lallier, 2012; Edwards, 2017). Points and vectors aligned on the drill hole can be obtained with wellbore data, and the petrophysics also help at defining the limits between stratigraphic layers.

1.1.4.3 Outcrop observations

The description of outcrops gives surface observations, from millimetric to kilometric scales. The structures and the lithology of the layers are accessible from the (2D+) topographic surface. Apparent punctual and line evidences of the stratigraphic surfaces and discontinuities, as well as their dips angles and azimuths are obtained from the observation of the outcrops. These data are generally interpreted into geological maps and conceptual 2D cross sections that go over a few hundreds of meters in depth. In the same way as a 2D seismic image, these cross sections can be digitalized by picking.

1.1.4.4 Contextual knowledge

Geological theories, i.e., hypotheses based on the regional context, and similarities with analog sites are subjective but have the strongest impact on a study. They can guide a modeling tool in three ways: add prior knowledge or acceptable simplifications (e.g., smooth modeling, (Mallet, 1992; Lajaunie *et al.*, 1997), global anisotropy (Aug, 2004)), add non observed structures and geometries (e.g., faults, fold periodicity (Laurent *et al.*, 2016; Grose *et al.*, 2017)), and question or reject a model once created.

1.1.4.5 Summary: The nature of the numerical data

All numerical objects used in geomodeling are the result of multiple interpretations of geologists and geophysicists. This concept must be considered when creating and using the related tools. The different natures of field data and techniques of interpretation produce heterogeneous numerical data

in terms of type of objects, quality of information, and spatial distribution (Mallet, 1992, 2002; Carmichael & Ailleres, 2016). The resolution of the acquisition tools influences the reliability of the obtained field data. Wellbore and surface data are thus considered more certain than seismic data when working at basin scale.

The interpretation by manual picking is human related, so the density of numerical data is rarely uniform in space and between picked units (i.e., often clustered in some parts and sparse in others). It can be noisy and involve a lot of missing data such as missing well tops, non-interpreted seismic cross sections, and even unsampled horizons and discontinuities. In comparison, interpretation by automation tools is often much denser and regular but the quality depends a lot on the source image (e.g., noisy and irregular sets of points may be obtained on low-quality seismic profiles). Finally, these numerical data must be considered all together to create one or several common structural models.

1.1.5 Expected models

A structural model is a set of surfaces delimiting the geological units. In addition to honoring the numerical data, these surfaces must also follow a set of rules guided by geological realism. As mentioned in Section 1.1.4.4, these rules are mainly based on similarities with analogue structures and, therefore, the subjective experience of the expert. Also, depending on the geological context of a study, a same modeled structure may be considered geologically realistic or not. In this thesis, we use the general assumption that geological structures should not involve excessively high curvatures in the stratigraphic layers (i.e., the modeled horizon surfaces should not bend abnormally in space, Mallet (1992); Lajaunie *et al.* (1997)). Note that this rule is also subjective as no limit is defined.

Nevertheless, Caumon *et al.* (2004) and Caumon *et al.* (2009) define a set of objective requirements for a boundary representation model to be potentially valid, creating the concept of sealed model. Some of these validity conditions are detailed here and summarized in Figure 1.3.

The horizons cannot intersect, they can only branch onto structural discontinuities. Therefore, if onlap, downlap, toplap and offlap structures are expected, a stratigraphic unconformity must be represented. A horizon is a continuous and infinite surface in the model, except at the boundaries and the discontinuities; it thus cannot vanish laterally in space on its own. For instance, if the correlation between two wells relates a disappearance of a stratigraphic unit, it must be explained by a discontinuity. Finally, horizons should not form closed surfaces as stratigraphic units are rarely affected by ductile deformations.

The structural discontinuities can commonly intersect any type of surface. For the rest, each discontinuity has its own specificities. The faults can vanish laterally in space, indicating the end of spatial displacement between layers. Intersections between faults and other discontinuities should be related to historical fault activity. Like the horizons, the stratigraphic unconformities and intrusive surfaces cannot vanish laterally in space, but they can terminate onto other discontinuities. In addition, intrusive geobodies may form closed surfaces as their deformation can be ductile.

All these validity conditions are necessary for the next steps of the modeling workflow (Figure 1.1). Relations of intersections between layers and discontinuities entrust the creation of a water tight model which is unavoidable for the gridding stage and all the following numerical computations with the current technology. For instance, some model repair algorithms use these conditions not only for validity, but also to enhance computational efficiency (Anquez *et al.*, 2017, 2019).

The underlying uncertainty of the heterogeneous numerical data used in structural modeling, and the volumes in between, is generally assessed with the creation of multiple possible models (Holden *et al.*, 2003). These multiple realizations are generally handled with deterministic algorithms using stochastic distributions of data (Wellmann *et al.*, 2010, 2011; Lindsay *et al.*, 2013; de la Varga *et al.*, 2019), stochastic parameters for model creation (Cherpeau *et al.*, 2010; Godefroy, 2018), or stochastic parameters for perturbing existing models (Caumon *et al.*, 2007; Cherpeau *et al.*, 2010). In this thesis, the aim is to improve the deterministic algorithms themselves to create one realistic model for a given set of data and a given set of parameters. The question of uncertainty in the models is not further discussed, but the developed tools can be used in a stochastic framework.

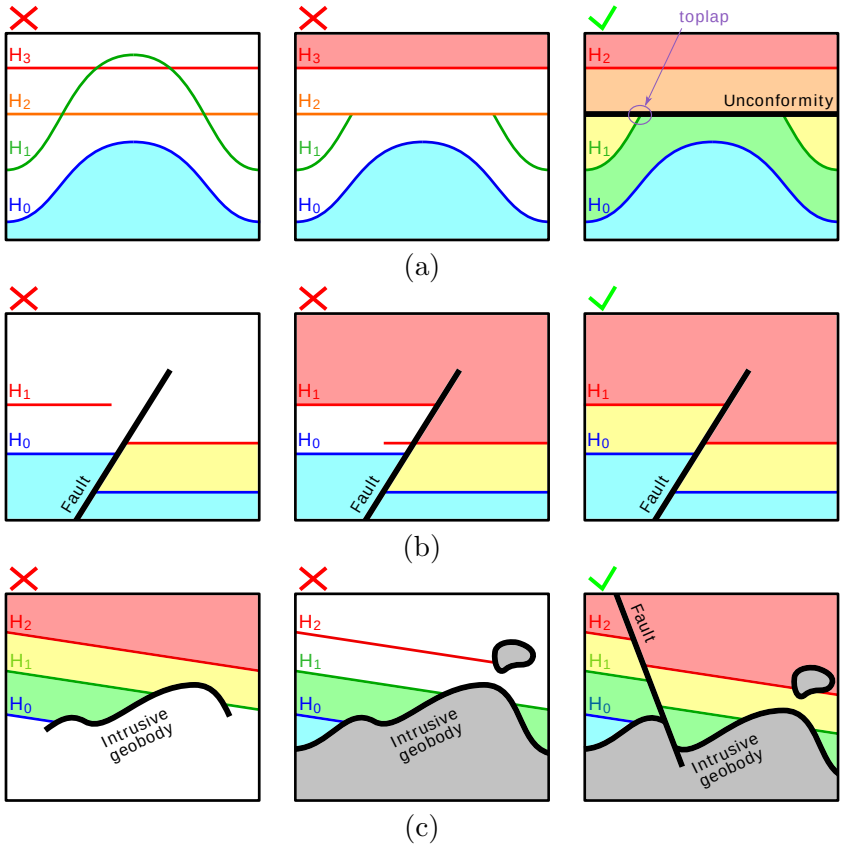


Figure 1.3: *2D Sketches of a few examples of validity conditions for a sealed geological model. Valid and invalid intersections are illustrated between (a) Horizons and a stratigraphic unconformity, (b) Horizons and a fault, and (c) Horizons, a fault, and an intrusive surface (sketches modified from and inspired by* Caumon et al. *(2004) and* Caumon et al. *(2009)).*

1.1.6 Two modeling approaches

The first structural modeling algorithms use pure geometrical concepts to solve the problem. It is called explicit modeling. Each horizon and discontinuity surfaces are considered separately one from another. Best-known explicit methods use either parametric or polygonal surfaces.

Parametric surfaces are described by polynomial equations. They are typically supported spatially by control nodes, such as the non-uniform rational B-splines (NURBS) (Piegl & Tiller, 1997). Parametric surfaces have been used in structural modeling to model fault blocks (e.g., infinite faults) (Gjøystdal *et al.*, 1985), complex geological surfaces (de Kemp & Sprague, 2003) and some conceptual stratigraphic structures (Jacquemyn *et al.*, 2018). In geomodeling, they also have been used to model ore volumes (Sprague & De Kemp, 2005) and sedimentary geobodies (Pyrcz *et al.*, 2005; Ruiu, 2015; Ruiu *et al.*, 2016; Parquer *et al.*, 2017).

Polygonal surfaces are supported by a set of vertices linked by edges and forming together a collection of closed polygonal faces. The faces can theoretically be defined by any plane geometry but are generally based on simple, well defined geometries (i.e., simplexes). Also, a surface could be composed of heterogeneous polygonal faces, but most methods use uniform faces in practice. In structural modeling, some algorithms rely on quadrilateral faces as it is convenient for the gridding step (pillar grids) (Fremming, 2002; Schlumberger, 2010) but they are limited when modeling complex structures such as recumbent folds or envelopes of intrusive geobodies. Other more flexible algorithms rely on triangulated surfaces which allow a better representation of complex structures (Mallet, 1989; Lemon & Jones, 2003).

Once created, both parametric and polygonal surfaces are processed to fit the data while honoring the validity conditions of a geological model. The polynomial equations are inverted and vertices are displaced with kriging or smoothing (Mallet, 1988, 2002; Kaven *et al.*, 2009) constrained by data points positions. Additional truncations, projections, and remeshing are also performed to create

a sealed model (Gjøystdal *et al.*, 1985; Mallet, 2002). These post-processes can be performed automatically but often require user interactions to ensure the model quality, and the number of required interactions increases with the dimensions and complexity of the represented structures.

The second approach for structural modeling, called implicit modeling, is an alternative to these limits. It considers all data jointly, all horizons and discontinuities, and most validity conditions can be formulated mathematically as unbreakable constraints. Though computationally more demanding than explicit methods, implicit methods have become more and more promising together with the development in computational hardware. Nowadays, its popularity has evolved to the point where explicit modeling tends to be forgotten even in cases where it is much more efficient and flexible. A study from Collon *et al.* (2015) shows some applied advantages and limits to both approaches and how there combination can handle a complex real case study.

This thesis deals with implicit algorithms for structural modeling, so explicit methods are not further discussed.

1.2 Implicit structural modeling

The concept of implicit modeling is not new (Blinn, 1982), and even more so when considering potential and contouring techniques (Bhattacharyya, 1969). Implicit techniques have mainly been used in computer graphics to represent shapes in animation, simulation, and visualization (Gomes *et al.*, 2009). It has also been applied in different geoscience applications, such as contour mapping (Wessel & Bercovici, 1998), geobody representations (Ledez, 2003; Frank *et al.*, 2007) and structural modeling (Lajaunie *et al.*, 1997; Frank *et al.*, 2007; Caumon *et al.*, 2013). This section introduces the concept of implicit modeling applied to structural modeling.

1.2.1 Definition of implicit modeling

1.2.1.1 Scalar field and iso-surfaces

Implicit modeling consists in constructing a scalar field u defined over the entire volume of interest as

$$u : \left| \begin{array}{ccc} \mathbb{R}^3 & \rightarrow & \mathbb{R} \\ \boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} & \rightarrow & u(\boldsymbol{x}) \end{array} \right. . \tag{1.1}$$

This scalar function is called implicit function and is assumed not constant over the domain. A surface in 3D is represented by an iso-value of this function and is called iso-surface. It is defined by

$$\forall \boldsymbol{x} \in \mathbb{R}^3, \quad u(\boldsymbol{x}) = \alpha, \tag{1.2}$$

with $\alpha \in \mathbb{R}$ a constant.

This concept also applies in 2D where iso-values represent curves, and in 1D where they represent points. Figure 1.4 illustrates the concept of iso-values representing modeling objects. In the following, *iso-surface* will be used as a generic term for all modeling objects created with an iso-value and *volume* may designate a volumetric space in 3D, an area in 2D and a section in 1D.

1.2.1.2 Sets of sub-parallel iso-surfaces

One implicit function can represent an infinity of iso-surfaces. If an implicit function is strictly continuous, its iso-surfaces cannot intersect as they have a sub-parallel geometric relationship (Figure 1.5(a)). Two sets of sub-parallel iso-surfaces may be put in contact to create intersecting surfaces. This can be managed with several continuous implicit functions and Boolean operations (Figures 1.5(b) and (c)), or with a unique implicit function which is continuous by parts (Figure 1.5(d)). The number of functions varies depending on the number of expected surfaces, the surfaces mutual connectivity, and the requirements on each function.

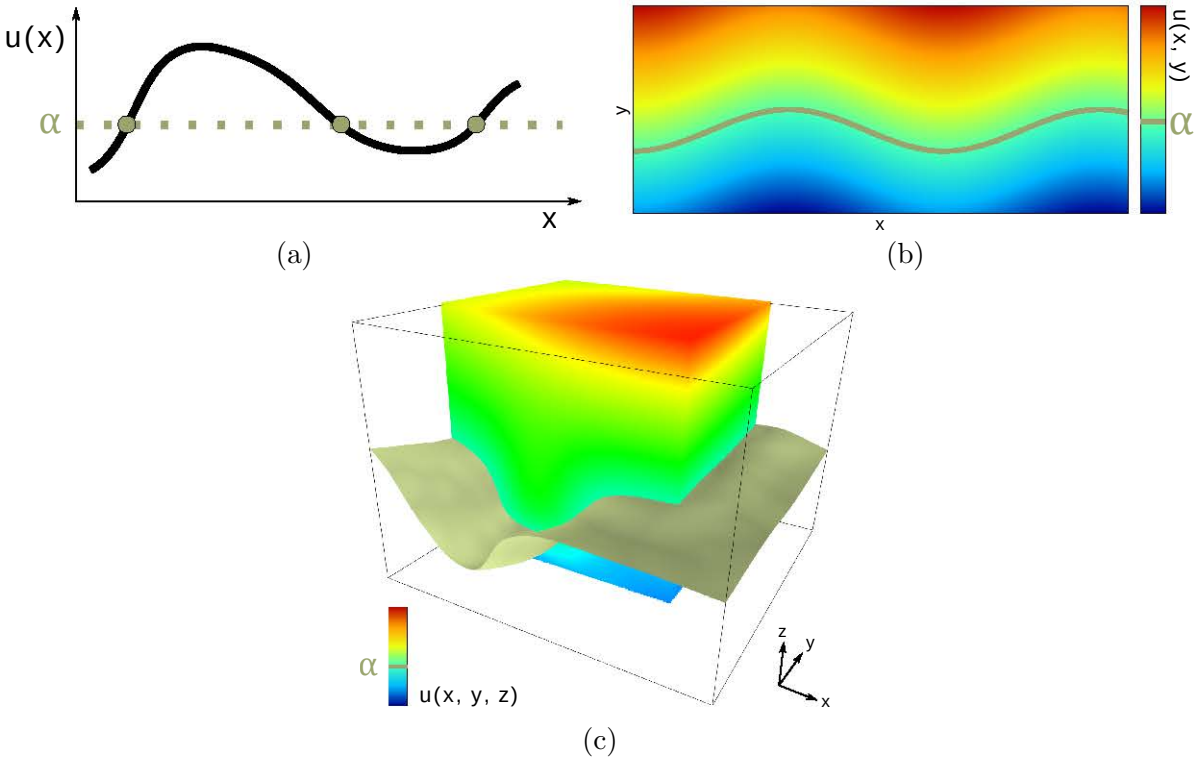


Figure 1.4: *Concept of implicit function and iso-surface in (a) One dimension, (b) Two dimensions, and (c) Three dimensions.*

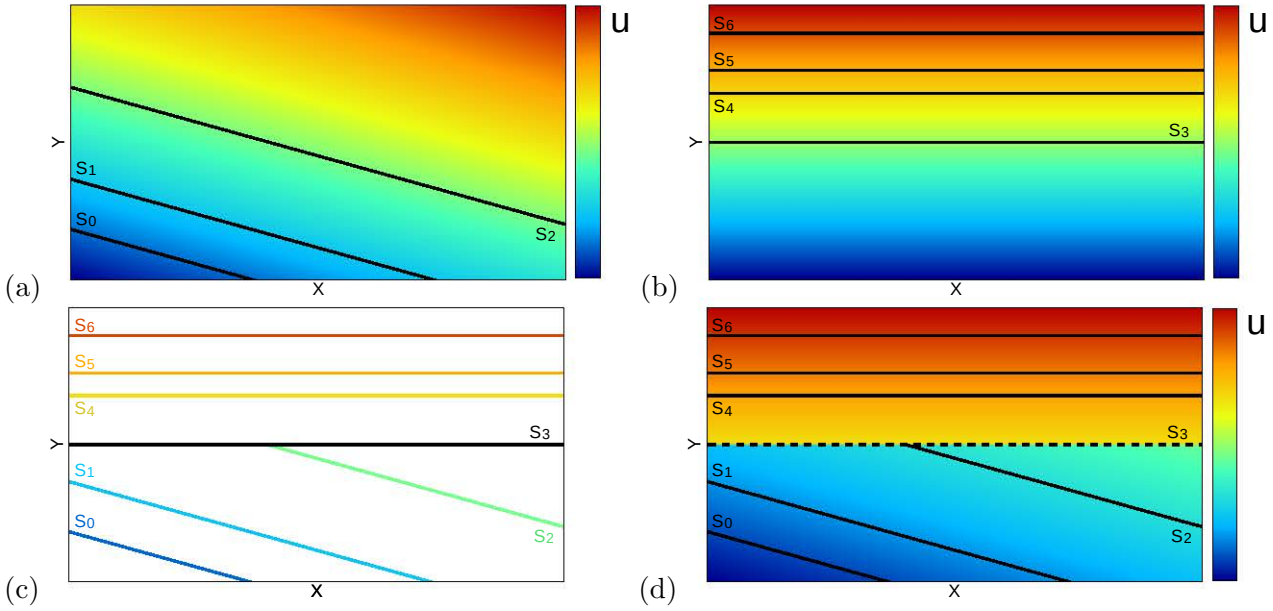


Figure 1.5: *How to represent intersecting and non-intersecting surfaces with implicit modeling. (a) An infinite set of sub-parallel iso-surfaces can be extracted from a continuous implicit function, (b) A second sub-parallel set extracted from a different implicit function, (c) Boolean operations performed on (a) and (b) can create intersecting surfaces, and (d) The same intersecting sets of iso-surfaces can be extracted from a discontinuous implicit function.*

1.2.1.3 Representing a set of points

Modeling a set of points by an iso-surface therefore calls for constructing a function on a volumetric surrounding space in which all the points hold the same value. The function is supposed to honor both

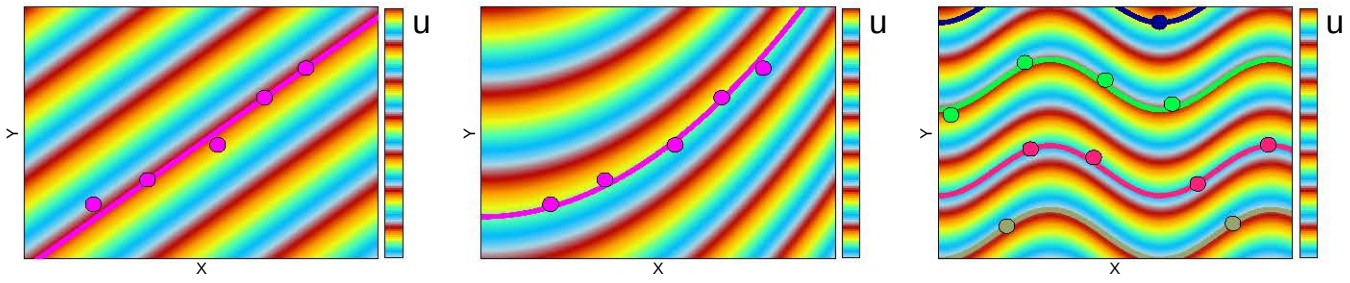


Figure 1.6: *Representing one or several sets of points with an implicit function. (a) Approaching a set of points with a linear function and their average iso-value, (b) The same with a quadratic function, (c) Several sets of points represented by several iso-surfaces.*

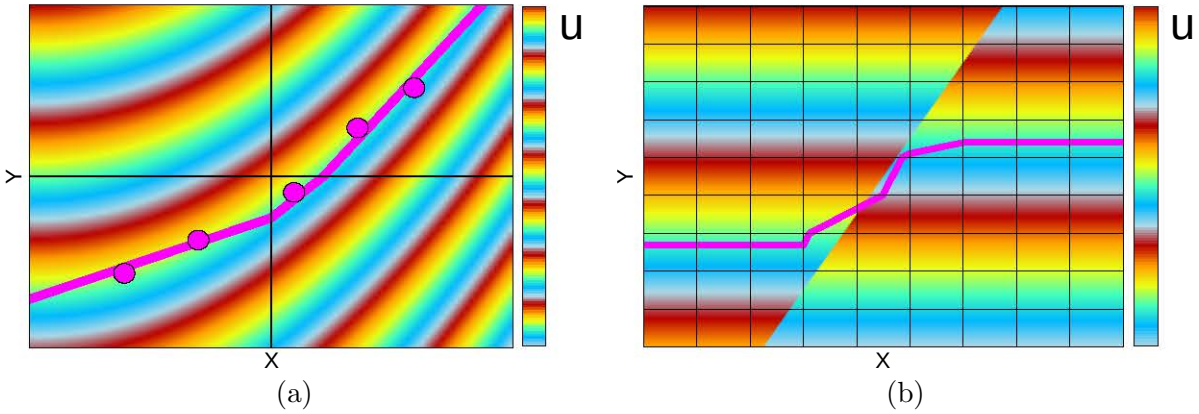


Figure 1.7: *Approximation created during the marching square extraction stage. (a) An iso-surface of a set of points in a quadratic implicit function is approximated by a polygonal curve (extraction grid:* (3×3)*). (b) An iso-surface approximated on a discontinuous implicit function, with a visible stair-step effect (extraction grid:* (10×10)*).*

the position of the points, and their expected iso-value. Figures 1.6(a) and 1.6(b) show two different implicit functions more or less adapted to represent a given set of points. Several sets of points can be modeled by different iso-values of a same implicit function (Figure 1.6(c)).

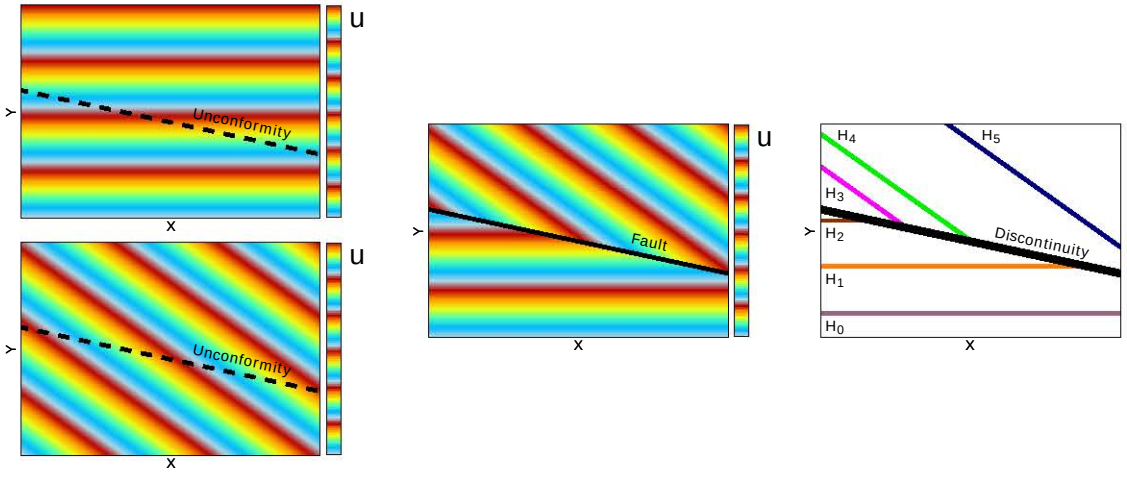
1.2.1.4 Extraction of iso-surfaces

Creating an implicit function is not enough to obtain the numerical objects of the corresponding iso-surfaces. These objects are extracted from the function, meaning that a parametric or polygonal surface following the iso-value is created. The best-known extraction techniques are the marching element-based techniques, such as marching cubes (Lorensen & Cline, 1987) which have been derived into marching tetrahedra, squares and triangles. Such techniques assume an evaluation of the implicit function on a mesh, which approximates the continuity of the function at the mesh's resolution (Figure 1.7(a)). Also, they do not consider inherently the discontinuity surfaces and extract continuous surfaces with artifacts (Figure 1.7(b)). In this case, the extracted models do not honor the validity conditions illustrated in Figure 1.3(b). Projections and truncations can be considered to create a sealed model from these results. More sophisticated techniques of extraction can be found in the review works of (Gomes *et al.*, 2009) and (De Araújo *et al.*, 2015).

1.2.2 Application to structural modeling

1.2.2.1 Stratigraphic function

In implicit structural modeling, the stratigraphic layers are represented by a scalar field. The horizons belonging to the same stratigraphic sequence are assumed sub-parallel and are represented by different



(a) (b) (c)

Figure 1.8: *Equivalence between infinite fault and stratigraphic unconformity. (a) Two stratigraphic functions representing two sequences before Boolean operations, (c) Infinite fault cutting a discontinuous stratigraphic function, (d) Common extracted model.*

iso-values of one continuous implicit function (e.g., Figures 1.5(a) and (b)) (Chilès *et al.*, 2004). Evidences of horizons are sorted by horizons and they constrain all together the creation of the stratigraphic function (e.g., Figure 1.6(c)). With this setting, it is impossible to create two intersecting horizons, or laterally vanishing horizons (i.e., validity conditions, Section 1.1.5).

1.2.2.2 Stratigraphic unconformities as contact surfaces between functions

An unconformity represents the surface of contact between several sequences. Most implicit techniques create a different stratigraphic function per sequence and then use Boolean operations to create a common model (Calcagno *et al.*, 2008). However, this process calls for a truncation of the stratigraphic functions (or of the extracted iso-surfaces) following the unconformity. A representation of the unconformity is therefore needed; for instance, with a polygonal surface or an iso-surface of one of the computed implicit functions (e.g., S_3 in Figure 1.5(a), (b) and (c)).

1.2.2.3 Faults as stratigraphic jumps

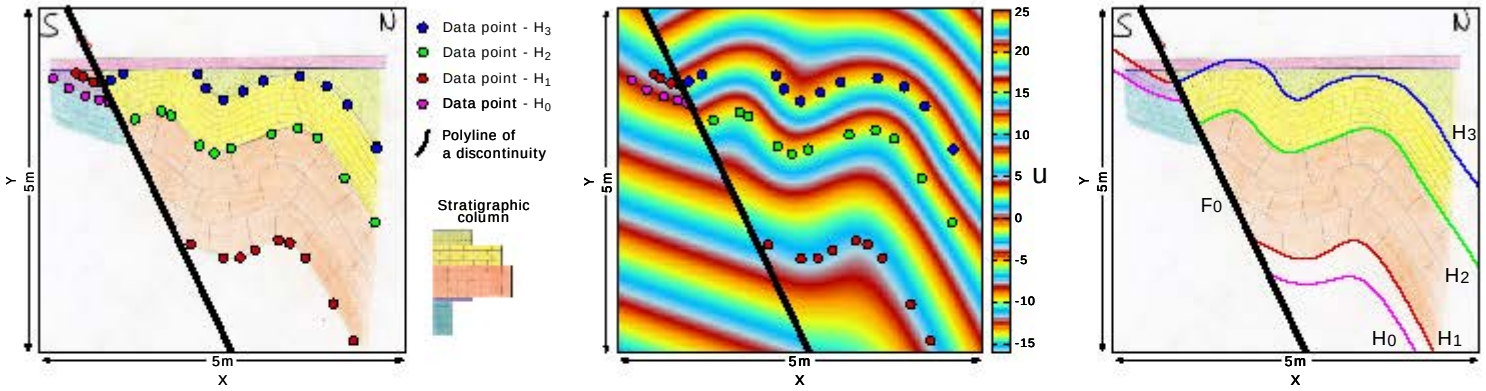
Faults displace layers of one or several stratigraphic sequences. Therefore, faults are generally represented by a discontinuous jump in the stratigraphic functions (e.g., Figures 1.5(d) and 1.7(b)) (Chilès *et al.*, 2004; Frank *et al.*, 2007). By following this jump, the fault surface could be extracted. In practice, creating a jump in an implicit function and constraining it to pass through the geometry of a fault already calls for a representation of this fault. Polygonal surfaces are generally created with explicit methods, and are then given as inputs to implicit modeling (Chilès *et al.*, 2004; Frank *et al.*, 2007). Implicit iso-surfaces are also possible: Tertois (2007) and Cherpeau *et al.* (2010) represent fault networks with several implicit functions and Boolean operations.

1.2.2.4 Intrusive geobodies as stratigraphic cuts

In structural modeling, an intrusive geobody is represented as a separate unit from the stratigraphy. Stratigraphic functions are first created with no concern of the geobodies. These functions (or their extracted iso-surfaces) are then truncated by a representation of the geobodies envelopes. Just like faults, these surfaces can be defined explicitly or implicitly.

1.2.2.5 Discussion on structural discontinuities

The implicit approach applied to structural modeling is mainly restricted to constructing the stratigraphic function. First, structural discontinuities are modeled, for instance, with explicit approaches,



(a) (b) (c)

Figure 1.9: *Implicit modeling workflow to represent geological structures. (a) A handmade cross section from field observations manually interpreted, (b) Stratigraphic function, (c) Extracted sealed model. The same workflow can be applied to seismic and/or borehole images.*

and then, the resulting surfaces are fed to the implicit algorithm to model horizons. Though this workflow seems relying more on explicit than implicit, it creates a setting where most of the validity conditions defined in Section 1.1.5 are naturally honored. In addition, though not entirely avoided, the limits of explicit modeling (Section 1.1.6) are mainly restricted to the discontinuities. As the validity conditions mostly concern horizons, it reduces the number of truncations, projections and other constraints on explicit objects.

Other settings for implicit structural modeling are possible. For instance, the assumption of horizontal deposition of stratigraphic layers is not always pertinent. Onlap and downlap structures are sequent to the unconformity surface, and therefore cannot be directly represented in the presented scheme. The surface of unconformity is thus created explicitly or implicitly and given as input to the creation of the stratigraphic function. In this case, a stratigraphic unconformity is equivalent to an infinite fault: they both are contact surfaces between two sets of sub-parallel surfaces (Figure 1.8). Therefore, a cut between two stratigraphic functions and a unique stratigraphic function with a discontinuous jump crossing all the model are defining the same objects.

A summary of the structural implicit modeling workflow on a synthetic cross section is illustrated in Figure 1.9.

1.2.3 Level-set methods in structural modeling

For most authors, the terms level-set and implicit are equivalent as a level-set function is defined by Equation (1.1). A level-set surface is then equivalent to an iso-surface and defined by Equation (1.2).

A level-set function can also be related to time (Osher & Sethian, 1988). Two classes of level-set methods exist with this definition:

- Static methods where a unique level-set function is created, and each iso-surface is represented by different times t:
$$u(\boldsymbol{x}) = t. \tag{1.3}$$
- Dynamic methods where the level-set function evolves in time, and multiple iso-surfaces can be represented by the same constant α at different times t:
$$u(\boldsymbol{x}, t) = \alpha. \tag{1.4}$$

Figure 1.10 illustrates the difference between these two approaches.

In geomodeling, both static and dynamic methods have been studied to represent the surfaces of geobody envelopes (Ledez, 2003). In structural modeling, only a static case application is known by the author.

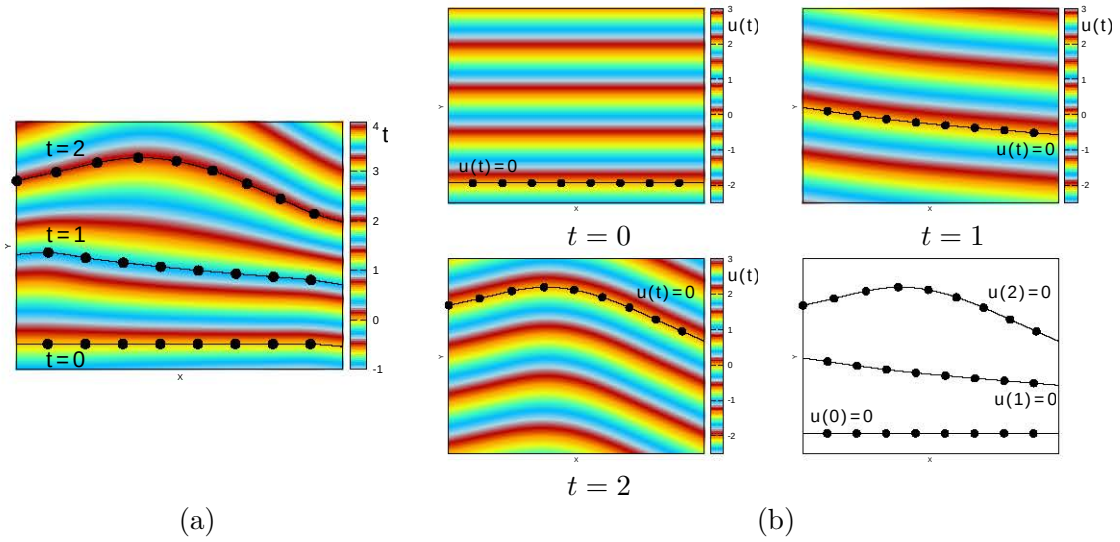


Figure 1.10: *Difference between static and dynamic approaches in level-set methods. (a) Static approach: one distance function dependent on time, (b) Dynamic approach: a distance function per time lap.*

Hjelle & Petersen (2011) model horizons by a propagating front with the static version of the Hamilton-Jacobi equation. In this formalism, a front is an iso-surface of the time t, respecting Equation (1.3). This front displaces from a time t_1 to a time t_2 following a direction $\boldsymbol{N}$ which is spatially dependent. By influencing $\boldsymbol{N}$ in space, Hjelle & Petersen (2011) can produce any fold type in Ramsay's classification (Ramsay, 1967). For instance, if $\boldsymbol{N}$ is the normal vector to the front, parallel folds are created (i.e., Class 1B folds), and if $\boldsymbol{N}$ is along the vertical, similar folds are created (i.e., Class 2 folds). These relations are summarized in Hjelle *et al.* (2013).

This approach poses the structural modeling problem as a partial differential equation, the Hamilton-Jacobi equation, which is robust mathematically. In addition, the flexibility in the definition of direction of propagation $\boldsymbol{N}$ offers the possibility to represent features in between the Ramsay's classification. However, the determination of the directions to use based on local data analysis has not been studied yet. Also, solving the non-linear Hamilton-Jacobi equation is challenging, especially on big models. Level-set methods will not be further studied in this thesis as one of the goals is to create numerical methods with at least the perspective to be adapted for big models.

1.3 Numerical methods for implicit structural modeling

There exist two main classes of implicit techniques for structural modeling: the ones issued from the Discrete Smooth Interpolation (Mallet, 1992) and the ones issued from the Potential Field Method (Lajaunie *et al.*, 1997). Both are defined in a general manner, so most of the existing implicit techniques in structural modeling can be classified in either of the two. There still exist some exceptions, such as the front propagation algorithm presented in Section 1.2.3.

This section describes in detail both methods separately and then discusses the available strategies for improvement.

1.3.1 Discrete Smooth Interpolation (DSI)

1.3.1.1 Origin and applications

Its precursor idea was initiated in Mallet (1988), it then took the name of Discrete Smooth Interpolation in Mallet (1989), and was further generalized in Mallet (1992, 1997, 2002). DSI was first applied to explicit modeling making triangulated surfaces fit the data points by smoothing. Later, it has been successfully applied to 3D implicit modeling of geological structures (Frank *et al.*, 2007).

This technology has mainly been supported and developed by and for the oil and gas industry which allowed the creation of different software such as SKUA-Gocad (Paradigm, 2019) and Volume

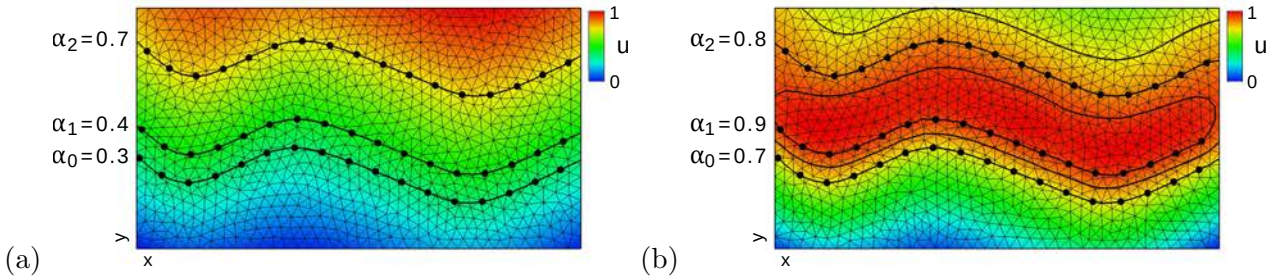


Figure 1.11: *Influence of the constrained iso-values α_H on the results. (a) Iso-values computed with a preprocessing, (b) Iso-values changed manually to constrain a nonmonotonic stratigraphic function.*

Based Modeling (VBM) for Petrel (Souche *et al.*, 2014; Schlumberger, 2018). There are also some applications of SKUA-Gocad in the mining industry.

DSI specializes in modeling geological structures as horizons and major faults using data interpreted from seismic and borehole images. Its main particularity is to assume that the smoothest interpretation of geological structures honoring the data at best is to be preferred.

Access to VBM source code in 2D and 3D was granted by Schlumberger for this thesis. The examples to illustrate DSI were produced with this software. Only 2D examples are discussed for several reasons: (i) they often provide better or simpler illustrations for a theoretical concept than 3D examples, (ii) the 2D code of VBM was simple enough to erase all additional heuristics of the code, (iii) only a 2D version of PFM was created during the thesis, so it eases the discussions between the two methods.

1.3.1.2 Constraints on numerical data

Many numerical data (Section 1.1.4) can be considered in DSI and an exhaustive list of data constraints can be found in Mallet (2014) (Section 14.2). In most applications, a DSI algorithm accepts data points and can use, but does not need, other types of numerical objects (Mallet, 2002; Frank *et al.*, 2007; Souche *et al.*, 2014).

Each horizon H is respectively represented by an iso-value α_H (Section 1.2.2.1). The data points are sorted by horizons and associated to the corresponding iso-value. Let $\boldsymbol{D}(H)$ be the set of data points belonging to a given horizon H. The implicit function u is constrained to hold the value α_H at each data point positon $\boldsymbol{p}$ as

$$\forall H,\ \forall \boldsymbol{p} \in \boldsymbol{D}(H), \quad u(\boldsymbol{p}) = \alpha_H. \tag{1.5}$$

This constraint assumes that the iso-value α_H of each horizon H is known at the beginning. A difficulty is that the geologist has *a priori* no possibility to know the expected α_H values. However, these values have a strong impact on the solution quality (Figure 1.11). Expected iso-values should be chosen: (i) monotonous following the relative geological times, and (ii) adapted to the average thicknesses of the layers. The closer two horizons are one to another, the smaller should be the gap in their associated iso-values. Some preprocessing steps on the data points exist to determine such values and were successfully applied to real case studies (Collon *et al.*, 2015).

1.3.1.3 Mesh-based interpolation

DSI discretizes the implicit function u on a volumetric mesh. It can be defined on any type of mesh element as long as Finite Element Method (FEM) shape functions can be constructed on it (Section 2.1.3.2). Let $\boldsymbol{N}$ be the set of mesh nodes. The implicit function u is defined as

$$\forall \boldsymbol{x} \in \Omega, \quad u(\boldsymbol{x}) = \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}} \Phi_{\overline{\boldsymbol{p}}}^{FEM}(\boldsymbol{x})\ u_{\overline{\boldsymbol{p}}} = \boldsymbol{\Phi}(\boldsymbol{x})^T \cdot \boldsymbol{U}, \tag{1.6}$$

with $\Phi_{\overline{\boldsymbol{p}}}^{FEM}$ the FEM shape function associated to the mesh node $\overline{\boldsymbol{p}}$, and $u_{\overline{\boldsymbol{p}}}$ the corresponding coefficient, which is also an unknown of the problem. By definition, $\Phi_{\overline{\boldsymbol{p}}}^{FEM}$ is null for any position $\boldsymbol{x}$ away from the elements constructed with $\overline{\boldsymbol{p}}$ as a top.

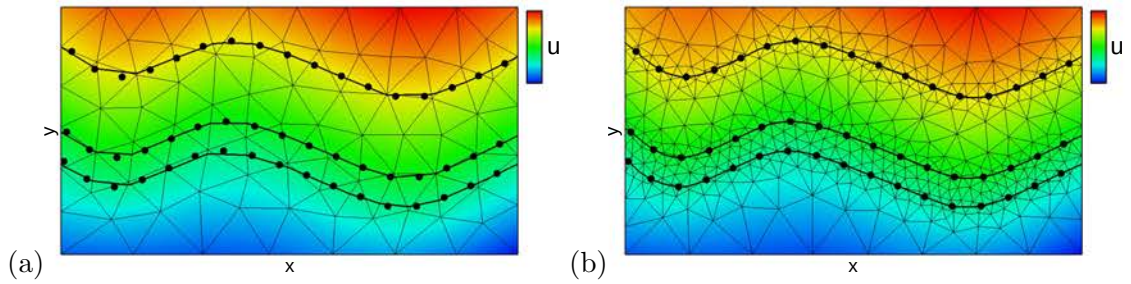


Figure 1.12: *Example of different mesh resolutions used on the same numerical data. (a) Homogeneously coarser resolution than in Figure 1.11(a), and (b) Heterogeneous resolution. The extracted iso-surfaces in (a) do not pass exactly through all the data points because of the linear interpolation in coarse triangles.*

The mesh discretization is usually performed on simplexes. In the explicit variant of DSI, triangle meshes are used for 3D surface smoothing (Mallet, 1988, 2002; Levy & Mallet, 1999) for their flexibility to fit complex geometries. In the implicit variant, they are used in 2D, and tetrahedra are generally used in 3D (Frank *et al.*, 2007; Souche *et al.*, 2014). In both cases, these mesh elements support linear interpolation functions (for equations, see Dhatt & Touzot (1981): Example 1.11, Section 2.3.2 and Section 2.5.2). The coefficients $u_{\overline{\boldsymbol{p}}}$ of the function u are then linearly interpolated within the mesh elements. Other meshes composed of well known elements can be used provided that they can support a basis functions, for instance: quadrilaterals in 2D and hexahedra in 3D. Also, different sizes of element can be used in a same mesh, which may involve different degrees of approximations of the geological structures (Figure 1.12).

1.3.1.4 Smooth regularization

Data point constraints can be written on N_D data points giving N_D equations for N_N unknowns (Equation (1.5)). It is possible to have $N_D < N_N$, and therefore deal with an underdetermined system. In addition, because of the data irregularity (Section 1.1.4.5) and the locally defined shape functions Φ^{FEM}, some nodes may not be involved in any data constraint. DSI solves both issues by adding a roughness factor to the modeling problem (Mallet, 1992).

The assumption made is that the smoothest possible solution is preferred for geological structures. There is no limit to the way this smoothness can be expressed in the algorithm; DSI is in this sense generic. Examples of applied formulations of this factor are the Laplacian operator (Levy & Mallet, 1999) (for explicit surfaces only), the constant gradient constraint (Frank *et al.*, 2007), and the smooth gradient constraint (Souche *et al.*, 2014). The constant gradient is the best-known constraint; it corresponds to

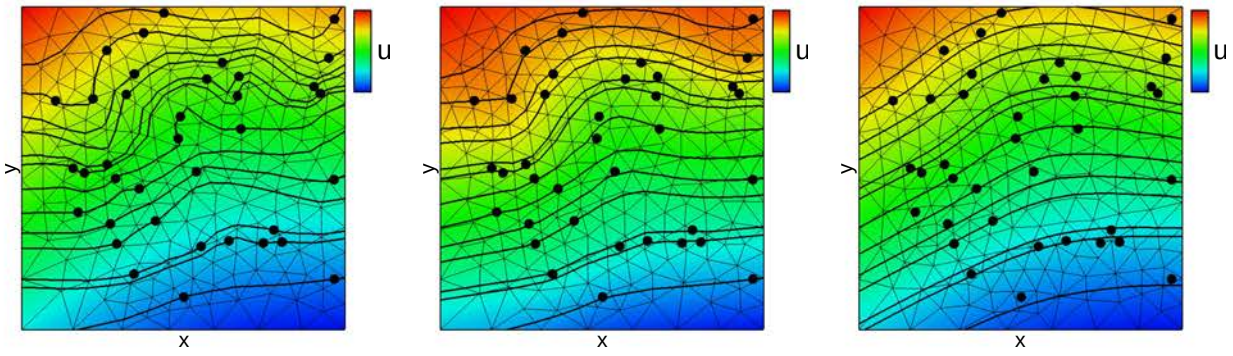
$$\forall e \in \boldsymbol{E}, \forall e' \in \boldsymbol{E}(e), \quad \nabla u_{|e} - \nabla u_{|e'} = \boldsymbol{0} \Leftrightarrow \left(\nabla \boldsymbol{\Phi}_{|e} - \nabla \boldsymbol{\Phi}_{|e'}\right)^T \cdot \boldsymbol{U} = \boldsymbol{0}, \tag{1.7}$$

with $\nabla u_{|e}$ the gradient of the function u evaluated in the element e as

$$\forall e \in \boldsymbol{E}, \quad \nabla u_{|e} = \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}} \nabla \Phi^{FEM}_{\overline{\boldsymbol{p}}}(\boldsymbol{x})_{|e}\ u_{\overline{\boldsymbol{p}}} = \nabla \boldsymbol{\Phi}^T_{|e} \cdot \boldsymbol{U}, \tag{1.8}$$

and with $\boldsymbol{E}$ the set of mesh elements, and $\boldsymbol{E}(e)$ the set of adjacent elements e' of e. This constraint involves one equation per axis and per pair of adjacent elements in the mesh. The position of evaluation $\boldsymbol{x}_{|e}$ depends on the FEM shape function and the approximation made on the gradient $\nabla u_{|e}$. In the specific case of piecewise linear FEM functions (i.e., constructed on triangles in 2D and tetrahedra in 3D), the gradient is constant within the mesh elements and is thus computed directly with each element's nodes (Frank *et al.*, 2007). In the VBM software we use, the smooth gradient constraint is implemented as

$$\forall e \in \boldsymbol{E}, \quad \nabla u_{|e} - \frac{1}{\#\boldsymbol{E}(e)} \sum_{e' \in \boldsymbol{E}(e)} \nabla u_{|e'} = 0. \tag{1.9}$$



(a) (b) (c)

Figure 1.13: *Influence of the smooth factor on irregularly distributed and noisy data points. (a) A low λ_ϵ fits perfectly the data points, (b) An average λ_ϵ filters the noise, and (c) A great λ_ϵ oversmooths the structures.*

In addition to the data points constraints, DSI's smooth constraints ensure to have at least one equation per element in the mesh. This is supposed, in general, to create a system with more equations than unknowns (i.e., $\#Equations \geq \#\boldsymbol{N}$). This overdetermined system is then solved in the least squares sense.

1.3.1.5 Handling heterogeneous data

The accepted numerical data and the related constraints are described in Section 1.3.1.2. Here, the quality and the spatial distribution of numerical data are discussed (Section 1.1.4.5).

The problem posed in DSI honoring both the data constraints and the smooth regularization constraints is to be seen as a spatial regression of data points penalized by a smoothing factor. In the case of the constant gradient constraint (Equation (1.7)), this problem minimizes in the least squares sense

$$\begin{aligned} J(u) &= \frac{1}{2} \sum_{(e,e')\in \boldsymbol{E}(e,e')} \lambda_\epsilon^2 (\nabla u_{|e} - \nabla u_{|e'})^2 &&+ \frac{1}{2} \sum_H \sum_{\boldsymbol{p}\in \boldsymbol{D}(H)} \lambda_{\boldsymbol{p}}^2 \, (u(\boldsymbol{p}) - \alpha_H)^2, \\ &= \frac{1}{2} \sum_{(e,e')\in \boldsymbol{E}(e,e')} \lambda_\epsilon^2 \left(\left(\nabla \boldsymbol{\Phi}_{|e} - \nabla \boldsymbol{\Phi}_{|e'}\right)^T \cdot \boldsymbol{U} \right)^2 &&+ \frac{1}{2} \sum_H \sum_{\boldsymbol{p}\in \boldsymbol{D}(H)} \lambda_{\boldsymbol{p}}^2 \, (\boldsymbol{\Phi}(\boldsymbol{p})^T \cdot U - \alpha_H)^2, \end{aligned} \tag{1.10}$$

with $\boldsymbol{E}(e, e')$ the set of pairs of adjacent elements e and e' in the mesh, and λ_ϵ and $\lambda_{\boldsymbol{p}}$ the weights respectively associated to the smoothing constraint and each data constraint.

In this minimization problem, the greater the weighting factor, the better a constraint is honored. Figure 1.13 shows the smoothing ability of DSI when increasing the weight λ_ϵ with fixed weights $\lambda_{\boldsymbol{p}}$: a small value of λ_ϵ creates noisy horizons that perfectly fit the data (Figure 1.13(a)), an average value creates smooth horizons that filter the noise in the data (Figure 1.13(b)), and a high value oversmooths the structures (Figure 1.13(c)). Also, smoothing the structures allows to handle sparsity and irregularly sampled data (Figure 1.13) by extrapolating where data are missing and averaging clusters of data.

Finally, the data reliability problem is simplified by distinguishing two types of data (Mallet (2002)): (i) the hard data that must be honored (borehole and surface data), and (ii) the soft data that should be honored as much as possible under the smoothness criterion (seismic data). The hard data are then perfectly fitted by adding a node in the mesh at their position $\boldsymbol{p}_{hard}$ and fixing its nodal coefficient u_{hard} at the hard data value α_{hard} (Figure 1.14). This is possible because shape functions constructed on mesh elements exactly honor nodal values (i.e., Kronecker delta property, Section 2.1.3.1).

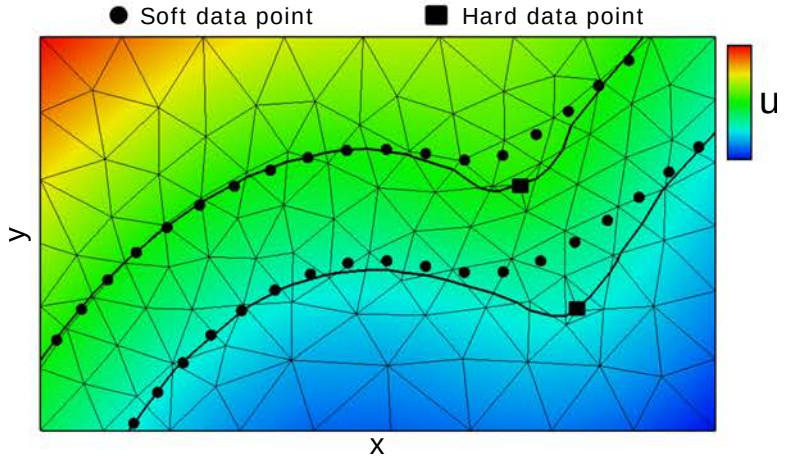


Figure 1.14: *Model created with two hard data points in VBM. Control nodes are added into the mesh and iso-surfaces pass exactly through them.*

1.3.1.6 Handling structural discontinuities

Stratigraphic unconformities and intrusive geobodies

As described in Section 1.2.2.2, stratigraphic unconformities are handled with a separate stratigraphic function per sequence. For the intrusive geobodies, the implicit function is cut by the surface envelopes as described in Section 1.2.2.4.

Faults

The mesh-based shape functions $\Phi_{\overline{\boldsymbol{p}}}^{FEM}$ are C^0 on the domain of study Ω. Introducing a discontinuous jump in the implicit function u within an element means either breaking the continuity in the related shape function or breaking the element itself. DSI uses the second solution (Figure 1.15(a)): the mesh is created conformal to the faults, meaning that an element of the mesh cannot cut through them. All nodes touching a fault are then duplicated on either side. Therefore, a fault is a boundary to the domain of study and a different unknown $u_{\overline{\boldsymbol{p}}}$ is associated to each created node $\overline{\boldsymbol{p}}$. As the unknowns can have a different value between duplicated nodes, it allows to create jumps in the implicit function u (Figure 1.15(b)).

1.3.1.7 Example of DSI system of equations

An example of DSI least squares system is given here. It is obtained from the minimization of the term J (Equation (1.10)) considering all the data constraints (Equations (1.5)) and the constant gradient constraints (Equation (1.7)). This system is written as

$$\begin{cases} \lambda_\epsilon \left(\nabla \boldsymbol{\Phi}_{|e} - \nabla \boldsymbol{\Phi}_{|e'}\right)^T & \cdot \ \boldsymbol{U} & = & \boldsymbol{0}, & \forall (e,e') \in \boldsymbol{E}(e,e') \\ \lambda_{\boldsymbol{p}} \boldsymbol{\Phi}(\boldsymbol{p}) & \cdot \ \boldsymbol{U} & = & \lambda_{\boldsymbol{p}} \alpha_H, & \forall H, \ \forall \boldsymbol{p} \in \boldsymbol{D}(H) \end{cases} . \tag{1.11}$$

1.3.1.8 Visualizing the results

An implicit function created with DSI can be directly visualized by printing each obtained coefficient value $u_{\overline{\boldsymbol{p}}}$ on its associated node. Therefore, no further evaluations of the implicit function are necessary after solving system (1.11). Further evaluations are only necessary to obtain a finer resolution of the function, which is rarely done as the mesh-based shape functions have a low degree of continuity (i.e., only piecewise linear for triangulated meshes). When a finer resolution is expected, either the implicit algorithm is relaunched entirely with a finer mesh or refinement algorithms are employed, involving remeshing and specific constraints. Marching element-based algorithms are then employed to extract the iso-surfaces.

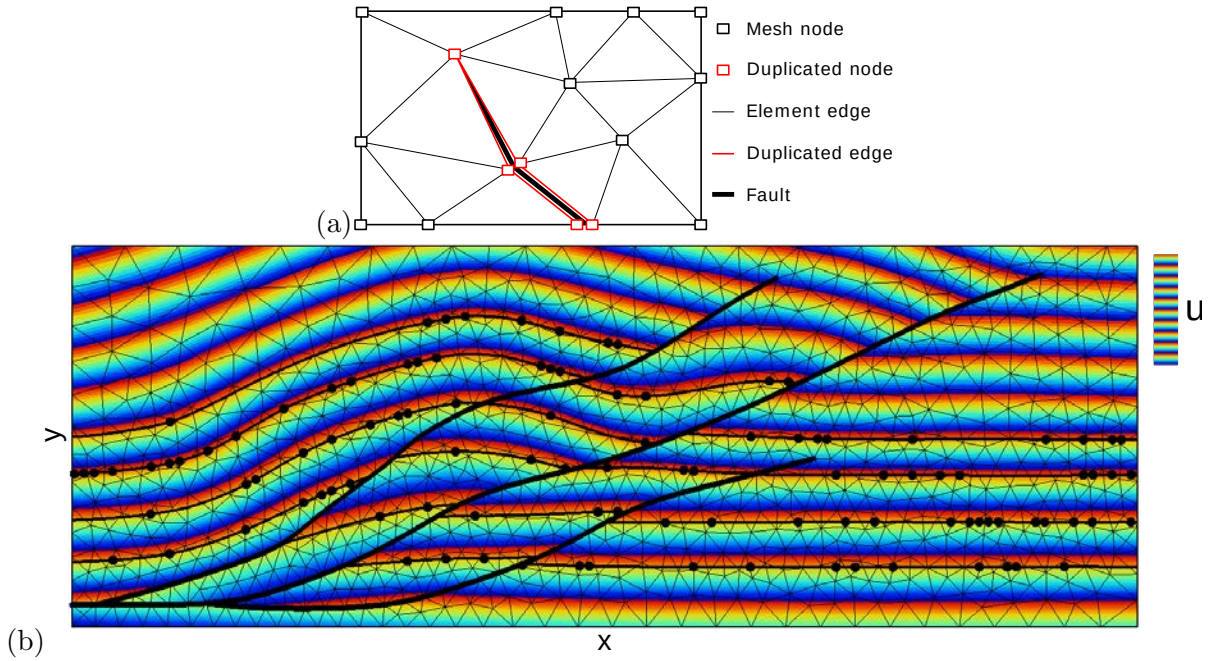


Figure 1.15: *Handling structural faults with DSI. (a) Illustration of the concept of conformal mesh to the faults with duplication of the nodes on either side of the fault, and (b) Faulted model computed with VBM.*

1.3.1.9 Advantages and limits

A local method

DSI performs the interpolation with local shape functions $\Phi_{\boldsymbol{p}}^{FEM}$ constructed on the mesh elements. This gives a sparse system of equations (i.e., most of the coefficients in system (1.11) are null). Such a system can be solved efficiently even with millions of data points and elements.

These locally supported functions also allow to adapt the interpolation to the structures and their dimensions. Meshes with varying resolution (Figure 1.12(b)) can hence adapt the relative size of the mesh elements to the needed continuity to represent a given feature. Small structures are locally assessed with small elements, and large homogeneous structures are assessed with coarser elements. As the dimensions of system (1.11) are also dependent on the number of nodes $\#\boldsymbol{N}$, adapting the mesh resolution to the expected continuity can optimize the system complexity. The mesh being created conformal to the faults, the extraction of a sealed model with marching element type methods is straightforward.

Although the interpolation is locally performed, the mesh and the roughness factor ensure that a solution is obtained everywhere in the studied domain Ω. It also enables to control the model features away from the data by changing the roughness factor in space (Mallet, 1997; Caumon *et al.*, 2013).

A mesh dependent method

Using a mesh to perform the interpolation produces results dependent on the mesh. If the mesh is too coarse, the implicit function smooths the local features. If the mesh elements are not adapted to the constraints, they can completely alter the implicit function (e.g., antagonist gradient constraints in two adjacent triangles in Laurent (2016), Figure 2). Finally, if some mesh elements are of bad quality, such as triangles with small angles, the constructed shape functions may themselves be altered.

The additional difficulty in geology is that the mesh must be conformal to the faults (i.e., the triangles of the discontinuities should be faces of the mesh elements, Section 1.3.1.6). Depending on the complexity of the fault network, the construction of the mesh can be challenging (Pellerin *et al.*, 2014; Karimi-Fard & Durlofsky, 2016). For instance, mesh algorithms cannot always ensure to create a mesh in which the gradient of the implicit function can be correctly computed (Shewchuk,

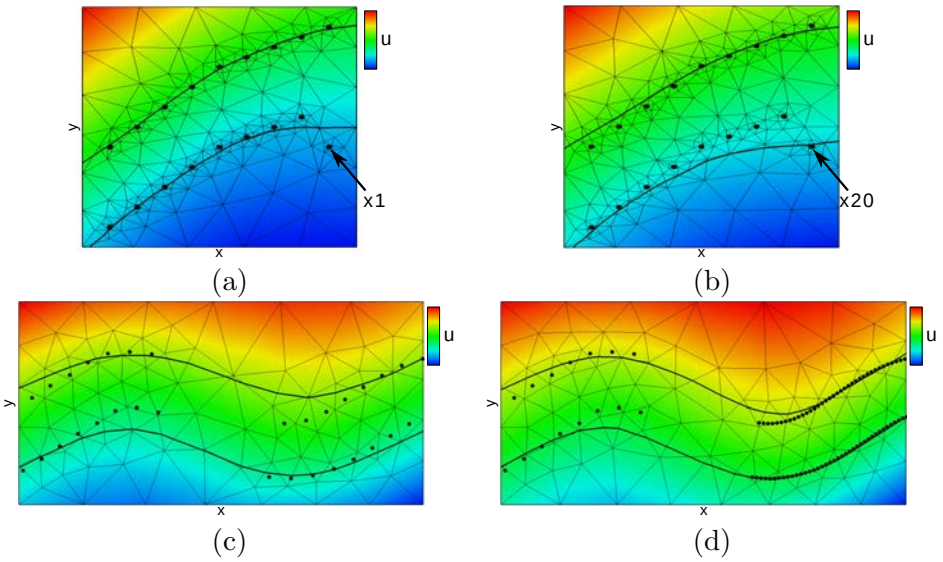


Figure 1.16: *Influence of the number of data equations with the same weight $\lambda_{\boldsymbol{p}}$ on the modeling results. (a) A point is isolated far-off the bottom horizon and is poorly honored, (b) The point is added twenty times to the model and is over-fitted as compared to the other points, (c) A poorly sampled fold is smoothed away by a high smoothing factor, and (d) The right side of the fold is densely sampled and better honored with the same smoothing factor.*

2002), especially when dealing with fault networks of complex geometries and intersections. It may also be computationally expensive and may require significant user interactions.

Creating a mesh in addition to the numerical data to support the interpolation also involves an additional cost of memory. For large scale models (e.g., basin scale) even a mesh with an adaptive resolution may contain a huge number (e.g., billions) of elements. The related memory challenges may represent a limit to the scalability of the method.

A non-physical weighting system

None of the weights λ_ϵ and $\lambda_{\boldsymbol{p}}$ have a physical meaning. Therefore, they cannot be simply associated to data geometric errors and several models are generally created to find adapted values to a given data set.

Every single constraint in the created least squares system (1.11) has an influence on the results. If a data point is added multiple times in the model, its constraint is repeated as many times in the system and overexpressed as compared to the other constraints (i.e., the point is over-fitted, Figure 1.16(a)). This phenomenon can also be applied to clustered data (Figures 1.16(b) and (c)) where densely sampled regions can over-express redundant parts of the structures. This is not a real limitation as different picking strategies can use this behavior to better assess some features.

The same concept also affects the smooth regularization constraints. The created models are expected to converge to a unique model for a same set of parameters when refining infinitely the resolution of the mesh. In DSI's case, this convergence cannot be ensured without an additional normalization on the weight λ_ϵ as refining the mesh increases the number of weighted equations in the least squares system.

1.3.2 Potential Field Method (PFM)

1.3.2.1 Origin and applications

The Potential Field Method (PFM) was first invented with a geostatistical approach. Lajaunie *et al.* (1997) defined the implicit function as a potential field and applied it to 2D geological modeling. Chilès *et al.* (2004) further developed the idea considering structural discontinuities with an application to 3D geological modeling. Another formulation, closer to numerical methods such as meshless techniques, has been proposed in parallel (Cowan *et al.*, 2002, 2003).

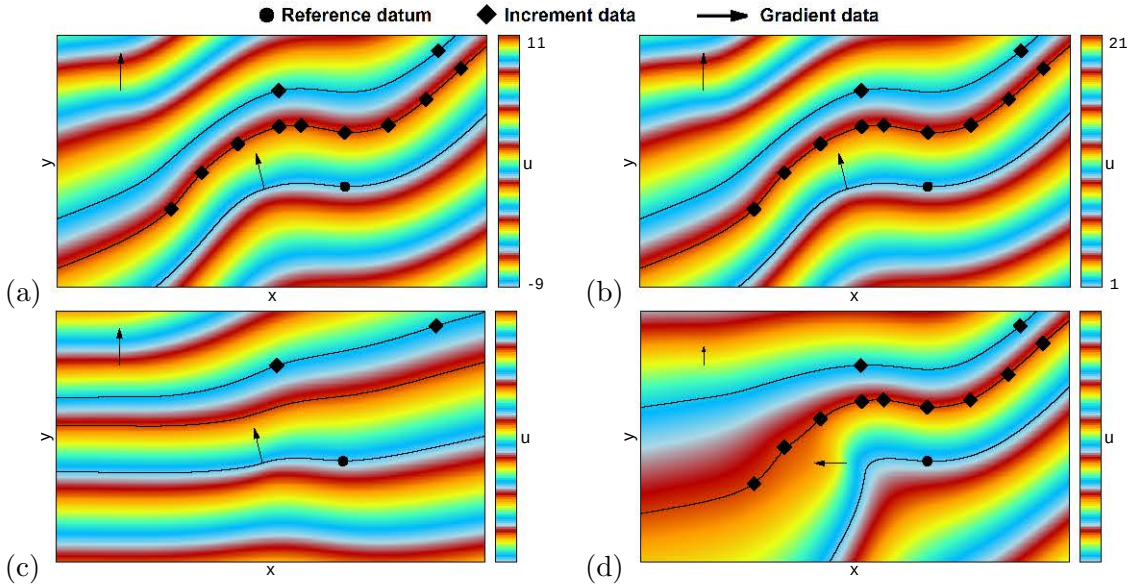


Figure 1.17: *Influence of each presented data constraint on the modeling results. (a) The reference model, (b) The value α_0 of the reference data point is changed from 0 to 10, (c) The increment data points capturing the fold are missing, (d) The gradient data are modified (norm and direction).*

PFM has mainly been supported and developed by and for the mining industry to model both stratigraphic structures and ore concentration contouring. Some popular software in this area thus use the concept of PFM, such as Geomodeller (Calcagno *et al.*, 2008; Intrepid-Geophysics, 2018), LeapFrog (Cowan *et al.*, 2002; ARANZ Geo, 2019) and Minestis (Geovariances, 2019). The geostatistical formulation (ubiquitous in mining geosciences) of this method is the governing formulation. Geomodeling in the mining industry has allowed many applications (Maxelon, 2004; Knight *et al.*, 2007; Alcaraz *et al.*, 2011; Husson, 2013; Vollgger *et al.*, 2015), but the details on practical improvements of the algorithm for real case data are not always published. Recently, some authors shared an open source prototype of PFM called GemPy (de la Varga *et al.*, 2019) with a complete access to the Python code and an active support.

As property software do not disclose their algorithms, we developed a code of PFM in 2D to observe the relation between theory and practice. This code is a direct implementation of the theory from Chilès *et al.* (2004), Calcagno *et al.* (2008) and Hillier *et al.* (2014). All the results and illustrations of this section have been produced with this code. GemPy (de la Varga *et al.*, 2019) could have been used, but the study of PFM was one of the cornerstones of this thesis and thus took place before its first release. In addition, some basic features such as finite faults are not handled yet by GemPy, which is impractical for further discussions.

1.3.2.2 Numerical constraints

Two types of numerical objects are usually considered in PFM: points and vectors. Data points represent evidences of horizons, so they belong to an interface between two layers. Data vectors give information on the dip and azimuth of the layers and they can be sampled anywhere in the volume. The related constraints come from Lajaunie *et al.* (1997) who created the concept of PFM for maps, later implemented in Chilès *et al.* (2004) to obtain the structures at depth.

The constraints are classified into three different categories. Note that the tangent constraints are not presented because they are not necessary to create a working version of PFM. Figure 1.17 relates a few influences of these constraints on the modeling results.

- **Reference data point.** A reference position $\boldsymbol{p}_0$ together with its implicit value α_0 are arbitrarily chosen. The constant α_0 has no influence on the modeled structures: it influences the values of the implicit function u, not its differential evolution in space (Figure 1.17(b)). The constraint is

written as

$$u(\boldsymbol{p}_0) = \alpha_0. \tag{1.12}$$

- **Increment data points.** Each horizon H has an unknown iso-value α_H. The implicit function u evaluated at two data point positions $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ belonging to a same horizon H should have the same unknown value (Figure 1.17(c)). Hence, it is assumed that each horizon H has at least two data points. The constraint corresponds to

$$\forall H,\ \forall (\boldsymbol{p}_1, \boldsymbol{p}_2) \in \boldsymbol{D}(H),\quad u(\boldsymbol{p}_1) - u(\boldsymbol{p}_2) = 0. \tag{1.13}$$

  In practice, a reference point $\boldsymbol{p}_H$ is chosen for each horizon H, and only the independent pairs with this reference point are written (i.e., $\forall (\boldsymbol{p}, \boldsymbol{p}_H) \in \boldsymbol{D}(H)$). These pairs are called increment pairs in the following.

- **Gradient data.** Let $\boldsymbol{p}$ be the tail point of a vector $\boldsymbol{g}$ corresponding to a gradient datum. The tangential plane of the iso-surface passing by $\boldsymbol{p}$ is then assumed perpendicular to the direction of $\boldsymbol{g}$. Also, the variations of the function u should honor the gradient datum's norm $||\boldsymbol{g}||$. In other terms, the gradient of the implicit function ∇u at the position $\boldsymbol{p}$ is constrained to be equal to the gradient datum $\boldsymbol{g}$:

$$\forall \boldsymbol{p} \in \boldsymbol{D}(G),\ \nabla u(\boldsymbol{p}) = \boldsymbol{g}, \tag{1.14}$$

  with $\boldsymbol{D}(G)$ the set of tail points. In practice, the vector $\boldsymbol{g}$ is normalized to reduce the variations of the gradient in space (Figure 1.17(d)). This constraint is also known as structural constraint in the literature (Chilès *et al.*, 2004; Hillier *et al.*, 2014).

1.3.2.3 Geostatistical formulation

The geostatistical approach of PFM uses all the data constraints to define the implicit function u (Lajaunie *et al.*, 1997). Chilès *et al.* (2004) give a simplified version of the definition of u:

$$u(\boldsymbol{x}) - u(\boldsymbol{p}_0) = \sum_H \sum_{(\boldsymbol{p}, \boldsymbol{p}_H) \in \boldsymbol{D}(H)} \mu_{\boldsymbol{p}}(\boldsymbol{x})\ (u_{\boldsymbol{p}} - u_{\boldsymbol{p}_H}) + \sum_{\boldsymbol{p} \in \boldsymbol{D}(G)} \boldsymbol{\eta}_{\boldsymbol{p}}(\boldsymbol{x})^T \cdot (\nabla u_{\boldsymbol{p}} - \boldsymbol{g}), \tag{1.15}$$

with $\mu_{\boldsymbol{p}}$ and $u_{\boldsymbol{p}}$ the function and coefficient associated to the data point $\boldsymbol{p} \in \boldsymbol{D}(H)$, $\boldsymbol{\eta}_{\boldsymbol{p}}$ and $\nabla u_{\boldsymbol{p}}$ the sets of derivative functions and coefficients in each coordinate of space associated to the gradient datum $\boldsymbol{g}$'s tail position $\boldsymbol{p} \in \boldsymbol{D}(G)$, and $u_{\boldsymbol{p}_H}$ the coefficient associated to the horizon H's reference point $\boldsymbol{p}_H$.

In addition, the function u is taken as a random function with a polynomial drift of small degree (1 usually). This drift stabilizes the solution and accounts for the inherent non-stationarity of the problem. This polynomial term must honor some regularization constraints written as

$$\sum_H \sum_{\boldsymbol{p} \in \boldsymbol{D}(H)} m(\boldsymbol{p})\ a_m + \sum_{\boldsymbol{p} \in \boldsymbol{D}(G)} \nabla m(\boldsymbol{p})^T \cdot \boldsymbol{a}_m = 0 \quad \forall m \in \mathcal{P}, \tag{1.16}$$

$$\tag{1.17}$$

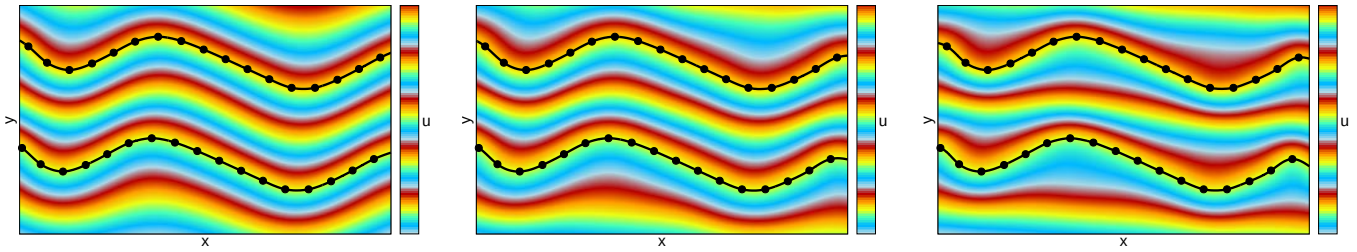
with m a monomial of the polynomial basis $\mathcal{P}$ and a_m its associated constant coefficient. These regularization constraints come in addition to the data constraints in the final system of equations (Section 1.3.2.8).

1.3.2.4 Global meshless formulation

An equivalence between kriging and splines is given by Matheron (1981) and Dubrule (1984). The analytic form of u (without gradient) is written

$$u(\boldsymbol{x}) = \sum_{\boldsymbol{p} \in \boldsymbol{D}} \Phi(||\boldsymbol{x} - \boldsymbol{p}||)\ u_{\boldsymbol{p}} + \sum_{m \in \mathcal{P}} m(\boldsymbol{x})\ a_m, \tag{1.18}$$

with $\boldsymbol{D}$ the set of data points, Φ the generalized covariance or spline and $u_{\boldsymbol{p}}$ and a_m the unknowns of the problem.



Thin Plate Splines | Cubic covariance | Wendland C2 (Wendland, 1995)

Figure 1.18: *PFM results with different interpolants. The parameters of each interpolant can be tuned to produce closer results.*

In meshless vocabulary, splines are part of the Radial Basis Functions (RBF) (Section 2.1.3.3, Equation (2.22)). Equation (1.18) then corresponds to

$$\begin{aligned} u(\boldsymbol{x}) &= \sum_{\boldsymbol{p}\in\boldsymbol{D}} \Phi_{\boldsymbol{p}}^{RBF}(\boldsymbol{x})\ u_{\boldsymbol{p}} \quad + \quad \sum_{m\in\mathcal{P}} m(\boldsymbol{x})\ a_m \quad , \\ &= \boldsymbol{\Phi}(\boldsymbol{x})^T \cdot \boldsymbol{U} \quad + \quad \mathcal{P}(\boldsymbol{x})^T \cdot \boldsymbol{a} \quad . \end{aligned} \tag{1.19}$$

This statement opens many opportunities as applications of RBF are numerous and implicit structural modeling can benefit from the amount of research realized in both geostatistics and meshless approaches.

The analytic form of u was first proposed by Cowan *et al.* (2002) for volume contouring and then, ARANZ Geo adapted it to structural modeling (Alcaraz *et al.* (2011), p.3). Hillier *et al.* (2014) describe thoroughly the analytical approach of PFM for structural modeling. They create a single surface to represent fold structures, and three types of constraints are considered: data points on the single iso-surface (Equation (1.5) with $\alpha_H = 0$), tangent gradient data (not discussed here), and gradient data. A fourth constraint is also considered for inequality data points (discussed in Section 1.4.2.1). Equation (1.18) is rewritten using the Hermite-Birkhoff interpolation with RBF (Wu, 1992), also called Hermitian RBF (Section 2.1.3.6), which handles the gradient data. A simplified version of their implicit function u is of the form

$$u(\boldsymbol{x}) = \sum_H \sum_{\boldsymbol{p}\in\boldsymbol{D}(H)} \Phi_{\boldsymbol{p}}^{RBF}(\boldsymbol{x})\ u_{\boldsymbol{p}} + \sum_{\boldsymbol{p}\in\boldsymbol{D}(G)} \nabla\Phi_{\boldsymbol{p}}^{RBF}(\boldsymbol{x})\ \boldsymbol{\eta}_{\boldsymbol{p}} + \sum_{m\in\mathcal{P}} m(\boldsymbol{x})\ a_m, \tag{1.20}$$

where $\boldsymbol{\eta}_{\boldsymbol{p}}$ are the unknowns in each coordinate of space for each gradient datum's associated node $\boldsymbol{p} \in \boldsymbol{D}(G)$.

The formalism of Hillier *et al.* (2014) can be directly applied to structural modeling with additional iso-surfaces defined and constrained by increment data points (Section 1.3.2.2, Equation (1.13)). In this case, Equation (1.15) and Equation (1.20) are equivalent for given pairs of generalized covariance and RBF.

To shorten the notations in the further sections, PFM is discussed with the global meshless formulation without the gradient data (i.e., Equation (1.19)). In this case, obtaining a solution is ensured by using the data point constraints from Section 1.3.1.2 where iso-values are guessed by a preprocessing. The next examples also use this framework.

1.3.2.5 Global smooth interpolation

PFM constructs the implicit function u with the data constraints. In between data positions, the interpolation is controlled by the RBF and may vary greatly depending on the chosen functions and their parameters (Figure 1.18). As in DSI, the assumption of smoothness to represent stratigraphic surfaces is often made. The used RBF are therefore chosen accordingly.

The most used interpolant in the geostatistics approach of PFM is the cubic covariance (Chilès *et al.*, 2004; Aug, 2004; Calcagno *et al.*, 2008; de la Varga *et al.*, 2019):

$$\Phi_{\boldsymbol{p}}(\boldsymbol{x}) = \Phi(r) = \begin{cases} C_0 \left(1 - 7(\frac{r}{\rho})^2 + \frac{35}{4}(\frac{r}{\rho})^3 - \frac{7}{2}(\frac{r}{\rho})^5 + \frac{3}{4}(\frac{r}{\rho})^7\right) & \forall r \in [0;\rho[\\ 0 & \forall r \geq \rho \end{cases} , \tag{1.21}$$

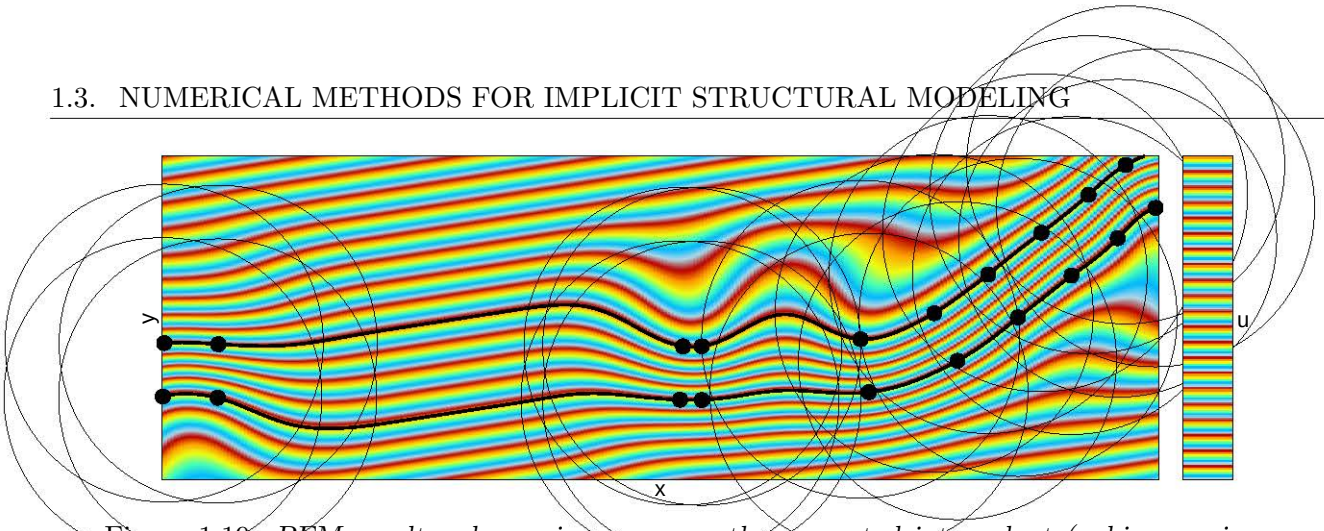


Figure 1.19: *PFM results when using a compactly supported interpolant (cubic covariance here) with a range smaller than the model's dimensions. The limit of each support is represented by a circle centered on its data point.*

with $r = ||\boldsymbol{x} - \boldsymbol{p}||$ the Euclidian distance between $\boldsymbol{x}$ and the center point $\boldsymbol{p}$ of the interpolant $\Phi_{\boldsymbol{p}}$, C_0 the sill and ρ the range of the variogram.

In meshless vocabulary, this interpolant is called a Compactly supported Radial Basis Function (CRBF, Section 2.1.3.3) where the center node $\boldsymbol{p}$ has an influence on its surroundings up to the distance ρ. In PFM, the points describing a same stratigraphic sequence should influence each other. The range ρ is thus scaled on the dimensions of the area where a sequence exists. It can be slightly smaller than the area to avoid considering all data everywhere without exception, but a great number of neighbors is generally preferred. Figure 1.19 shows the influence of a small range ρ on the results: the transition between areas influenced by CRBF and areas only defined by the polynomial drift lead to non-negligible artifacts in the solution. The interpolation is therefore necessarily performed globally: even CRBF are enforced to play the role of a global interpolant.

Though judged less robust than CRBF (Calcagno *et al.*, 2008), globally supported RBF have also been used and studied for PFM, such as the Gaussian function, polyharmonic splines, and multiquadrics (Lajaunie *et al.*, 1997; Hillier *et al.*, 2014; Jessell *et al.*, 2014). The polyharmonic Thin Plate Splines (TPS) (Duchon, 1977) represent a special interest in this thesis and are further discussed in Sections 2.1.3.3 and 2.2 (p.79). In the following examples, the TPS are used as interpolation functions.

1.3.2.6 Handling heterogeneous data

The accepted numerical objects and the related data constraints are described in Section 1.3.2.2. Here, the quality and the spatial distribution of numerical data are discussed (Section 1.1.4.5).

Dubrule (1984) uses the work of Ahlberg *et al.* (1967) to differentiate interpolating splines and smoothing splines. The system of equations defined without gradient data is written with data point constraints (Equation (1.5)) and regularization constraints on the polynomial term (Equation (1.16)) as

$$\begin{cases} u(\boldsymbol{p}) = \alpha_H & \forall H,\ \forall \boldsymbol{p} \in \boldsymbol{D}(H) \\ \sum\limits_{H} \sum\limits_{\boldsymbol{p} \in \boldsymbol{D}(H)} m(\boldsymbol{p})\ a_m = 0 & \forall m \in \mathcal{P} \end{cases} . \tag{1.22}$$

This is using interpolating splines, and it is equivalent to kriging interpolation. It performs an exact interpolation where the iso-surfaces pass exactly through the data points positions. Figure 1.20(a) shows how PFM behaves on a noisy and irregularly sampled data set when written with Equation (1.22).

Smoothing splines, or cokriging, add an error $\epsilon_{\boldsymbol{p}}$ to the interpolation at each data point $\boldsymbol{p}$:

$$\forall H,\ \forall \boldsymbol{p} \in \boldsymbol{D}(H), \quad u(\boldsymbol{p}) = \alpha_H + \epsilon_{\boldsymbol{p}}. \tag{1.23}$$

The system to solve is re-written with the variance of this error $s^2_{\boldsymbol{p}}$ as

$$\begin{cases} u(\boldsymbol{p}) + s^2_{\boldsymbol{p}}\ u_{\boldsymbol{p}} = \alpha_H & \forall H, \forall \boldsymbol{p} \in \boldsymbol{D}(H) \\ \sum\limits_{H} \sum\limits_{\boldsymbol{p} \in \boldsymbol{D}(H)} m(\boldsymbol{p})\ a_m = 0 & \forall m \in \mathcal{P} \end{cases} . \tag{1.24}$$

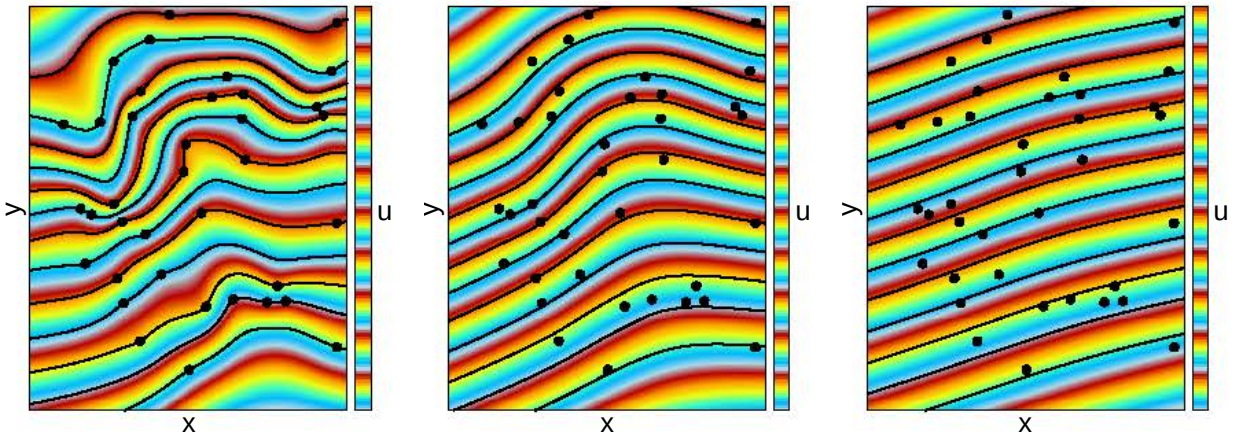


Figure 1.20: *Influence of the nugget $s^2_{\boldsymbol{p}}$ on irregularly distributed and noisy data points. (a) A null nugget fits perfectly the data points, (b) An average nugget filters the noise, and (c) A great nugget oversmooths the structures.*

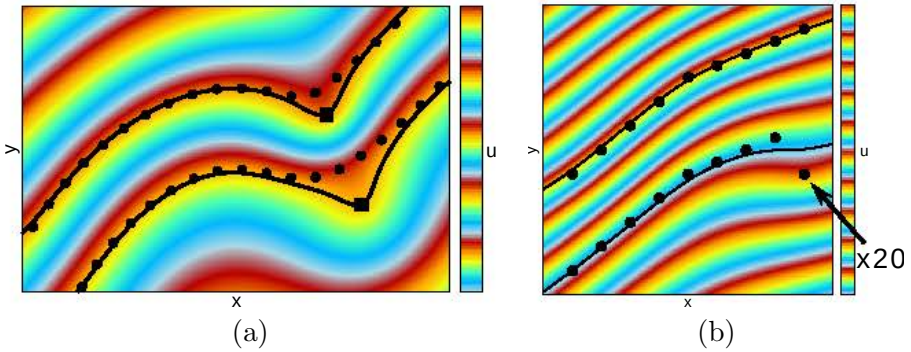


Figure 1.21: *Advantages of the smoothing splines with nugget for handling data heterogeneity. (a) Handling hard data with a null nugget, and (b) Repetitive equations have no influence on the results (test from Figure 1.16(b)).*

Inversely, interpolating splines (Equation (1.22)) are a specific case of smoothing splines where the interpolating error is null (i.e., Equation (1.24) with $s^2_{\boldsymbol{p}} = 0$). In practice, it is recommended to put at least an infinitesimal error value for stability reasons (de la Varga *et al.*, 2019).

In geostatistics, the variance $s^2_{\boldsymbol{p}}$ is the nugget of the variogram estimated at point $\boldsymbol{p}$. It is dependent on the type of RBF, the range of influence, and the distribution of the data points. It can be understood as a maximum distance away from a data point where the associated iso-surface can pass. The higher the nugget's value, the smoother the interpolation. Figure 1.20(b) and (c) show the effects of two nuggets uniformly applied to all data points: an average value efficiently smooths the noise while a too great value also smooths the structures.

An advantage with PFM is the simplicity with which the highly trusted data can be ensured. Like in Mallet (2002), the data points are often separated into two sets depending on their reliability: the hard data and the soft data. With the smoothing splines, a non-negligible nugget value is given to the soft data, and a null nugget is given to the hard data. Figure 1.21(a) shows how both soft data and hard data are handled with these constraints.

Another advantage of PFM is the objectivity concerning repetitive data points. In opposition to DSI, the concept of weight on the constraints does not exist, so a repetitive point in the model does not influence the global system of equations (Figure 1.21(b)). However, as the nugget is dependent on the distribution of data points, the grouping effect observed in DSI also affects the results obtained with PFM (Figure 1.22).

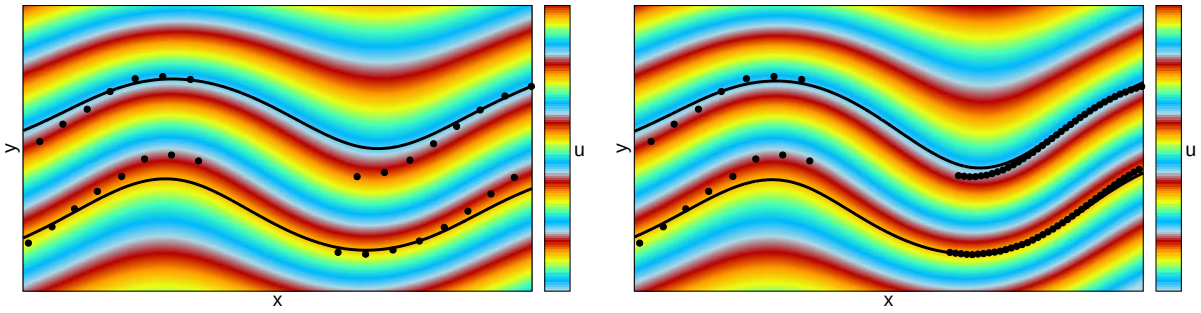


Figure 1.22: *Influence of the data density on the results with PFM. (a) The fold is sparsely sampled, and (b) The right side of the fold is densely sampled and better honored with the same nugget value.*

1.3.2.7 Handling structural discontinuities

The approach presented and discussed in this section comes from the works of Chilès *et al.* (2004) and Calcagno *et al.* (2008). It may be related to the needs in the mining industry, but the discontinuities (except from the unconformities) are rarely discussed for PFM in the literature.

Stratigraphic unconformities and intrusive geobodies

As described in Section 1.2.2.2, stratigraphic unconformities are handled with a separate stratigraphic function per sequence. For the intrusive geobodies, the implicit function is truncated as described in Section 1.2.2.4.

Jumping drifts for faults

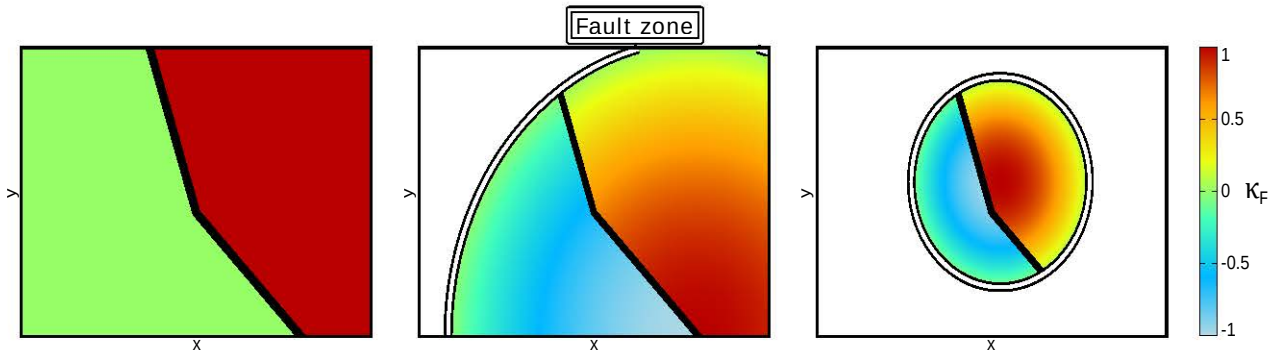
Each fault F is associated with an enrichment of u by a polynomial drift. Each polynomial drift is itself associated with a jump function κ_F defined on the fault zone of F. As its name indicates, the role of the jump function is to make the implicit function u jump from one side of the fault to the other by weighting discontinuously the enrichment on the solution. The implicit function u is re-written as

$$\begin{aligned} u(\boldsymbol{x}) &= \sum_{H} \sum_{\boldsymbol{p}\in\boldsymbol{D}(H)} \Phi_{\boldsymbol{p}}^{RBF}(\boldsymbol{x})\; u_{\boldsymbol{p}} \;+\; \sum_{m\in\mathcal{P}} m(\boldsymbol{x})\; a_m \;+\; \sum_{F} \sum_{m_F\in\mathcal{P}_F} \kappa_F(\boldsymbol{x})\; m_F(\boldsymbol{x})\; a_{mF} \quad , \\ &= \boldsymbol{\Phi}(\boldsymbol{x})^T \cdot \boldsymbol{U} \;+\; \boldsymbol{\mathcal{P}}(\boldsymbol{x})^T \cdot \boldsymbol{a} \;+\; \sum_{F} \kappa_F(\boldsymbol{x})\; \boldsymbol{\mathcal{P}}_F(\boldsymbol{x})^T \cdot \boldsymbol{a}_F \quad , \\ &= \boldsymbol{\Phi}(\boldsymbol{x})^T \cdot \boldsymbol{U} \;+\; \boldsymbol{\mathcal{P}}(\boldsymbol{x})^T \cdot \boldsymbol{a} \;+\; \boldsymbol{B}(\boldsymbol{x})^T \cdot \boldsymbol{b} \quad , \end{aligned} \tag{1.25}$$

with $\mathcal{P}_F$ the polynomial basis associated to the fault F and $\boldsymbol{a}_F$ their corresponding coefficients.

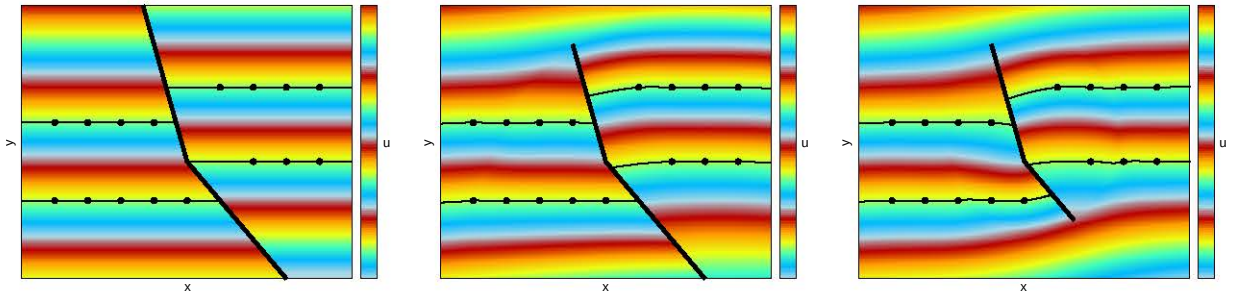
Fault zones

In 2D, faults are classified by the number of tips they have in the domain of study. Three cases are considered: infinite faults with zero tip, semi-finite faults with one tip, and finite faults with two tips. The fault zone of an infinite fault is infinite (i.e., it contains the domain Ω entirely, Figure 1.23(a)). The fault zone of a semi-finite or a finite fault is represented as a limited volume which boundary intersects the visible fault tips. For simplicity, these volumes are here created with circles: centered on the intersection between the domain's boundary and the semi-finite fault, or centered in the middle of the two tips of a finite fault (Figure 1.23(c) and (e)). Other shapes are possible but their influence on the solution is not discussed in the literature, only the influence of the length of a fault is shown on a few synthetic schemes in Calcagno *et al.* (2008). The fault zones are separated into two areas called, hereafter, the positive and the negative sides.



(a) Infinite fault (b) Semi-finite fault (c) Finite fault

Figure 1.23: *Definition of fault zones and related jump functions κ_F for three fault categories.*



(a) Infinite fault (b) Semi-finite fault (c) Finite fault

Figure 1.24: *PFM results using the jump functions defined in Figure 1.23.*

Jump functions

Jump functions κ_F are defined following two ideas: (i) the closer to the fault zone's visible boundaries, the closer to 0, and (ii) the farther from the fault zone's visible boundaries, the closer to 1 on the positive side and -1 on the negative side. The color scale of Figure 1.23 illustrates the evolution of κ_F in space. These fault zones are automatically created in our prototype. The jump function of an infinite fault is created as a polarity function with Boolean operations. The negative side of the fault is set to zero as done in the literature. The other jump functions are created with a circle equation (elliptical profiles are used in Calcagno *et al.* (2008)) and Boolean operations to delimit the positive and negative sides. The resulting stratigraphic functions using the presented jump functions are shown in Figure 1.24. Note the slight deviation at the borders of the circles delimiting the fault zones in the implicit function u.

Fault connectivity issue

Dealing with sealed fault blocks (Figure 1.25(a)) is equivalent to considering several stratigraphic sequences (Section 1.2.2.5). In this case, a stable approach is to compute an implicit function for each fault block separately and use Boolean operations to form the resulting stratigraphic function (Figure 1.25(b)). To preserve the dependency of the structures from one fault block to another, it should be possible to compute a unique discontinuous function using the jumping drifts. In Figure 1.25(c), the jump functions used to relate the discontinuities are created with the infinite approach for each fault separately. The fault zones are thus constructed as in Figure 1.25(a) and they overlap. Strong deviations in the stratigraphic function are observed which is not realistic geologically (Section 1.1.5). In Figure 1.25(d), we attempted to reduce the mutual dependency between fault blocks by defining a different polynomial drift per fault block (i.e., the jump functions are not overlapping, but all the data points still influence each other through the RBF). The results are more stable, but the structures also dive abnormally at the bottom of the model.

In the case of complex fault networks involving semi-finite and finite faults, non-overlapping fault zones may be tedious to define if created with circles and Boolean operations as in Figures 1.23(b)

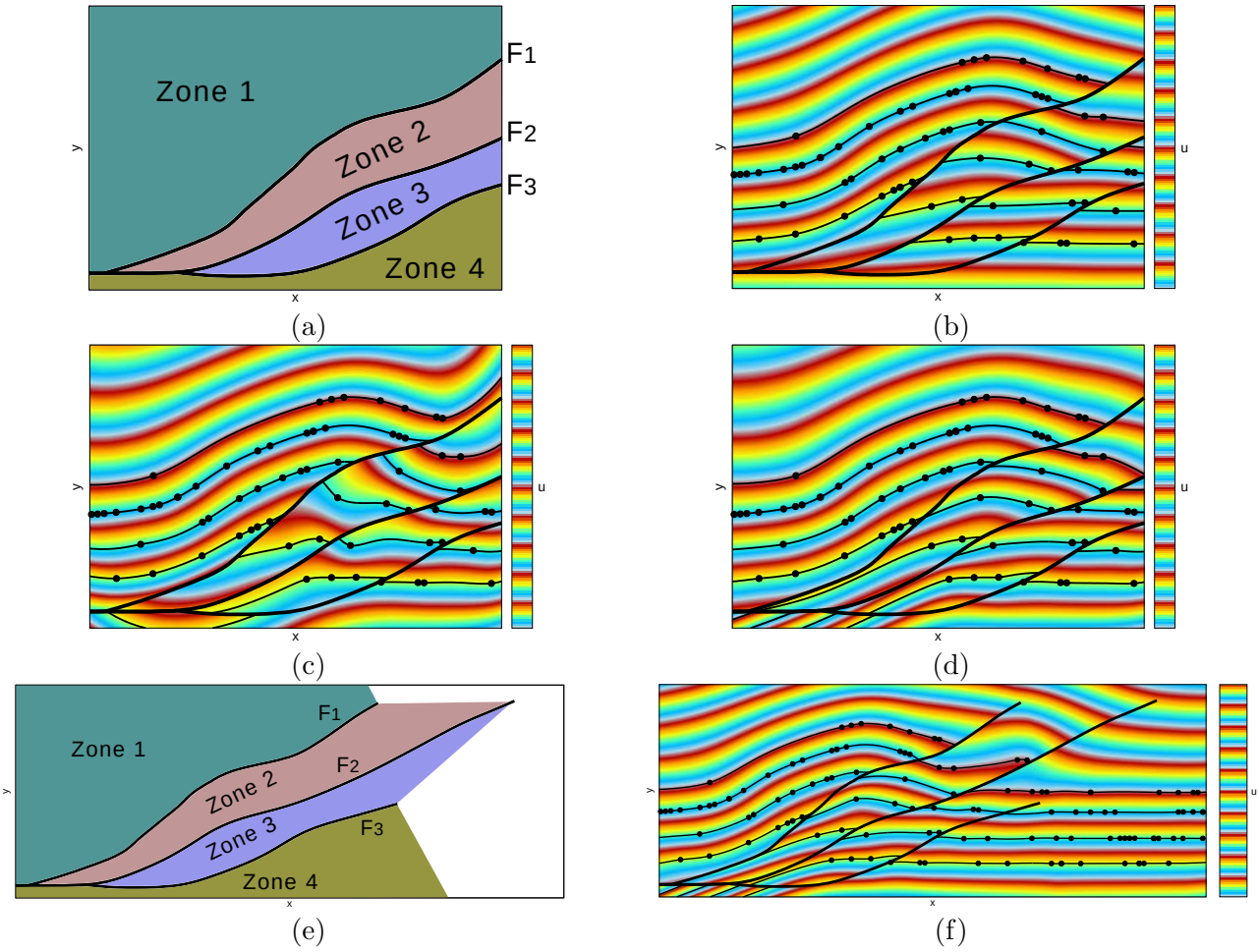


Figure 1.25: *Definition of fault zones and modeling results with PFM when dealing with fault connectivity. (a) The fault blocks used to compute (b), (c) and (d), (b) Obtained result computing a different stratigraphic function per fault block and combining these functions with Boolean operations, (c) Obtained result using jumping drifts with overlapping jump functions, (d) Obtained result using jumping drifts with a jump function per fault block, and (e) The fault zones manually interpreted to compute (f) using jumping drifts with a jump function per fault zone.*

and (c). In the model of Figure 1.25(e) for instance, the fault zone of F2 would contain the other two faults, which is inconvenient to define the jump function. This is not restrictive to the approach with circles; the method generally calls for complex algorithms to create the fault zones of arbitrary fault networks, and it often requires manual interventions. Therefore, we have created the fault zones manually (Figure 1.25(e)) and have computed a jump function per fault zone that vanishes at the boundaries of the fault zone. The obtained results also dive abnormally at the bottom left of the model (Figure 1.25(f)).

The approaches used to produce Figures 1.25(d) and (f) are the same as presented in Calcagno *et al.* (2008), but the deviation issue is not discussed in the paper. To the author's knowledge, no other article has ever mentioned it. It is probable that we are missing a key point here. If not, most PFM papers use normal gradient data which stabilize efficiently the solution; they also rarely extract intermediate iso-values between horizons, neither do they visualize the stratigraphic function's variation within the layers. Such specific conditions could explain why such behaviors were not observed before. We do not observe this issue in faulted models with less pronounced folds.

1.3.2.8 Example of PFM system of equations

The smoothing splines system (Equation (1.24)) is developed with the expression of u considering the faults (Equation (1.25)). This system is written with the data point constraints (Equation (1.5)) and the regularity constraints on the general and the fault related polynomial drifts (Equation (1.16)):

$$\begin{cases} [\quad (\boldsymbol{\Phi}(\boldsymbol{p}) + \boldsymbol{s_p})^T \quad \mathcal{P}(\boldsymbol{p})^T \quad \boldsymbol{B}(\boldsymbol{p})^T \quad] \cdot \mathcal{U} = \alpha_H & \forall H, \ \forall \boldsymbol{p} \in \boldsymbol{D}(H), \\ [\quad m(\boldsymbol{p}_1) \quad \dots \quad m(\boldsymbol{p}_{\#\boldsymbol{D}}) \quad \mathbf{0} \quad \mathbf{0} \quad] \cdot \mathcal{U} = 0 & \forall m \in \mathcal{P}, \\ [\quad \kappa_F \, m_F(\boldsymbol{p}_1) \quad \dots \quad \kappa_F \, m_F(\boldsymbol{p}_{\#\boldsymbol{D}}) \quad \mathbf{0} \quad \mathbf{0} \quad] \cdot \mathcal{U} = 0 & \forall F, \ \forall m_F \in \mathcal{P}_F, \end{cases} \tag{1.26}$$

with $\boldsymbol{s_p}$ a vector of null coefficients except for the nugget value $s^2_{\boldsymbol{p}}$ at the position associated to the data point $\boldsymbol{p}$, $\mathcal{U}^T = [\boldsymbol{U} \quad \boldsymbol{a} \quad \boldsymbol{b}]$ the vector of unknowns, and $\{\boldsymbol{p}_1, \dots, \boldsymbol{p}_{\#\boldsymbol{D}}\} = \boldsymbol{D}$ the set gathering all data points.

1.3.2.9 Visualizing the results

As the implicit function is supported by data points, extracting the iso-surfaces call for an additional discretization of the domain Ω. Generally, the results are evaluated on a grid using Equation (1.25), and marching element-based extractions are used. This evaluation stage may be computationally demanding (i.e., most of the interpolants $\Phi^{RBF}_{\boldsymbol{p}}$ are not null for all corner grid points) and holds the same approximations as discussed in Section 1.2.1.4.

1.3.2.10 Advantages and limits

A method assessing the data

The main advantage of PFM resides in its dual meaning in geostatistics and numerical methods. For instance, the geostatistician can give some knowledge on the anisotropy of the structures described by a given distribution of data. This can be partially included in the interpolation by modifying the generalized covariance parameters (Aug (2004): Section 4.3.3, Appendices B.2). There are many research studies on controlling the interpolation on a data set with different RBF parameters (Fasshauer & Zhang, 2007; Mongillo, 2011; Scheuerer *et al.*, 2013). These studies could be of benefit to better relate geological structures.

The created implicit function in PFM is fixed for a given data set. Therefore, the differences observed between two resolutions of visualization are related to the approximation during the extraction phase only, and not to the continuity of the implicit function itself. In addition, this data-based structure simplifies the code architecture as compared to other methods (e.g., see the open source code of de la Varga *et al.* (2019)). This formalism also allows to handle the concepts of hard and soft data efficiently with the nugget effect.

A method of poor scalability

The benefits of PFM in most applications in structural modeling are unfortunately limited. RBF functions are defined globally, giving rise to a dense system to solve (i.e., almost all the coefficients $\Phi^{RBF}_{\boldsymbol{p}}$ in system (1.26) are not null). The size of this system is dependent on the number of data N_D and it becomes unsolvable for more than a few thousand data without optimization techniques (Cowan *et al.*, 2003; Yokota *et al.*, 2010).

Handling the faults calls for a definition of fault zones which is challenging to perform automatically. Yet, the presented issues in 2D remain simple as compared to 3D cases. Though the concepts are the same, the definition of volumetric fault zones considering all the connectivities in 3D space is limiting when dealing with big fault networks (hundreds or thousands of faults). It is also difficult to create a water tight model (Section 1.1.5) with meshless techniques and marching element-based extractions (Section 1.2.1.4).

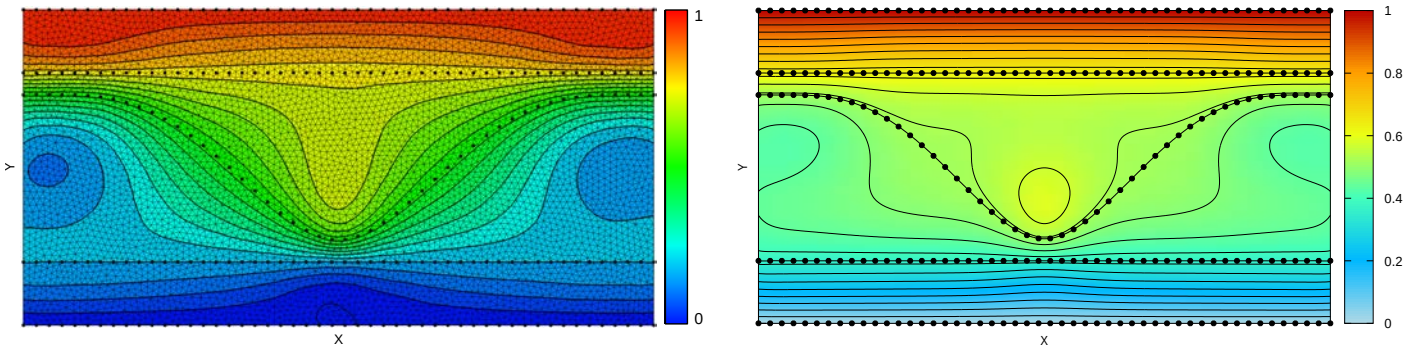


Figure 1.26: *Common thickness variation issue with (a) DSI and (b) PFM, where stratigraphic inconsistencies are created.*

1.3.3 Common modeling limits

The limits of both PFM and DSI concerning geological faults have already been discussed in the Sections 1.3.1.9 and 1.3.2.10. This is the major scalability restriction of these two methods. The memory challenges for big models in DSI and the dense system to solve in PFM also represent some difficulties, but they can be optimized to a certain extent (Section 1.4.2.2).

Both methods are using an assumption of smoothness on the results that has been proven to be efficient in applied studies (Lindsay *et al.*, 2013; Philippon *et al.*, 2015). It indeed gives an advantage to filter the noise in the data and handle sparsely distributed data (Figures 1.13 and 1.20). However, as smooth implicit modeling involves smoothing a function over the entire volume, smoothing horizon surfaces also smooths the volumes in between. It therefore states intrinsically that layers are of constant width. This reasoning shows that the smoothness assumption is in contradiction with thickness variations which are frequent in geological studies (e.g., for growth strata for instance). To illustrate this, Figure 1.26 shows how DSI and PFM are performing on an unrealistic thickness variation case. In several parts of the models, the norm of the implicit function's gradient drops to zero, creating stratigraphic inconsistencies (i.e., closed horizons, Section 1.1.5) often called the *bubble effect*.

Another limit of smoothing methods is the difficulty to propagate geometries where data are missing. In areas with no information, smoothing tends to create regular, parallel iso-surfaces by reducing their curvature (Hillier *et al.*, 2014).

Lastly, although it is more a comment than a limit, DSI and PFM have a different assumption concerning the inter-dependence of structures on either side of a fault. The local interpolation and conformal mesh of DSI render two fault blocks completely independent one from another (Figure 1.27(a)). The global interpolation and polynomial drift (Equation (1.19)) inherently impose a mutual influence in the case of PFM (Figure 1.27(b)). The question is: what is the most relevant scheme? It depends on the scale of the study, the fault deformations intensity, and the data sampling. In any case, it is possible to impose the inter-dependency in DSI with additional constraints on the fault's duplicated elements (throw constraints), and the independency in PFM with multiple stratigraphic functions.

1.4 Improving the implicit structural methods

This section discusses the approaches used to improve the modeling results in the joint case of DSI and PFM. Two separate ideas can be distinguished: improving the available information or improving the modeling tools themselves.

1.4.1 Improving the data

For a given modeling tool, the data can be adapted to improve the modeling results and the efficiency of their production.

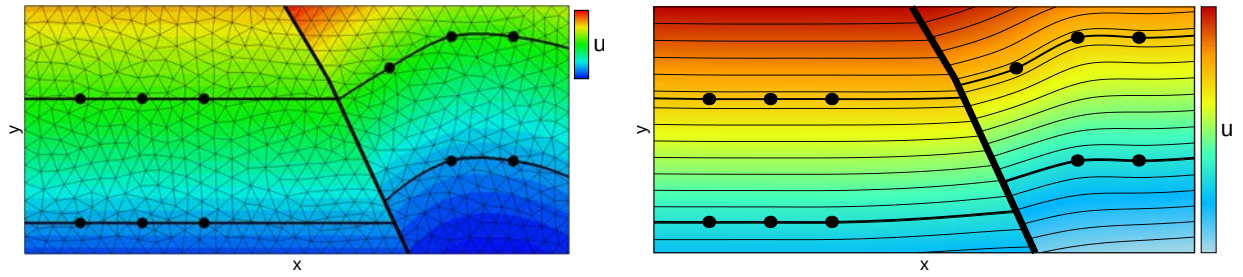


Figure 1.27: *Relation between two fault blocks and their structures. (a) Independent in DSI and (b) Inherently dependent in PFM.*

Cleaning the data

In PFM for instance, although a large amount of data is not computationally conceivable, the structures only need to be sampled with a few, meaningful data points (Figure 1.21). Therefore, it is possible to decimate the abnormally clustered data to increase the computational efficiency of PFM and obtain the same results. Automatic decimation algorithms in this context are complex as they have to account for the *meaning* of data, which is both dependent on the modeling capability of PFM and the expected structures, but it is an existing approach (Carmichael & Ailleres, 2016).

Adding artificial data

The problem of preserving the geological structures away from the data can be solved by adding non-observed data. For instance, Hillier *et al.* (2014) propagate a fold in 3D by adding gradient data along the fold hinge. The equivalent problem and solution in 2D are proposed in Laurent (2016). Kink folds not captured by the data are generated in Caumon (2009) and Caumon *et al.* (2013) by separating the domain of study in two and guiding the smoothing factor of DSI with a different gradient orientation in each subdomain. This concept has been further developed in Massiot & Caumon (2010) where a 3D vectorial field orients the smoothing continuously in space. In the same idea, Laurent *et al.* (2016) and Grose *et al.* (2017) develop a framework to create 3D vectorial fields and enforce them as gradient data to model periodic folds and foliations. Finally, the norm of the implicit function's gradient can also be iteratively controlled with gradient data to reduce abnormal thickness variations Laurent (2016).

Prior knowledge imposed with data

Adding these vectorial fields as numerical constraints is a way to constrain the algorithm to a prior knowledge of the expected results. de la Varga *et al.* (2019) use a Bayesian inference approach to stochastically tune the distribution of data points based on several prior information; the inversion of gravity fields and topologies of layer connectivity across a fault are used.

In all these examples, many modeling tools may have been developed, but the implicit algorithm remains the same. The improvements concern the numerical data, their quality and representability of the expected structures. These improvements were made according to the used implicit algorithm.

1.4.2 Improving the modeling tool

For a given set of field information, the algorithm can be adapted to improve the modeling results and the efficiency of their production. A numerical algorithm can be improved in two ways: by addition of new capabilities and modification of its core functioning, or by numerical optimization.

1.4.2.1 Improving the algorithm

The minimum necessary to create a working DSI or PFM algorithm is described in Section 1.3. The cited authors suggest more than these simple versions.

Inequality constraints

Additional data constraints are suggested together with a way to incorporate them in the implicit algorithm. The most noticeable is the inequality data point constraint (Mallet, 2002; Chilès *et al.*, 2004). Classical data point constraints are sorted by horizons and are therefore sampled on the interface between two layers (Sections 1.2.1.3). In geological studies, there are many situations where only a facies is observed at a given position (e.g., a borehole often ends within a layer). In this case, the geologist may want to indicate that a given layer should contain the observed position. To ensure that, inequality constraints are used stating that the observed point should be above and/or under the layer's iso-surfaces. Such constraints lead to non-linear systems which are solved by iterative algorithms. In structural modeling, several algorithms were tested, such as the Gibbs sampler method (Chilès *et al.*, 2004) and the interior-point method (Hillier *et al.*, 2014). Though decisive in many geological studies, such constraints are rarely used in PFM and DSI because these solving techniques are computationally demanding.

Global anisotropy

The smoothing assumption in the interpolation/regularization can also be tuned in relation to a given data distribution. Aug (2004) (Section 4.3.3) modifies the cubic covariance parameters to reflect the global anisotropy of a data set with PFM. In the same idea, the tools given in Scheuerer *et al.* (2013) to make a suitable choice on the kernel definition (i.e., generalized covariance or RBF) regarding data distribution could be used in PFM. Gonçalves *et al.* (2017) intend to even avoid the decision making on the parameters of a given generalized covariance by using the maximization of the log-likelihood.

Local anisotropy in PFM

In PFM, the global interpolation limits the application of the previously presented studies when assessing the local anisotropy. In volume contouring, a local version of PFM has been proposed in Martin & Boisvert (2017) using a partitioning of the domain and an RBF interpolation per partition (i.e., Partition of Unity Methods with RBF, inherited from the literature on meshless methods (Section 2.1.3.5) and surface construction (Section 1.5.3.4). The local anisotropy of the targeted geobody is then imposed iteratively by adapting the partitions of the domain and the local interpolations per partition. Although this approach is promising, its application to structural modeling remains unclear (discussed in Section 1.5.3.4).

Normalizing the weighting system of DSI

The influence of the number of equations in the least squares system is addressed in Section 1.3.1.9 (Figure 1.16). This is usually controlled by some normalizations on the constraints weights sorted per type of constraints. For instance, the influence of the regularization constraints on the system is normalized by the number of nodes in the mesh. The same can be done with the data constraints and the number of data points. Other normalization techniques are possible, such as spatially defined to handle clustered data, but the overall concept usually lacks a robust mathematical background.

1.4.2.2 Optimizing the algorithm

Optimizing the numerical structure

The scalability limits of DSI are dependent on the fault network complexity and concern both the generation of the mesh and its storage. An efficient way to improve DSI is therefore to develop meshing tools adapted to the structural modeling problem. The creation of mesh generation algorithms adapted to boundary representation models is an active research topic (Pellerin *et al.*, 2014; Botella *et al.*, 2016). In PFM, the major limit is the dense system to solve, which comes from the shared influence of all data points. Domain decomposition methods (Beatson & Greengard, 1997) applied to RBF interpolation create the so-called Fast RBF (Beatson *et al.*, 2001; Carr *et al.*, 2001). The domain of study is discretized on a grid with overlapping subdomains (i.e., slightly bigger than a grid cell). The global solution is then obtained by computing each subdomain solution and iteratively updating their results. The overlapping areas ensure the convergence of the method. This concept can be applied

in PFM to reduce the computational cost of the method (Cowan *et al.*, 2003) but the definition of adequate subdomains may be challenging with irregularly distributed data.

Choosing the solver

The solver itself should be adapted to the modeling techniques. Well known linear system solvers such as Gaussian elimination (LU decomposition in matrix form), or Gauss-Seidel method are generally used regarding the system to solve (e.g., dense or sparse, symmetric or nonsymmetric, positive definite or not). More sophisticated solvers are also investigated. For instance, the Fast RBF are generally applied together with a conjugate gradient method, or a Generalized Minimal Residual method where the domain decomposition methods act as good preconditioners to these iterative solvers (Beatson *et al.*, 2001). Other methods could be considered, such as the H-matrix (Hackbusch, 1999) which could further structure the coefficients of the dense system to solve in PFM.

Parallelization

After optimization, the code can be further accelerated by parallelization on Graphic Processing Units (GPU). Both DSI and PFM have sequential steps which cannot be performed in parallel: (i) The fault network is manually created, this may include the definition of fault zones in PFM; (ii) DSI has the additional meshing step; (iii) they both create the system of equations; (iv) the system is solved; and (v) the function is generally evaluated on a mesh for visualization. Each automatic step is partly parallelized in commercial software (Schlumberger, 2018; ARANZ Geo, 2019), but no paper publicly details the employed methods. In surface construction, Fast RBF with Generalized Minimal Residual solver were efficiently parallelized on GPU to handle millions of data points (Yokota *et al.*, 2010; Cuomo *et al.*, 2013). No direct application to PFM are known by the author.

1.4.3 Aims of the thesis

Both the data and the algorithm approaches are complementary and decisive for the current and future challenges to represent models with increasing complexity and dimensions. In this thesis, we focus on the improvement of implicit algorithms.

Let us consider a hypothetic data set not handled by the presented versions of DSI and PFM. This can be because of an ill-conditioned problem for which no stratigraphic function can be found. It can also be that the interpolation is not considered geologically realistic by the expert (Section 1.1.5). As presented in Sections 1.3.1.9, 1.3.2.10 and 1.3.3, the issue may be related to the management of faults or the assumption of smoothness. In these cases, what opportunities of improvement of DSI and PFM do we have to fix the issue without modifying the data?

For the fault management in DSI, we can adapt the algorithm of mesh generation to boundary representation models (Section 1.4.2.2), or we can investigate techniques to introduce discontinuous jumps into the mesh-based shape functions. In PFM, we can investigate techniques to introduce a jump into the RBF (or the generalized covariance) without the definition of volumetric fault zones. For the assumption of smoothness, we can adapt the roughness criterion and the RBF to the data distribution and the aimed structures; with the constraints that the roughness criterion must be defined on a mesh, and that the RBF must be globally supported.

Therefore, improvements in DSI and PFM are dependent on their respective supports of interpolation: the mesh and the RBF. In both cases, the modeling problem is posed based on these respective spatial discretizations, which narrows the scope of opportunities for future developments. Instead, many numerical methods approach a modeling problem with continuous equations, which are then approximated by a discretization. The discretization hence comes after the definition of the problem.

In this thesis, we suggest investigating continuous equations for implicit structural modeling. It enables a large range of numerical methods and techniques to perform the interpolation and introduce the discontinuities. It also allows to address the smoothing issues at the continuous level, creating solutions which are conceptually independent from the chosen discretization. It finally allows to compare different modeling techniques by solving the same modeling problem with different discretizations. Our aim is therefore to suggest a formal mathematical framework for implicit structural modeling and to propose some numerical methods to efficiently solve this problem.

Table 1.1: *Similarities and differences between structural modeling and surface construction.*

Numerical data		**Expected models**	
Similarities			
Heterogeneous		Smoothness	
Noisy		High precision in local areas	
Irregular and clustered			
Differences			
Structural modeling	**Surface construction**	**Structural modeling**	**Surface construction**
Globally sparse	Globally dense	Multiple opened surfaces	One closed surface
Different reliabilities	Constant reliability	Discontinuities	No discontinuities
Wide portions of missing data	Small portions of missing data	A few sharp angles	Many sharp angles

1.5 Unique surface construction: A modeling problem analogous to structural modeling

1.5.1 Modeling problem

In 3D computer graphics, many numerical methods deal with the modeling of volumetric objects by their external surface. The numerical data describing these objects may be points, orientation vectors giving the tangential plan of the surface by its normal, and triangulated surfaces. Such data are generally obtained after 3D scanning of a real object. It generates dense and regular data, although this is dependent on the scanner. After the acquisition, the data are often decimated or perturbed to test a method's capabilities. Examples of well known test objects for such methods are the Stanford Bunny, Happy Buddha, Dragon, Lucy, Armadillo (Stanford Computer Graphics Laboratory) and the Utah teapot. The expected model is a unique closed surface representing at best the object, with its sharp angles and smooth areas, while fitting the data.

This problem is dealt with explicit and implicit methods. Some reviews of existing methods in this field can be found in Schall *et al.* (2005), Cheng *et al.* (2008), Gomes *et al.* (2009) and Berger *et al.* (2013, 2014). In the case of implicit modeling, the surface construction is posed as a level-set problem in the large sense (Equation (1.1)). Generally, the iso-surface is the null set, re-defining Equation (1.2) as

$$\forall \boldsymbol{x} \in \mathbb{R}^3 \quad u(\boldsymbol{x}) = 0, \tag{1.27}$$

where positive values often represent the interior of the modeled object, and negative values the exterior.

1.5.2 Analogy with structural modeling

Both structural modeling and surface construction are creating the same types of models using the same types of numerical data. Their similarities and differences on these two matters are presented in Table 1.1. These considerations are generalizations and are not always true, may it be for structural modeling or surface construction, but they give an idea of the possible bridges between the two approaches.

We give a short review of some numerical methods using the concept of implicit modeling in surface construction. Many of the cited articles in the following have played a key role in the progress of this thesis and we believe that further investigations in this field could benefit greatly the research in structural modeling algorithms.

1.5.3 Some implicit modeling techniques

1.5.3.1 Solving the Poisson equation

Kazhdan (2005) and Kazhdan *et al.* (2006) approximate the implicit function u as the solution of the Poisson equation

$$\Delta u(\boldsymbol{x}) = \nabla \cdot \boldsymbol{N}(\boldsymbol{x}), \tag{1.28}$$

where Δ is the Laplacian operator and $\nabla \cdot \boldsymbol{N}$ is the divergence of a known vector field representing the surface. This continuous field is computed with the numerical data which are required to be oriented points (i.e., each point is associated with a normal vector to the expected surface). In Kazhdan *et al.* (2006), this is performed with compactly supported Gaussian functions. Equation (1.28) is then solved in a multigrid approach. The surface is finally extracted with the average iso-value of all data points.

This Poisson approach is popular in surface construction (Kazhdan & Hoppe, 2013; Estellers *et al.*, 2016), and it has initiated new ideas (e.g., using wavelets, Manson *et al.* (2008), or signed distances, Calakli & Taubin (2011)). Although not applicable to structural modeling because of the requirement on oriented data points, it shows that formulating a surface modeling problem with a PDE can create a robust approach.

1.5.3.2 Radial Basis Functions (RBF) surfaces

This paragraph shows the close relationship between RBF surface construction and the Potential Field Method (PFM) (Section 1.3.2). Although discovered separately, many improvements in both fields can be compared and related. We believe that a thorough review of both literatures could benefit the two approaches.

Interpolation with Variational surfaces

The first noticeable application of RBF (Section 2.1.3.3) to surface construction is introduced in Turk & O'brien (1999) as the variational implicit surfaces. In this paper, the data points are also assumed oriented to define points with a small offset in the inside and/or on the outside of the closed surface to model. The points describing the surface are associated with a null iso-value, and the other points are associated with a fixed positive (inside) and negative (outside) iso-value. Such constraints are used in the exact same interpolation scheme as in Section 1.3.2.6 (Equation (1.22)), with Thin Plate Splines (TPS) and with an additional polynomial term and regularization constraints too.

Smooth surfaces and sharp features

Dinh *et al.* (2001) add a regularization weight on the diagonal of the square system to solve to control the smoothness of the created surface. This is equivalent to the smoothing splines from Section 1.3.2.6 (Equation (1.24)) and this weight can be compared to the nugget covariance structure in geostatistics. Dinh et al. also suggest to enforce some anisotropy in the global RBF they use (Section 1.4.2.1) in order to handle sharp edges.

Compactly supported interpolation

Compactly supported RBF (CRBF) (Wendland, 1995) is proposed in surface construction by Morse *et al.* (2001). They present the benefits of such functions as compared to the TPS and even test their method for different data sparsity. Nevertheless, the data remain regularly sampled in the domain of study (i.e., no wide areas with missing data are studied) and a closed surface is targeted, which eases the control of the compact supports dimensions.

Fast algorithms

The first modeling application of the domain decomposition methods with RBF is in surface construction in Carr *et al.* (2001) and the method is called Fast RBF. This paper has inspired the application of Fast RBF in volume contouring by Cowan *et al.* (2002, 2003) and, later, in structural modeling with the creation of the software LeapFrog (ARANZ Geo, 2019).

Handling orientation constraints

The Hermite-Birkhoff interpolation scheme with RBF (HRBF) (Wu, 1992) is also used in surface construction techniques to handle orientation data (Macêdo *et al.*, 2009; Gois *et al.*, 2013; Guo *et al.*, 2016). As in Section 1.3.2.4 (Equation (1.20)), the implicit function u is defined as a sum of conventional and derivative RBF. The interpolation system is constructed with data point constraints (Equation (1.5)) with the null iso-value (i.e., $\alpha_0 = 0$) for the surface to model and with gradient constraints (Equation (1.14)):

$$\begin{cases} \sum\limits_{\overline{\boldsymbol{p}}\in\boldsymbol{D}(0)} \Phi^{RBF}_{\overline{\boldsymbol{p}}}(\boldsymbol{p})\ u_{\overline{\boldsymbol{p}}} & + \sum\limits_{\overline{\boldsymbol{p}}\in\boldsymbol{D}(G)} \nabla\Phi^{RBF}_{\overline{\boldsymbol{p}}}(\boldsymbol{p})\cdot\boldsymbol{\eta}_{\overline{\boldsymbol{p}}} & = 0 \quad \forall \boldsymbol{p}\in\boldsymbol{D}(0) \\ \sum\limits_{\overline{\boldsymbol{p}}\in\boldsymbol{D}(0)} \nabla\Phi^{RBF}_{\overline{\boldsymbol{p}}}(\boldsymbol{p})\ u_{\overline{\boldsymbol{p}}} & + \sum\limits_{\overline{\boldsymbol{p}}\in\boldsymbol{D}(G)} \boldsymbol{H}\Phi^{RBF}_{\overline{\boldsymbol{p}}}(\boldsymbol{p})\cdot\boldsymbol{\eta}_{\overline{\boldsymbol{p}}} & = \boldsymbol{g} \quad \forall \boldsymbol{p}\in\boldsymbol{D}(G) \end{cases}, \tag{1.29}$$

with $\boldsymbol{D}(0)$ the set of data points on the surface to model, $\boldsymbol{D}(G)$ the set of gradient data positions, $\boldsymbol{g}$ the gradient datum at the position $\boldsymbol{p}\in\boldsymbol{D}(G)$, and $\boldsymbol{H}$ the Hessian matrix.

1.5.3.3 Moving Least Squares (MLS) surfaces

The Moving Least Squares (Section 2.1.3.4) are introduced in surface construction in Alexa *et al.* (2001) as an explicit technique. The local tangential plane to the surface is computed at any position $\boldsymbol{x}$ using MLS functions; and it is then projected on the data point set to approximate the targeted surface. MLS techniques are popular in surface construction for their stability, adaptivity, and inherent smoothing capability. Many different numerical methods based on MLS have been created in this field, a review dedicated to this topic can be found in Cheng *et al.* (2008). Most of these techniques are defined explicitly (called projection MLS surfaces), but bridges with implicit techniques (implicit MLS surfaces) are demonstrated in the review.

MLS functions are generally centered on data points in surface construction. In implicit techniques, some relations with PUMs are made and the supports are not always centered on data points then. Several techniques exist for MLS surfaces, such as adaptive supports (Dey & Sun, 2005), anisotropic supports (Adamson & Alexa, 2006), regularization terms for smoothing (Shen *et al.*, 2004) and sharp angles (Guennebaud & Gross, 2007), and Hermite-type MLS (Alexa & Adamson, 2009).

1.5.3.4 Partition of Unity Method (PUM) surfaces

A Partition of Unity Method (Section 2.1.3.5) is introduced in surface construction by Ohtake *et al.* (2003) as an implicit approach called Multi-level Partition of Unity (MPU). Three different bases depending on the oriented data points distribution are defined. This allows them to separately: smooth large regions, fit complex local features with an interpolation of higher order continuity, and handle local sharp angles with a piecewise linear basis. Each separate partition (i.e., support) is then smoothly combined with an extrinsic basis close to the Shepard function. The data points are oriented as to ensure a solution in all partitions.

Over the years, several sets of intrinsic bases have been proposed for MPU, such as CRBF (Co *et al.*, 2004; Tobor *et al.*, 2004; Wu *et al.*, 2005a; Ohtake *et al.*, 2006), polynomial functions (Mederos *et al.*, 2007), and implicit B-splines (Rouhani *et al.*, 2015). The main difficulty with this approach is to handle the large areas of missing data. This is generally addressed with large partitions and post processing steps smoothing the results (Nagai *et al.*, 2009).

In geology, PUMs have been applied to contour mapping (Pouderoux *et al.*, 2004) and volume contouring (Martin & Boisvert, 2017) with CRBF. These two applications are close to surface construction as a single iso-surface is created, and this surface is supposed to be extracted relatively close to data points. This formalism is adapted in Martin & Boisvert (2017) as the intrinsic bases and partitions are separately tuned to represent the local anisotropy with a rotation matrix during the evaluation of the RBF distances.

In structural modeling, iso-surfaces are assumed infinite and can be extracted anywhere in the domain, even far away from the data points. If we want to satisfy this constraint with PUM while having a few data points in each partition, the partitions may have altered geometries (e.g., thin rectangles in Figure 1.28). Such partitions can be complex to create and may deteriorate the interpolation away from the data (e.g., there may exist high curvatures in the computed iso-surface passing by the green

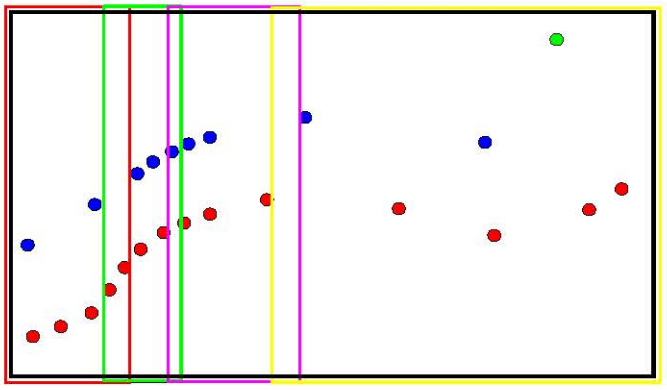


Figure 1.28: *Sketch of a domain partitioning to ensure the interpolation everywhere in the domain with a restricted number of data points in each partition (i.e., maximum of 7 points belonging to at least two different horizons). The distribution of the data points imposes the creation of thin and elongated partitions.*

point in Figure 1.28). Adding a polynomial drift defined everywhere in the domain (i.e., with the same monomials and coefficients in all subdomains and beyond) could help defining simpler partition geometries, but it could also create unwanted artifacts as discussed in Section 1.3.2.5 (Figure 1.19). Another solution is to sample additional nodes associated to regularization constraints as in DSI, but this would globally increase the number of unknowns.

Résumé du Chapitre 2

Dans ce Chapitre 2, nous présentons un problème continu pour la modélisation de structures géologiques par méthodes implicites.

Tout d'abord, un état de l'art est présenté sur les méthodes numériques permettant de résoudre un problème de modélisation continu. Nous proposons pour cela de considérer une équation aux dérivées partielles arbitraire définie avec un opérateur linéaire et des conditions au bords quelconques. Ce problème est alors approché par une approximation discrétisée où les intégrales sur le domaine d'étude issues des équations continues sont préservées. Plusieurs méthodes numériques sont ensuite présentées et classées selon les fonctions de test et les fonctions de forme qu'elles utilisent. D'autres concepts sont également présentés comme la gestion de surfaces de discontinuité par critères optiques ou par enrichissement avec des fonctions de saut locales.

L'ensemble de cet état de l'art est alors discuté dans le cadre d'une application en modélisation structurale, et un choix restreint de méthodes adaptés est sélectionné. Ce choix regroupe les fonctions de forme dites de moindres carrés glissants (Moving Least Squares) et d'éléments finis étendus (Extended Finite Element Method), ainsi que l'utilisation des fonctions de forme comme fonction de test, ou des fonctions Dirac.

Dans la suite de ce chapitre, nous définissons le problème continu proposé dans cette thèse. Ce dernier consiste à minimiser une somme de fonctionnelles : des fonctionnelles discrètes pour les contraintes de données, et des fonctionnelles continues pour la régularisation de l'interpolation. Le problème standard de modélisation structurale est alors donné par une régression spatiale de données ponctuelles pénalisée par l'énergie de courbure. Ce problème est ensuite approché par une approximation discrétisée, équivalente à celle utilisée comme référence dans l'état de l'art. Ainsi, cette nouvelle formulation de la modélisation structurale peut bénéficier de l'ensemble des méthodes numériques présentées dans ce chapitre.

Le problème de modélisation est ensuite adapté pour une grande partie des contraintes connues en modélisation structurale : les contraintes ponctuelles imposant une valeure ou un incrément, les contraintes vectorielles perpendiculaires et tangentes, les contraintes de lignes et de surfaces, les contraintes fortes et faibles, et les contraintes d'inégalité.

Enfin, le problème de modélisation est comparé aux méthodes présentées dans le Chapitre 1, et plusieurs liens sont établis.

Chapter 2

Continuous framework of implicit structural modeling

In this section, we present a typical approach to pose and solve continuous equations with numerical methods. We then discuss the applicability of this short review to implicit structural modeling. Based on this discussion, we propose a generic framework involving discrete and continuous terms, and we develop it for most of the existing structural modeling constraints.

2.1 Review of numerical methods for continuous modeling problems

2.1.1 Partial Differential Equation under constraints

Continuous modeling problems often deal with the characterization of the physical behavior of a continuous system approached by a PDE. It is defined on a domain Ω of the system as

$$\forall \boldsymbol{x} \in \Omega, L(u(\boldsymbol{x})) = f(\boldsymbol{x}), \tag{2.1}$$

with L a differential operator, u the searched solution function and f a known function. Only linear differential operators L are considered in this thesis.

A PDE is subject to boundary conditions (BC), such as Essential (or Dirichlet) Boundary Conditions (ECBs) written on the borders Γ_D of the domain as

$$\forall \boldsymbol{x} \in \Gamma_D, u(\boldsymbol{x}) = \alpha, \tag{2.2}$$

and/or Neumann Boundary Conditions (NBC) written on the borders Γ_N of the domain as

$$\forall \boldsymbol{x} \in \Gamma_N, < \nabla u(\boldsymbol{x}), \boldsymbol{N} >= \beta, \tag{2.3}$$

with α and β some scalar constants, $< ., . >$ the dot product operator, and $\boldsymbol{N}$ the normal vector to the boundary Γ_N. Although it is not explicit with this notation, several borders of EBC and NBC with different constants are generally employed. Other types of BC exist but they are not discussed in this thesis.

This section deals with some existing approaches and numerical methods to solve continuous equations such as a PDE.

2.1.1.1 Weak formulation

The problem described above is called strong formulation of the PDE, and it may be difficult to solve analytically. The method of weighted residuals yields that, under certain conditions, if a function u verifies

$$\int_\Omega v(\boldsymbol{x})\ L(u(\boldsymbol{x}))d\Omega = \int_\Omega v(\boldsymbol{x})\ f(\boldsymbol{x})d\Omega, \tag{2.4}$$

for any test function v on the global domain of study Ω, then u is also a solution to Equation (2.1) under constraint of Equations (2.2) and (2.3). Equation (2.4) is called weak formulation, or integral

formulation of the PDE. It exists in other derived forms but for simplicity, we will keep the present form as a reference.

Depending on the problem and its BC, Equation (2.4) can be further modified by applying the integration by parts. This diminishes (*resp.* increases) the order of derivation of the function u (*resp.* v) and let boundary terms appear. These boundary terms are generally eliminated with the BC. Therefore, PDEs involving high degrees of derivation may be approached by an approximation function u with a lower degree of derivability (e.g., an elliptical PDE approached with a piecewise linear function u).

The weak formulation still defines a continuous problem, difficult to solve analytically in most cases. To solve it numerically, the problem is discretized.

2.1.1.2 Discretization

Shape and test functions

The continuous functions u and v are approximated by discrete functions $\overline{u}$ and $\overline{v}$ defined on the domain Ω. The approximation function $\overline{u}$ is then taken in the space U as

$$U = \{\overline{u}(\boldsymbol{x}) = \sum_{l=1}^{N_U} \Phi_l(\boldsymbol{x})\ u_l = \boldsymbol{\Phi}^T \cdot \boldsymbol{U} \mid \boldsymbol{x} \in \Omega\}, \tag{2.5}$$

and the approximation function $\overline{v}$ is taken in the space V as

$$V = \{\overline{v}(\boldsymbol{x}) = \sum_{l=1}^{N_V} \Psi_l(\boldsymbol{x})\ v_l = \boldsymbol{\Psi}^T \cdot \boldsymbol{V} \mid \boldsymbol{x} \in \Omega\}, \tag{2.6}$$

with $\boldsymbol{\Phi} = \{\Phi_1, \ldots, \Phi_{N_U}\}$ the basis shape functions, $\boldsymbol{\Psi} = \{\Psi_1, \ldots, \Psi_{N_V}\}$ the basis test functions, $\boldsymbol{U} = \{u_1, \ldots, u_{N_U}\}$ the vector of unknown coefficients associated to the shape functions and $\boldsymbol{V} = \{v_1, \ldots, v_{N_V}\}$ the vector of chosen coefficients associated to the test functions. The number of test functions should be greater or equal to the number of shape functions (i.e., $N_V \geq N_U$). In the following, the functions $\overline{u}$ and $\overline{v}$ are respectively noted u and v for simplicity, the approximation being assumed.

Using the matrix form of u and v, the weak formulation can be written

$$\int_\Omega (\boldsymbol{\Psi}^T \cdot \boldsymbol{V})\ L(\boldsymbol{\Phi}^T \cdot \boldsymbol{U}) d\Omega = \int_\Omega (\boldsymbol{\Psi}^T \cdot \boldsymbol{V})\ f(\boldsymbol{x}) d\Omega, \tag{2.7}$$

with the operator L assumed linear in regards to $\boldsymbol{U}$ (i.e., $L(\boldsymbol{\Phi}^T \cdot \boldsymbol{U}) = L(\boldsymbol{\Phi})^T \cdot \boldsymbol{U}$), giving

$$\begin{aligned} \boldsymbol{V}^T \cdot \left(\int_\Omega \boldsymbol{\Psi} \cdot L(\boldsymbol{\Phi})^T d\Omega \right) \cdot \boldsymbol{U} &= \boldsymbol{V}^T \cdot \left(\int_\Omega \boldsymbol{\Psi}\ f(\boldsymbol{x}) d\Omega \right), \\ \left(\int_\Omega \boldsymbol{\Psi} \cdot L(\boldsymbol{\Phi})^T d\Omega \right) \cdot \boldsymbol{U} &= \left(\int_\Omega \boldsymbol{\Psi}\ f(\boldsymbol{x}) d\Omega \right). \end{aligned} \tag{2.8}$$

Discretization of the domain of study

This step depends on the chosen shape and test functions. Some numerical methods do not require a discretization of the domain Ω while others subdivide it into a set of subdomains $\boldsymbol{\Omega}$. These subdomains may overlap (i.e., $\exists(\omega, \omega') \in \boldsymbol{\Omega},\ \omega \cap \omega' \neq \emptyset$) or not (i.e., $\forall(\omega, \omega') \in \boldsymbol{\Omega},\ \omega \cap \omega' = \emptyset$). The best-known numerical methods are discretizing Ω into non-overlapping subdomains $\boldsymbol{\Omega}$. In this case, Equation (2.8) reduces to

$$\left(\sum_{\omega \in \boldsymbol{\Omega}} \int_\omega \boldsymbol{\Psi} \cdot L(\boldsymbol{\Phi})^T d\omega \right) \cdot \boldsymbol{U} = \left(\sum_{\omega \in \boldsymbol{\Omega}} \int_\omega \boldsymbol{\Psi}\ f(\boldsymbol{x}) d\omega \right). \tag{2.9}$$

The subdomain integrals are then approximated by quadrature rules such as Gaussian quadrature with a chosen number of approximation points. In the case where subdomains overlap, specific numerical integration rules are performed, such as special Gauss rules (De & Bathe, 2000, 2001) or integration on intersection domains and subdomain borders (Atluri & Shen, 2002).

2.1.1.3 System of equations and solution

The system of equations is given by Equation (2.8) after evaluation of the integral term. This matrix system can be written as

$$\boldsymbol{K} \cdot \boldsymbol{U} = \boldsymbol{F}, \tag{2.10}$$

where $\boldsymbol{U}$ is the vector of unknowns to be determined.

The system of Equation (2.10) is solved with numerical solvers supposedly adapted to the matrix $\boldsymbol{K}$ structure (e.g., dense, sparse, positive definite, etc...). If the problem is well posed (i.e., has a unique solution), the coefficients u_l are deduced, and the function u can be evaluated everywhere in the domain Ω with Equation (2.5).

2.1.1.4 Finite Element Method and meshless methods

The Finite Element Method (Bathe & Wilson, 1976; Dhatt & Touzot, 1981) is a numerical method that solves Partial Differential Equations under boundary conditions. It approximates the solution function on a domain of study Ω with the weak formulation, and discretizes the problem on a mesh. It is then a specific case of the presented approach where the domain is discretized into a mesh, and this mesh is also used to build the test and shape functions.

As numerous physical problems deal with complex geometries, and therefore require complex mesh generation algorithms, a common intent has been to reduce the mesh related issues or even avoid the generation of any mesh. This is why other types of discretizations have been investigated in the past forty years, creating the concept of meshless methods (MM). Using the weak formulation of a problem, these methods can be defined by their specific way of constructing the test and shape functions. The following sections represent a mere preview of these methods; excellent surveys can be found in Belytschko *et al.* (1996b), Fries & Matthias (2004), Liu & Gu (2005) and Nguyen *et al.* (2008).

2.1.2 Test functions

2.1.2.1 Collocation method

Collocation method is a generic term for any method that defines test functions as Dirac functions (Dirac, 1958):

$$\forall \boldsymbol{x} \in \Omega, \quad \Psi_{\boldsymbol{p}}(\boldsymbol{x}) = \delta(\boldsymbol{x} - \boldsymbol{p}) = \delta_{\boldsymbol{p}}(\boldsymbol{x}), \tag{2.11}$$

with $\boldsymbol{p}$ a collocation point. Let $\boldsymbol{N}_V$ be the set of collocation points in Ω, then the Dirac function simplifies the integral as

$$\forall \boldsymbol{p} \in \boldsymbol{N}_V, \quad \int_\Omega \delta_{\boldsymbol{p}}(\boldsymbol{x})\ g(\boldsymbol{x}) d\Omega = g(\boldsymbol{p}), \tag{2.12}$$

with g an arbitrary function. The system of Equation (2.8) simplifies to

$$\begin{cases} \left(\int_\Omega \delta_{\boldsymbol{p}}(\boldsymbol{x})\ L(\boldsymbol{\Phi})^T d\Omega\right) \cdot \boldsymbol{U} = \left(\int_\Omega \delta_{\boldsymbol{p}}(\boldsymbol{x})\ f(\boldsymbol{x}) d\Omega\right), & \forall \boldsymbol{p} \in \boldsymbol{N}_V, \\ L(\boldsymbol{\Phi}(\boldsymbol{p}))^T \cdot \boldsymbol{U} = f(\boldsymbol{p}), & \forall \boldsymbol{p} \in \boldsymbol{N}_V. \end{cases} \tag{2.13}$$

The PDE strong formulation (Equation (2.1) is thus written on each collocation point separately in the system and no integration is computed. The absence of integration makes collocation methods conceptually simpler and computationally less demanding than other methods (Fries & Matthias, 2004). On the other hand, using a Dirac function gives a strong approximation of the solution. Solving the strong form of a PDE is not stable on Neumann boundary conditions either (Liu & Gu, 2005). Many comparative studies in different fields of application, including Liu's (in mechanics), have proven better results with other methods than collocation ones.

2.1.2.2 Bubnov-Galerkin

A Bubnov-Galerkin (BG) method defines test functions to be the same as the shape functions, hence

$$\forall \boldsymbol{x} \in \Omega, \quad \Psi_l(\boldsymbol{x}) = \Phi_l(\boldsymbol{x}). \tag{2.14}$$

The number of shape and test functions is therefore equal (i.e., $N_V = N_U$). The system of Equation (2.8) becomes

$$\left(\int_\Omega \boldsymbol{\Phi} \cdot L(\boldsymbol{\Phi})^T d\Omega \right) \cdot \boldsymbol{U} = \left(\int_\Omega \boldsymbol{\Phi} \; f(\boldsymbol{x}) d\Omega \right). \tag{2.15}$$

The BG method is used in the Finite Element Method (FEM) where the shape functions built on the mesh are also used for integration. Although not compulsory, a background grid is generally used for integration in MM defined with a BG scheme. In this case, the domain Ω is discretized on non-overlapping subdomains, the mesh elements $\boldsymbol{E}$, and Equation (2.9) reduces to

$$\left(\sum_{e \in \boldsymbol{E}} \int_e \boldsymbol{\Phi} \cdot L(\boldsymbol{\Phi})^T de \right) \cdot \boldsymbol{U} = \left(\sum_{e \in \boldsymbol{E}} \int_e \boldsymbol{\Phi} \; f(\boldsymbol{x}) de \right). \tag{2.16}$$

2.1.2.3 Petrov-Galerkin

A Petrov-Galerkin (PG) method defines test functions that are neither Dirac nor shape functions:

$$\forall \boldsymbol{x} \in \Omega, \quad \Psi_l(\boldsymbol{x}) \neq \delta_l(\boldsymbol{x}); \quad \Psi_l(\boldsymbol{x}) \neq \Phi_l(\boldsymbol{x}). \tag{2.17}$$

PG methods were born with the intention of creating *truly* meshless methods. The integration is performed on the local supports of the shape functions, with a meshless test function. This could be done in a BG scheme, but Atluri *et al.* (1999b) proved that with the complex form of meshless shape functions, a BG integration required too many Gauss points to obtain convergent results (hence the background grid). By using meshless test and shape functions that are different one from another, Atluri & Zhu (1998) and Zhu *et al.* (1998) created new methods later generalized in Atluri & Shen (2002) as Meshless Local Petrov-Galerkin (MLPG) methods.

In their paper, Atluri & Shen (2002) differentiate global weak form (Equation (2.4)) and local weak form where integration terms are written on each local support separately together with the boundary conditions written on the subdomain's borders. This is how they perform the integration on an arbitrary set of subdomains, overlapping or not.

Table 2.1: *Main features of the presented shape functions.*

		Type of functions	Compatibility	KΔP	PU	Consistency	Polynomial basis	Continuity
Global	**RBF**	Radial	-	No	No	None	-	$\propto \boldsymbol{\Phi}$
	FSol	Radial	-	No	No	None	-	$\propto \boldsymbol{\Phi}$
	PIM (on Ω)	Rational polynomial	-	Yes	Yes	$order(\mathcal{P})$	$\#\mathcal{P} = \#\boldsymbol{N}$	$order(\mathcal{P})$
	WLS (on Ω)	Rational polynomial	-	No	Yes	$order(\mathcal{P})$	$\#\mathcal{P} \geq \#\boldsymbol{N}$	$order(\mathcal{P})$
	RPIM (on Ω)	Rational radial, polynomial	-	Yes	Yes	$order(\mathcal{P})$	Small $order(\mathcal{P})$	$\propto (\boldsymbol{\Phi}, order(\mathcal{P}))$
Local	**FEM**	Mesh-based	Yes	Yes	Yes	$\propto e$	$\propto e$	$\propto e$ (C^0 at node)
	NEM	Voronoi-based	Yes	Yes	Yes	Linear	-	C^∞ (C^0 at node)
	CRBF	Radial	Yes	No	No	None	-	$\propto \boldsymbol{\Phi}$
	PIM (on ω)	Rational polynomial	No	Yes	Yes	$order(\mathcal{P})$	$\#\mathcal{P} = (\#\boldsymbol{N} \in \omega)$	$order(\mathcal{P})$
	WLS (on ω)	Rational polynomial	No	No	Yes	$order(\mathcal{P})$	$\#\mathcal{P} \geq (\#\boldsymbol{N} \in \omega)$	$order(\mathcal{P})$
	RPIM (on ω)	Rational radial, polynomial	No	Yes	Yes	$order(\mathcal{P})$	Small $order(\mathcal{P})$	$order(\mathcal{P})$
	MLS	Rational polynomial	Yes	No	Yes	$order(\mathcal{P})$	$\#\mathcal{P} \geq N_{ngb}(\boldsymbol{x})$	$\propto w$
	PUMs	Extrinsinc, Intrinsic basis	Yes	$\approx$	Yes	$\propto \boldsymbol{\Phi}$	-	$\propto w$
	hp-clouds	Extrinsic, Intrinsic basis	Yes	Yes	Yes	Enriched by $\boldsymbol{\Phi}$	$\boldsymbol{\Phi}$: arbitrary	$\propto \boldsymbol{W}$

CRBF	Compactly supported Radial Basis Functions	(Wendland, 1995)	e	Mesh element
FEM	Finite Element Method	(Dhatt & Touzot, 1981)	N_{ngb}	Number of neighbors
FSol	Fundamental Solutions	(Kupradze, 1964)	$\#\boldsymbol{N}$	Number of nodes
hp-clouds	hp-clouds	(Duarte & Oden, 1996)	$\#\mathcal{P}$	Number of monomials
NEM	Natural Element Method	(Sibson, 1980)	$order(\mathcal{P})$	Polynomial order
MLS	Moving Least Squares	(Lancaster & Salkauskas, 1981)	w	weight function
PIM	Polynomial Interpolation Method	(Liu & Gu, 2001c)	$\boldsymbol{W}$	Extrinsic basis
PUM	Partition of Unity Method	(Babuška & Melenk, 1997)	Ω	Domain of study
RBF	Radial Basis Functions	(Buhmann, 2000)	ω	Subdomain, support
RPIM	Radial Polynomial Interpolation Method	(Wang & Liu, 2002)	$\boldsymbol{\Phi}$	Intrinsic basis
WLS	Weighted Least Squares	(Liu & Gu, 2005)	$\propto$	Depend on

2.1.3 Shape functions

This section presents some shape functions and important concepts used in numerical methods. The main features of each presented basis functions are given in Table 2.1.

2.1.3.1 A few concepts

Nodal and Non-nodal basis functions

The function u is an approximation function defined on a basis functions. Two types of basis functions can be differentiated (Dhatt & Touzot, 1981): those supported by a set of nodes $\boldsymbol{N}$ in space, and the others. In the first case, each coefficient u_l and each function Φ_l are associated to a node $\overline{\boldsymbol{p}} \in \boldsymbol{N}$ and are respectively denoted as $u_{\overline{\boldsymbol{p}}}$ and $\Phi_{\overline{\boldsymbol{p}}}$ for unicity (i.e., $N_U = \#\boldsymbol{N}$). In the other case, no interpolation structure is used. The polynomial basis functions is an example of the latter case. In the following, nodal basis functions are presented but the discussed techniques and properties can also apply to non-nodal basis functions. A basis functions is always assumed nodal in the remainder of the thesis.

Compact supports

The support $S_{\overline{\boldsymbol{p}}}$ of a function $\Phi_{\overline{\boldsymbol{p}}}$ is an area where the function is not null. Such a function is hence null outside of its support. These areas may have different geometries, such as spheres or polygonal areas (i.e., mesh elements). They can be centered on a node, or not.

Complete cover

The cover Ω_C is the union area of all the compact supports (i.e., $\forall \overline{\boldsymbol{p}} \in \boldsymbol{N}$, $\Omega_C = \cup S_{\overline{\boldsymbol{p}}}$). The supports may overlap or not, the cover is only represented as the total area touched by the influence of any shape function. A cover Ω_C is said to be complete when the domain of study Ω is included in the cover (i.e., $\Omega \subset \Omega_C$).

Compatibility

Some locally supported bases functions produce a discontinuous interpolation at the transition from one compact support $S_{\overline{\boldsymbol{p}}}$ to another. A basis functions creating a continuous interpolation with compact supports is said compatible.

Kronecker Delta Property (KΔP)

A function $\Phi_{\overline{\boldsymbol{p}}}$ has the KΔP if and only if

$$\forall (\overline{\boldsymbol{p}}, \boldsymbol{p}) \in \boldsymbol{N}, \quad \Phi_{\overline{\boldsymbol{p}}}(\boldsymbol{p}) \begin{cases} = 1, & \boldsymbol{p} = \overline{\boldsymbol{p}}. \\ = 0, & \boldsymbol{p} \neq \overline{\boldsymbol{p}} \end{cases} . \tag{2.18}$$

It follows that a coefficient $u_{\overline{\boldsymbol{p}}}$ can substitute the evaluation of the implicit function at its node position $u(\overline{\boldsymbol{p}})$ as

$$u(\overline{\boldsymbol{p}}) = u_{\overline{\boldsymbol{p}}}. \tag{2.19}$$

Partition of Unity (PU)

When a basis functions forms a PU, it means that it can reproduce a constant field (Belytschko *et al.*, 1996b). This is a crucial notion as bases functions without the PU property are not able to find a constant solution u. A PU is formed if a basis functions $\boldsymbol{\Phi}$ honors

$$\forall \boldsymbol{x} \in \Omega, \quad \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}} \Phi_{\overline{\boldsymbol{p}}}(\boldsymbol{x}) = 1. \tag{2.20}$$

Consistency

The capacity of a method to reproduce a higher order polynomial solution is called consistency (Belytschko *et al.*, 1998). A basis functions is said p-consistent if it can fit exactly a polynomial solution of order p. "PU of order p" is also a term possible for this property.

Complete polynomial basis

Many shape functions are created using polynomials, and a complete polynomial basis is often preferred. A complete polynomial basis $\mathcal{P}$ of order $order(\mathcal{P})$ is composed of all the monomials with a total power of mixed variables smaller or equal to $order(\mathcal{P})$. Here are a few examples:

$$\begin{aligned}
1D,\quad & order(\mathcal{P}) = 0, \quad \mathcal{P} = \{1\}, \\
& order(\mathcal{P}) = 1, \quad \mathcal{P} = \{1, x\}, \\
& order(\mathcal{P}) = 2, \quad \mathcal{P} = \{1, x, x^2\}, \\
\\
2D,\quad & order(\mathcal{P}) = 0, \quad \mathcal{P} = \{1\}, \\
& order(\mathcal{P}) = 1, \quad \mathcal{P} = \{1, x, y\}, \\
& order(\mathcal{P}) = 2, \quad \mathcal{P} = \{1, x, y, x^2, y^2, xy\}, \\
\\
3D,\quad & order(\mathcal{P}) = 0, \quad \mathcal{P} = \{1\}, \\
& order(\mathcal{P}) = 1, \quad \mathcal{P} = \{1, x, y, z\}, \\
& order(\mathcal{P}) = 2, \quad \mathcal{P} = \{1, x, y, z, x^2, y^2, z^2, xy, xz, yz\}.
\end{aligned} \tag{2.21}$$

2.1.3.2 Using mesh elements

Finite Element Method

A finite element (Lagrangian element) is composed of the three following entities (translated from Pigeonneau (2011)):

1. A compact polyhedral with straight or curved faces and edges,
2. A set of nodes, also called degrees of freedom,
3. A vectorial space of functions called interpolation space.

An arbitrary function defined on such an element must be continuous. If it is also continuous from one element to another, the finite elements are said conformal. The FEM shape functions are constructed on the nodes of a mesh composed with such elements by assuming the created functions have the KΔP. Each function is therefore defined continuous everywhere in the mesh. It is associated to one node, it is not null in the finite elements having this node as degree of freedom, and it is null everywhere else in the mesh. Interested readers should follow the construction rules and processes detailed in Dhatt & Touzot (1981) for a given element.

Natural Element Method (NEM)

The NEM shape functions are constructed with Voronoi cells. Two types of NEM functions are differentiated: the Sibsonian (Sibson, 1980) and the non-Sibsonian functions (Belikov, 1997). The former ones are defined by ratio of polygonal areas of Voronoi cells, and the latter ones are constructed by ratio of Euclidian distances to neighboring nodes and lengths of Voronoi cell edges. Their construction and final form are thus different from FEM shape functions even if mesh cells are used. The construction of the Voronoi cells may not even be required to create the shape functions.

2.1.3.3 Using radial functions

Radial Basis Functions (RBF)

RBF is a generic term designating a basis functions centered on a node $\overline{\boldsymbol{p}}$ and dependent on the distance $r_{\overline{\boldsymbol{p}}}$ to this node:

$$\Phi(||\boldsymbol{x} - \overline{\boldsymbol{p}}||) = \Phi(r_{\overline{\boldsymbol{p}}}) = \Phi_{\overline{\boldsymbol{p}}}^{RBF}(\boldsymbol{x}). \tag{2.22}$$

Historically, RBF were introduced by Blinn (1982) with the Blobby Molecule. A multitude of RBF techniques followed, such as metaballs (Nishimura, 1985), soft objects (Wyvill *et al.*, 1986), and the blobby model (Muraki, 1991) (see (De Araújo *et al.*, 2015), p.4 and 5 for equations). Nowadays, most commonly used RBF are the Gaussian, the Multiquadrics and the Polyharmonic spline functions (see Mongillo (2011) and Buhmann (2000) for expressions).

Fundamental solutions

Some RBF are fundamental solutions (or Green's functions) of a given problem. If the function u is a linear combination of fundamental solutions of a PDE (Equation (2.1)), then u is necessarily a solution of this PDE. Depending on the considered PDE, these functions can be of different forms. For most of them, they are undefined at their center node (i.e., $\Phi_{\overline{\boldsymbol{p}}}^{FSol}(\boldsymbol{x}) = \emptyset$ at $\boldsymbol{x} = \overline{\boldsymbol{p}}$) and defined everywhere else.

The Thin Plate Splines (TPS) (Duchon, 1977) are widespread fundamental solutions. TPS are said to solve the biharmonic equation (i.e., $\Delta^2 u = 0$) and minimize the thin plate energy (Dubrule, 1984; Wahba, 1990). TPS are defined in 2D as

$$\Phi_{\overline{\boldsymbol{p}}}^{TPS}(\boldsymbol{x}) = (||\boldsymbol{x} - \overline{\boldsymbol{p}}||)^2 . log(||\boldsymbol{x} - \overline{\boldsymbol{p}}||), \tag{2.23}$$

and in 3D as

$$\Phi_{\overline{\boldsymbol{p}}}^{TPS}(\boldsymbol{x}) = (||\boldsymbol{x} - \overline{\boldsymbol{p}}||)^3. \tag{2.24}$$

Compactly supported RBF (CRBF)

Wendland (1995) defines RBF with a compact support. As an RBF shape function $\Phi_{\overline{\boldsymbol{p}}}$ is centered on a node $\overline{\boldsymbol{p}}$, the support $S_{\overline{\boldsymbol{p}}}$ is also centered on $\overline{\boldsymbol{p}}$ and is considered to be the domain of influence of this node up to a given radius of influence ρ, also called dilatation parameter. This dilatation parameter may vary from one node to another. In comparison, classical RBF may be seen as having an infinite domain of influence, or to be globally supported. Best-known CRBF and their expressions can be found in Wendland (1995). Another example is the fourth order spline, defined as

$$\Phi_{\overline{\boldsymbol{p}}}(\boldsymbol{x}) = \Phi(r) = \begin{cases} 1 - 6\left(\frac{r}{\rho}\right)^2 + 8\left(\frac{r}{\rho}\right)^3 - 3\left(\frac{r}{\rho}\right)^4 & if \quad r \leq \rho \\ 0 & if \quad r > \rho \end{cases} . \tag{2.25}$$

Concept of weight function

In most MM, a weight function is used as an enrichment of a function or an equation and is usually formulated as a CRBF. This may be used for two (nonexclusive) reasons: either to weight an expression in a least squares system, or to weight the pertinence of an expression in space. For instance, if a globally supported shape function Φ is only supposed to rule the interpolation in a restricted area, enriching it with a weight function w can control its influence in space.

Frequently used weight functions are the polynomial splines (e.g., the fourth order spline, Equation (3.10)), the normalized Gaussian and other normalized exponential functions (see Fries & Matthias (2004) p.45 for expressions). Different normalizations lead to different support geometries. The most usual ones are the spherical support as

$$w_{\overline{\boldsymbol{p}}}(\boldsymbol{x}) = w\left(\frac{||\boldsymbol{x} - \overline{\boldsymbol{p}}||}{||\boldsymbol{\rho}||}\right) = w\left(r\right), \tag{2.26}$$

and the cubic support as, in 2D,

$$w_{\overline{\boldsymbol{p}}}(\boldsymbol{x}) = w\left(\frac{|x - \overline{p}_x|}{\rho_x}\right) \; w\left(\frac{|y - \overline{p}_y|}{\rho_y}\right), \tag{2.27}$$

and as, in 3D,

$$w_{\overline{\boldsymbol{p}}}(\boldsymbol{x}) = w\left(\frac{|x - \overline{p}_x|}{\rho_x}\right) \; w\left(\frac{|y - \overline{p}_y|}{\rho_y}\right) \; w\left(\frac{|z - \overline{p}_z|}{\rho_z}\right), \tag{2.28}$$

with $\boldsymbol{\rho} = \{\rho_x, \rho_y, \rho_z\}$ the dilatation parameter of $\omega_{\overline{\boldsymbol{p}}}$.

Depending on the chosen function, there can be additional parameters to the problem (e.g., two in some exponential functions, Fries & Matthias (2004) p.45). Studies have shown that these parameters, and the chosen weight function itself, have a big impact on the results. Recommended values of parameters and functions can be found in Liu & Gu (2005), but it should be noted that these choices are application dependent.

2.1.3.4 Building an intrinsic Partition of Unity (PU)

Polynomial Interpolation Method (PIM)

Liu & Gu (2001c) approximate the solution function u by a polynomial function as

$$\begin{aligned} u(\boldsymbol{x}) &= \sum_{m \in \mathcal{P}} m(\boldsymbol{x})\ a_m &&= \boldsymbol{\mathcal{P}}^T(\boldsymbol{x}) \cdot \boldsymbol{a}, \\ &= \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}} \Phi_{\overline{\boldsymbol{p}}}^{PIM}(\boldsymbol{x})\ u_{\overline{\boldsymbol{p}}} &&= \boldsymbol{\Phi}^T(\boldsymbol{x}) \cdot \boldsymbol{U}. \end{aligned} \tag{2.29}$$

The polynomial form is used to enforce the KΔP by writing Equation (2.19) at each node, giving the system

$$\left\{ \quad \boldsymbol{\mathcal{P}}(\overline{\boldsymbol{p}})^T \ \cdot \ \boldsymbol{a} \ = \ \boldsymbol{U}, \quad \forall \overline{\boldsymbol{p}} \in \boldsymbol{N} \ , \right. \tag{2.30}$$

re-written with the moment matrix $\boldsymbol{A}$ gathering all the monomial terms as

$$\boldsymbol{A} \cdot \boldsymbol{a} = \boldsymbol{U}. \tag{2.31}$$

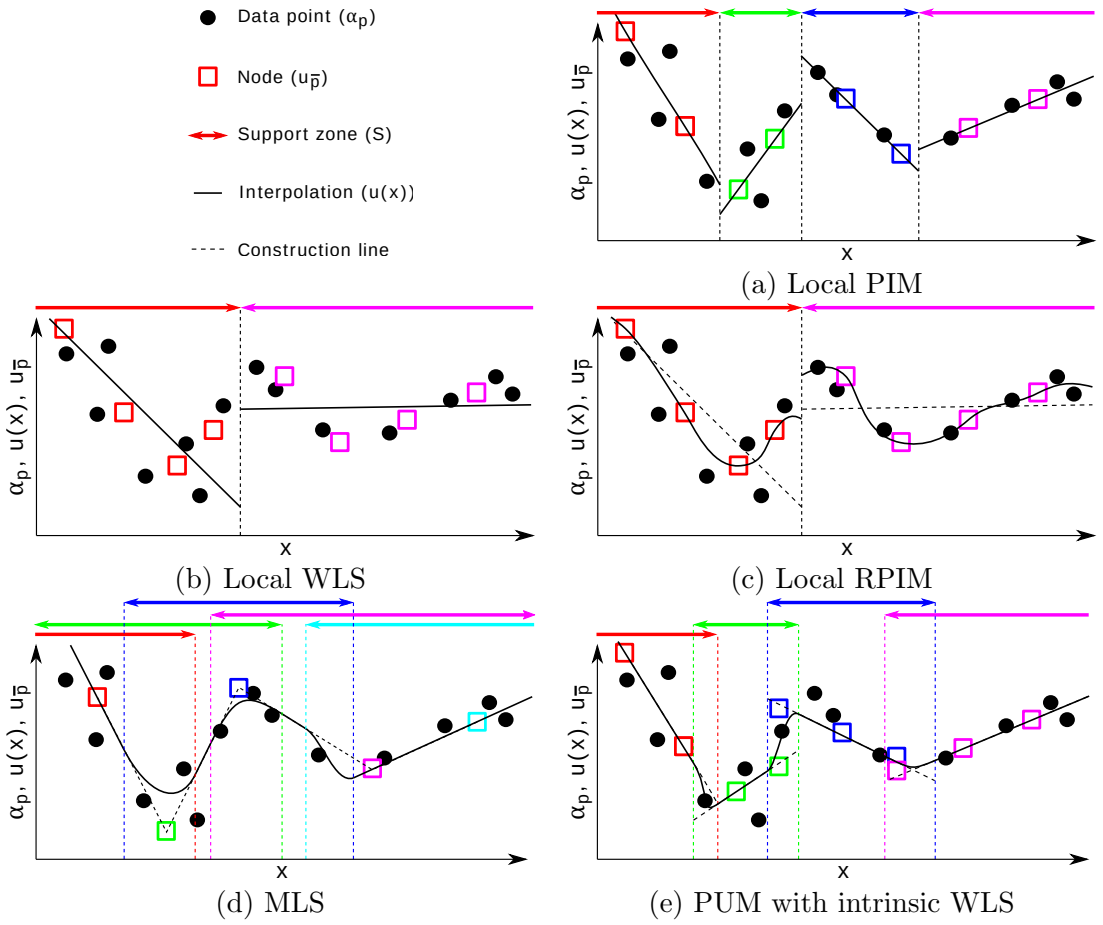


Figure 2.1: *Sketches of geometrical behaviors of some 1D interpolations using different bases functions. All methods use a first order polynomial. The local partitioning is purposely different between methods to illustrate some requirements on the length of the support.*

In PIM, the number of monomials must be equal to the number of nodes (i.e., $\#\boldsymbol{\mathcal{P}} = \#\boldsymbol{N}$). System (2.31) is therefore a square system to solve. Depending on the set of nodes, it may be ill conditioned or even singular. If invertible, the coefficients $\boldsymbol{a}$ are replaced in Equation (2.29) as

$$\begin{aligned} u(\boldsymbol{x}) &= \boldsymbol{\mathcal{P}}^T(\boldsymbol{x}) \cdot [\boldsymbol{A}]^{-1} \cdot \boldsymbol{U}, \\ &= \boldsymbol{\Phi}^T(\boldsymbol{x}) \cdot \boldsymbol{U}, \end{aligned} \tag{2.32}$$

giving the PIM shape functions.

PIM can be seen as an interpolation scheme written on the nodes to construct shape functions forming a PU and having the KΔP. The necessity to have an equal number of nodes $\#\boldsymbol{N}$ and of monomials $\#\boldsymbol{\mathcal{P}}$ can be handled either by tuning $\#\boldsymbol{N}$ for a given polynome, or tuning $\#\boldsymbol{\mathcal{P}}$ for a given set of nodes. Another possibility, illustrated in Figure 2.1(a), is to create distinct partitions of the domain of study Ω, with equal numbers $\#\boldsymbol{N}$ and $\#\boldsymbol{\mathcal{P}}$ in each. The PIM functions can then be computed in each partition separately.

PIM is thus considered to have a global form, written on the domain, and a local form, written on the partitions. In its local form, the support $S_{\overline{\boldsymbol{p}}}$ of a shape function $\Phi_{\overline{\boldsymbol{p}}}$ is the partition containing the associated node $\overline{\boldsymbol{p}}$. The influence of $\overline{\boldsymbol{p}}$ is null outside of the partition. In PIM, several nodes may share the same support. The constructed function u after resolution of a given problem (i.e., determining the coefficients $\boldsymbol{U}$) may be discontinuous at the interfaces between the supports: the local PIM functions are not compatible (Table 2.1).

Weighted Least Squares (WLS) approximation

Liu & Gu (2005) define the WLS as a way to overcome the singularity problem of PIM. Also, it is less restrictive in the sense that the number of monomials can be smaller than the number of nodes (i.e., $\#\boldsymbol{\mathcal{P}} \leq \#\boldsymbol{N}$). System (2.31) is thus overdetermined. This is solved in the least squares sense by enriching each equation with a weight value $\lambda_{\overline{\boldsymbol{p}}}$ and minimizing the corresponding weighted norm

$$J_{WLS}(\boldsymbol{a}) = \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}} \lambda_{\overline{\boldsymbol{p}}} (\boldsymbol{\mathcal{P}}(\overline{\boldsymbol{p}})^T \cdot \boldsymbol{a} - u_{\overline{\boldsymbol{p}}})^2. \tag{2.33}$$

The weights $\lambda_{\overline{\boldsymbol{p}}}$ are constants. As in PIM, the result of this inverse problem is then combined with Equation (2.29) to construct the WLS shape functions. Also, these functions can be defined globally or locally in the same manner as in PIM (Figure 2.1(b)). The difference is that Equation (2.19) is approached in the least squares sense, so the WLS functions do not have the KΔP (Table 2.1).

Radial Polynomial Interpolation Method (RPIM)

RPIM is another method introduced by Liu and Gu to overcome the singularity problem in PIM. It follows the same construction steps, but u is defined with RBF centered on the nodes $\boldsymbol{N}$ and enriched by a small order polynomial with a complete basis, giving

$$\begin{aligned} u(\boldsymbol{x}) &= \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}} \Phi^{RBF}_{\overline{\boldsymbol{p}}}(\boldsymbol{x})\ b_{\overline{\boldsymbol{p}}} + \sum_{m \in \boldsymbol{\mathcal{P}}} m(\boldsymbol{x})\ a_m &&\overset{(RBF)}{=} \boldsymbol{\Phi}^T(\boldsymbol{x}) \cdot \boldsymbol{b} + \boldsymbol{\mathcal{P}}^T(\boldsymbol{x}) \cdot \boldsymbol{a}, \\ &= \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}} \Phi^{RPIM}_{\overline{\boldsymbol{p}}}(\boldsymbol{x})\ u_{\overline{\boldsymbol{p}}} &&\overset{(RPIM)}{=} \boldsymbol{\Phi}^T(\boldsymbol{x}) \cdot \boldsymbol{U}. \end{aligned} \tag{2.34}$$

An interpolation scheme using the KΔP constraints (Equation (2.19)) and the polynomial regularization constraints (Equation (1.16), p.42) is then inverted to construct the RPIM shape functions. The created square system can be compared to the system of Equation (1.22) (p.44) (Dubrule, 1984), the difference being that the nodes are not necessarily centered on the data points in RPIM.

The RPIM functions can be defined globally or locally in the same manner as in PIM and WLS (Figure 2.1(c)). In their global form, even if CRBF are used instead of globally supported RBF in Equation (2.34), the RPIM functions remain globally supported because of the polynomial term. In their local form, their supports are the partitions of the domain, not the CRBF supports. Some applications do not need a polynomial term, but it reduces the consistency of the method (Table 2.1).

Moving Least Squares (MLS)

The MLS functions (McLain, 1976; Lancaster & Salkauskas, 1981) can be understood as a continuous version of the WLS functions. They approximate the solution function u as a polynomial function with coefficients that vary continuously in space as

$$\begin{aligned} u(\boldsymbol{x}) &= \sum_{m\in\mathcal{P}} m(\boldsymbol{x})\ a_m(\boldsymbol{x}) &&= \mathcal{P}^T(\boldsymbol{x})\cdot\boldsymbol{a}(\boldsymbol{x}), \\ &= \sum_{\overline{\boldsymbol{p}}\in\boldsymbol{N}} \Phi^{MLS}_{\overline{\boldsymbol{p}}}(\boldsymbol{x})\ u_{\overline{\boldsymbol{p}}} &&= \boldsymbol{\Phi}^T(\boldsymbol{x})\cdot\boldsymbol{U}. \end{aligned} \tag{2.35}$$

The same methodology as in WLS is then employed to construct the shape functions. The only difference is that in WLS, one set of coefficients $\boldsymbol{a}_S$ is inverted per support $S\in\boldsymbol{S}$. In MLS, the coefficients are inverted for any position $\boldsymbol{x}$ in the domain Ω. The continuous version of the WLS system (2.33) is minimized in the least squares sense, corresponding to

$$J_{MLS}(\boldsymbol{a}(\boldsymbol{x})) = \sum_{\overline{\boldsymbol{p}}\in\boldsymbol{N}} w_{\overline{\boldsymbol{p}}}(\boldsymbol{x})(\mathcal{P}^T(\overline{\boldsymbol{p}})\cdot\boldsymbol{a}(\boldsymbol{x}) - u_{\overline{\boldsymbol{p}}})^2, \tag{2.36}$$

where $w_{\overline{\boldsymbol{p}}}$ are spatially dependent weight functions. After recombining with Equation (2.35), the MLS shape functions are defined as

$$\Phi^{MLS}_{\overline{\boldsymbol{p}}}(\boldsymbol{x}) = w_{\overline{\boldsymbol{p}}}(\boldsymbol{x})\mathcal{P}^T(\boldsymbol{x})\cdot[\boldsymbol{A}(\boldsymbol{x})]^{-1}\cdot\mathcal{P}(\overline{\boldsymbol{p}}) = \boldsymbol{\Gamma}(\boldsymbol{x})\cdot\boldsymbol{B}_{\overline{\boldsymbol{p}}}(\boldsymbol{x}), \tag{2.37}$$

where

$$\boldsymbol{A}(\boldsymbol{x}) = \sum_{\overline{\boldsymbol{p}}\in\boldsymbol{N}} w_{\overline{\boldsymbol{p}}}(\boldsymbol{x})\mathcal{P}(\overline{\boldsymbol{p}})\cdot\mathcal{P}^T(\overline{\boldsymbol{p}}), \tag{2.38}$$

and with $\boldsymbol{A}\cdot\boldsymbol{\Gamma} = \mathcal{P}$ and $\boldsymbol{B}_{\overline{\boldsymbol{p}}} = w_{\overline{\boldsymbol{p}}}(\boldsymbol{x})\mathcal{P}(\overline{\boldsymbol{p}})$. The matrix $\boldsymbol{A}$ is called the moment matrix. Appendix A (p.165) gives the details of the MLS functions construction. Historically, the MLS were not formulated with the presented approach. Further details on the different origins of the MLS can be found in Fries & Matthias (2004).

The support $S_{\overline{\boldsymbol{p}}}$ of an MLS function is defined by its associated weight function $w_{\overline{\boldsymbol{p}}}$: it is centered on a node $\overline{\boldsymbol{p}}\in\boldsymbol{N}$ and its length of influence is controlled by the weight function's dilatation parameter ρ. To illustrate that, each support is centered on a different node in Figure 2.1(d).

MLS functions are also an alternative to the singularity problem of PIM: the moment matrix is theoretically invertible if the evaluated position $\boldsymbol{x}$ is within the support of at least $\#\mathcal{P}$ nodes. These nodes are called neighbors and their number $\#\boldsymbol{N}(\boldsymbol{x})$ is dependent on $\boldsymbol{x}$. Therefore, the supports must overlap, which explains the expansion of each subdomain in Figure 2.1(d). Further discussions on the MLS stability can be found in Liu & Gu (2005).

Reproducing Kernel Particle Method (RKPM)

These shape functions are created by discretization of the continuous Reproducing Kernel Method (RKM), which is inherited from the theory of wavelets. Details can be found in Fries & Matthias (2004). Only one point is noted here: a specific discrete form of RKM is exactly equivalent to the MLS functions. RKM is therefore a continuous generalization of MLS but coming from a completely different approach. Nowadays, most papers using RKPM functions are using the same discrete version as MLS functions.

2.1.3.5 Building an extrinsic Partition of Unity (PU)

Partition of Unity Method (PUM)

Babuška & Melenk (1997) introduce a method to ensure a PU everywhere in the domain of study with an arbitrary basis functions. The basis is enriched with regularization terms W_ξ as

$$\begin{aligned} u(\boldsymbol{x}) &= \sum_{S\in\boldsymbol{S}} W_S(\boldsymbol{x}) \sum_{\overline{\boldsymbol{p}}\in\boldsymbol{N}(S)} \Phi_{S\overline{\boldsymbol{p}}}(\boldsymbol{x})\ u_{S\overline{\boldsymbol{p}}}, \\ &= \sum_{S\in\boldsymbol{S}} W_S(\boldsymbol{x})\ \boldsymbol{\Phi}_S(\boldsymbol{x})^T\cdot\boldsymbol{U}_S, \end{aligned} \tag{2.39}$$

where $\boldsymbol{N}(S)$ is the set of nodes existing within the support S, the set of functions $\boldsymbol{\Phi}_S$ represents the intrinsic basis functions of each support S, $\boldsymbol{U}_S$ is the associated set of coefficients, and the functions W_S form the extrinsic basis. The functions W_S are often defined as the Shepard function:

$$W_S(\boldsymbol{x}) = \frac{w_S(\boldsymbol{x})}{\sum\limits_{S' \in \boldsymbol{S}} w_{S'}(\boldsymbol{x})}, \tag{2.40}$$

where w_S and $w_{S'}$ are the weight functions defining the supports S and S'. These functions are not necessarily centered on a node.

The intrinsic basis is arbitrary: all the shape functions described previously can be used, even a mix of different shape functions can be used between supports, and they do not need to form a PU. Figure 2.1(e) shows a PUM interpolation using WLS functions as intrinsic basis. The supports are chosen smaller than in the MLS case to show that there is no other requirement than having overlapping supports (i.e., no need to include neighboring nodes).

PUM functions ensure a continuous transition between subdomains interpolated separately. Therefore, each subdomain interpolation can be well adapted to local complexities or introduce *a priori* knowledge of the solution. The consistency of the PUMs depends on the consistency of the intrinsic basis. Although they do not have the KΔP, it is possible to use their extrinsic basis to recover it (Belytschko *et al.*, 1996b). The problem with PUMs is that they introduce more unknowns than the other techniques: as the supports overlap, there are more coefficients $u_{S\overline{\boldsymbol{p}}}$ than nodes $\#\boldsymbol{N}$. This is illustrated in Figure 2.1(e) by different colored nodes having the same position $\boldsymbol{x}$ and different coefficient values as they are involved in separate supports.

Hp-clouds

The hp-clouds (Duarte & Oden, 1996; Liszka *et al.*, 1996; Babuška & Melenk, 1997) represent a way to enrich a basis functions $\boldsymbol{\Phi}$ by local polynomial bases as

$$\begin{aligned} u(\boldsymbol{x}) &= \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}} W_{\overline{\boldsymbol{p}}}(\boldsymbol{x}) \left(u_{\overline{\boldsymbol{p}}} + \sum_{m \in \mathcal{P}_{\overline{\boldsymbol{p}}}} m(\boldsymbol{x})\ a_m \right), \\ &= \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}} W_{\overline{\boldsymbol{p}}}(\boldsymbol{x}) \left(u_{\overline{\boldsymbol{p}}} + \mathcal{P}_{\overline{\boldsymbol{p}}}(\boldsymbol{x})^T \cdot \boldsymbol{a}_{\overline{\boldsymbol{p}}} \right), \end{aligned} \tag{2.41}$$

where $\mathcal{P}_{\overline{\boldsymbol{p}}}$ and $\boldsymbol{a}_{\overline{\boldsymbol{p}}}$ are the polynomial basis and related coefficients associated to each node $\overline{\boldsymbol{p}}$ and the shape function $W_{\overline{\boldsymbol{p}}}$. The basis functions $\boldsymbol{W}$ is supposed to form a PU, so that each polynomial $\mathcal{P}_{\overline{\boldsymbol{p}}}$ can be chosen with a great flexibility: it may not be a complete polynomial basis, it may use different sets of monomials from one node to another, and it may not even enrich some shape functions (i.e., using one constant monomial). The hp-clouds have the same properties as the basis $\boldsymbol{W}$ (e.g., KΔP, consistency), but the consistency may be locally increased by the added polynomial terms (Table 2.1).

Discussion on Partition of Unity Methods (PUMs)

The notion of PUM is generic, meaning that many numerical methods can be classified as using the concept of PU to enrich a given basis functions. For instance, the hp-clouds are a specific case of PUMs where the usual basis shape functions $\boldsymbol{\Phi}$ replaces the extrinsic basis $\boldsymbol{W}$ forming a PU, and it is enriched by an intrinsic combination of monomials and coefficients. The Partition of Unity FEM (PUFEM) (Babuska & Melenk, 1995; Melenk & Babuška, 1996), Generalized FEM (GFEM) (Strouboulis *et al.*, 2000) and Extended FEM (XFEM) (Belytschko *et al.*, 2001) are also considered as PUMs (Babuška *et al.*, 2003).

2.1.3.6 Hermite-type shape functions

Concept and motivation

A Hermite-type basis adds $\#\boldsymbol{N}(G)$ sets of unknowns associated to derivative terms as

$$u(\boldsymbol{x}) = \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}(N)} \Phi_{\overline{\boldsymbol{p}}}(\boldsymbol{x})\ u_{\overline{\boldsymbol{p}}} + \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}(G)} \nabla \Phi_{\overline{\boldsymbol{p}}}(\boldsymbol{x})^T \cdot \boldsymbol{b}_{\overline{\boldsymbol{p}}}, \tag{2.42}$$

with ∇ the gradient operator, $\boldsymbol{N}(N)$ the set of nodes associated to conventional shape functions, $\boldsymbol{N}(G)$ the set of nodes associated to derivative functions, and $\boldsymbol{b}_{\overline{\boldsymbol{p}}}$ a set of coefficients associated to one derivative node $\overline{\boldsymbol{p}} \in \boldsymbol{N}(G)$. The derivative and conventional shape functions are supposed to be taken in the same class of functions, which is why the same letter Φ is used for both.

This concept can also be adapted to any shape functions with higher derivative terms if the expected order of derivation is possible with the chosen shape functions. Such approaches may increase the computational cost of a method and create discontinuities in derivative terms(Belytschko *et al.*, 1996a).

The enrichment with first order derivatives (Equation (2.42)) is widespread in numerical methods as it can be used to enforce the Neumann conditions efficiently (Liu & Gu, 2005). Also, such Hermite-type methods are used to enforce orientation constraints within the domain of study (Sections 1.3.2.4, p.42 and 1.5.3.2, p.55).

Some examples

This approach exists for most of the presented shape functions, such as Hermite-type RBF (Wendland, 2004), Hermite-type WLS and RPIM (Liu & Gu, 2005), Generalized MLS (Atluri *et al.*, 1999a) and Hermite RKPM (Liu *et al.*, 1996). A Hermite version of PUM is also possible using a Hermite-type intrinsic basis for instance.

Approximation of the MLS derivatives

In the cited Hermite-type methods, the shape functions are assumed to be at least twice derivable (i.e., C^2) to handle orientation data constraints with Equation (2.42). The derivative terms up to the second order are explicitly written. The computation of these terms for shape functions such as RBF (functions of a Euclidian distance) is straightforward and may not be computationally more demanding than the shape functions themselves. In the case of more complicated shape functions such as MLS (functions of rational polynomials and weight functions), the derivatives are often approximated to reduce computational costs.

The exact equations of MLS derivatives can be found in Fries & Matthias (2004). Here, we present the simplified forms given in Belytschko *et al.* (1996a) and Liu & Gu (2005) which are considered as good approximations of the exact forms. The first order derivative in the i axis corresponds to

$$\partial_i \Phi^{MLS}_{\overline{\boldsymbol{p}}}(\boldsymbol{x}) = \partial_i \boldsymbol{\Gamma}^T \cdot \boldsymbol{B}_{\overline{\boldsymbol{p}}} + \boldsymbol{\Gamma}^T \cdot \partial_i \boldsymbol{B}_{\overline{\boldsymbol{p}}}, \tag{2.43}$$

where $\partial_i \Gamma$ is solution of

$$\boldsymbol{A} \cdot \boldsymbol{\partial}_i \Gamma = \partial_i \mathcal{P} - \partial_i \boldsymbol{A} \cdot \boldsymbol{\Gamma}, \tag{2.44}$$

and the second order derivative in the i and j axes corresponds to

$$\partial^2_{ij} \Phi^{MLS}_l(\boldsymbol{x}) = \boldsymbol{\partial}^2_{ij} \Gamma^T \cdot \boldsymbol{B}_{\overline{\boldsymbol{p}}} + \boldsymbol{\partial}_i \Gamma^T \cdot \partial_j \boldsymbol{B}_{\overline{\boldsymbol{p}}} + \boldsymbol{\partial}_j \Gamma^T \cdot \partial_i \boldsymbol{B}_{\overline{\boldsymbol{p}}} + \boldsymbol{\Gamma}^T \cdot \partial^2_{ij} \boldsymbol{B}_{\overline{\boldsymbol{p}}}, \tag{2.45}$$

where $\partial^2_{ij} \Gamma$ is solution of

$$\boldsymbol{A} \cdot \boldsymbol{\partial}^2_{ij} \Gamma = \partial^2_{ij} \mathcal{P} - (\partial_i \boldsymbol{A} \cdot \partial_j \boldsymbol{\Gamma} + \partial_j \boldsymbol{A} \cdot \boldsymbol{\partial}_i \Gamma + \partial^2_{ij} \boldsymbol{A} \cdot \boldsymbol{\Gamma}). \tag{2.46}$$

With these equations, only the matrix $\boldsymbol{A}$ needs to be inverted to obtain all the derivatives, which does not require more computational effort than constructing the shape functions.

2.1.3.7 Other shape functions

Many other existing shape functions are not presented in this thesis. For the remaining shape functions, we can count the ones that (i) have no relation to the topic of this thesis, such as shape functions used in Smooth Particle Hydrodynamics (Lucy, 1977), (ii) are specific cases or variants of presented methods, such as the fixed least squares, variant to the Moving Least Squares (Onate *et al.*, 1996b), and (iii) are unknown to the author.

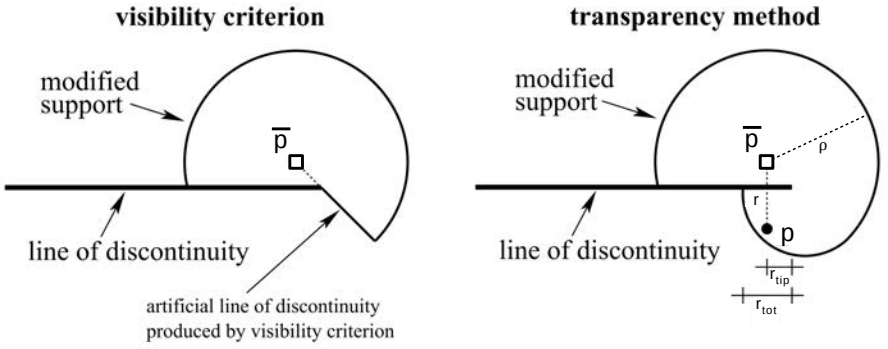


Figure 2.2: *Illustration of the effects of the visibility criterion and the transparency method on a compact support (modified from* Fries & Matthias *(2004)).*

2.1.4 Handling discontinuities

Discontinuities represent either an abrupt change in the physical properties of a continuous system, or a discontinuity in the system itself. If they have an influence on the studied physical phenomena, the discontinuities have to be included in the PDE problem. Generally, discontinuities are represented as boundaries to the domain Ω on which boundary conditions are applied. In addition to using a mesh conformal to the discontinuities with FEM shape functions as in DSI (Section 1.3.1.6, p.38), other techniques exist to introduce a discontinuous jump in the physical behavior approximated by the function u.

2.1.4.1 Optic criteria on compact supports

The visibility criterion is introduced in Belytschko *et al.* (1994b) with the same intention as the generation of a conformal mesh but with the local supports of weight functions. It is used on MLS and PUM functions. The idea is to separate the two sides of a discontinuity by preventing the nodes on either side to influence each other. The principle is as follows: each node $\overline{\boldsymbol{p}}$ emits a light and the discontinuities are opaque to this light. For any point $\boldsymbol{p}$, if a discontinuity happens to intersect a ray of light coming from a node $\overline{\boldsymbol{p}}$, then $\overline{\boldsymbol{p}}$ is not considered as a neighbor to $\boldsymbol{p}$. As in conformal meshes (Figure 1.15), the local supports $\omega_{\overline{\boldsymbol{p}}}$ are modified by the discontinuity's geometry (Figure 2.2(a)), but here, no nodes are duplicated.

The issue with the visibility criterion is that the function u behaves badly at the tip of the discontinuities. To solve this issue, Organ *et al.* (1996) created the diffraction and the transparency methods. They both keep the same concept, but the discontinuity becomes less and less opaque when getting closer to a tip. This dependency is controlled by overestimating the initial distance r between a node $\overline{\boldsymbol{p}}$ and a point $\boldsymbol{p}$ on either side of a discontinuity as

$$r^{trpcy} = r + \rho \left(\frac{r_{tip}}{r_{tot}} \right)^{\gamma}, \ \gamma \geq 2, \tag{2.47}$$

with r^{trpcy} the re-evaluated distance, ρ the influence distance of $\omega_{\overline{\boldsymbol{p}}}$, r_{tip} the distance between the intersection with the discontinuity and the tip, r_{tot} the distance to the tip where the discontinuity is completely opaque, and γ an empirically chosen parameter. By using r^{trpcy} in the weight functions, the influence of a node on the other side of the discontinuity is reduced when constructing the MLS or PUM functions. The distances and the effect of this method on the support's geometry are represented on Figure 2.2(b).

2.1.4.2 By enrichment of the basis functions

Moës *et al.* (1999) introduced a method to handle crack growth issues with FEM shape functions and without having to remesh on the crack. This method was later called Extended FEM (Moës & Belytschko, 2002) and generalized for other bases functions than the FEM functions (Belytschko

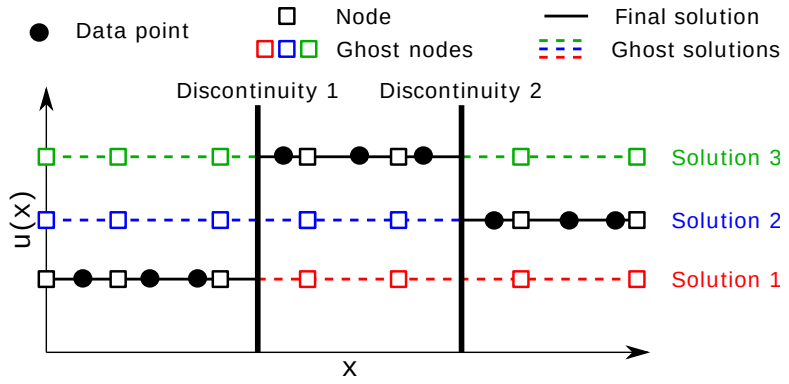


Figure 2.3: *Illustration of the ghost fluid method and the use of basis and ghost nodes when evaluating u. The final solution is discontinuous only at each discontinuity's position where it jumps from one solution to another.*

et al., 2001). In this approach, a node $\overline{\boldsymbol{p}}$ close to a discontinuity F is enriched by a jump function κ_F and an additional unknown $b_{\overline{\boldsymbol{p}}}$ as

$$u(\boldsymbol{x}) = \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}} \Phi_{\overline{\boldsymbol{p}}}(\boldsymbol{x})\ u_{\overline{\boldsymbol{p}}} + \sum_{F} \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}(F)} \kappa_F(\boldsymbol{x})\ \Phi_{\overline{\boldsymbol{p}}}(\boldsymbol{x})\ b_{\overline{\boldsymbol{p}}}, \tag{2.48}$$

where $\boldsymbol{N}(F)$ is the set of nodes close to the discontinuity F, $\boldsymbol{N}$ is the global set of nodes (including the nodes in $\boldsymbol{N}(F), \forall F$), and κ_F is equal to 1 on one side of F and -1 on the other side. In addition, they introduce a discontinuity crack tip function in 2D to ensure the continuity at the crack tip. This function has yet to be defined in 3D at the termination of discontinuity surfaces. When several discontinuities are intersecting, a mapping is used to handle the connectivity.

The same method was introduced by Fedkiw *et al.* (1999) as the ghost fluid method to handle interfaces between fluid flows. In this approach, ghost nodes are artificially added for each fluid on top of all the basis nodes. These new nodes are associated with an unknown coefficient and a shape function, as in XFEM. When evaluating u, a regular node or one of its associated ghost nodes is used depending on the existing fluid at the position of evaluation. Therefore, this is equivalent to computing, everywhere in the domain, a solution for each fluid separately and then to jump from one solution to another at the interfaces between fluids (Figure 2.3). This is similar to the XFEM approach but only the nodes close to the discontinuities are concerned by the enrichment in XFEM. The termination of discontinuities are not discussed in the paper.

2.1.5 Examples of numerical methods and applications

By combining the presented test and shape functions, many numerical methods were created over the past forty years. Some examples of numerical methods are classified in Table 2.2 using this principle. Additional legends are used for methods with a more complex definition. The blank spaces represent combinations of test and shape functions not yet implemented as a method, or unknown by the author. Noticeable methods with fields of applications are summarized in the following.

Abbreviations and reference papers for Table 2.2

BEM	Boundary Element Method	(Crouch, 1983)
BNM	Boundary Node Method	(Mukherjee & Mukherjee, 1997)
BPIM	Boundary Polynomial Interpolation Method	(Gu & Liu, 2001a)
BRPIM	Boundary Radial Polynomial Interpolation Method	(Gu & Liu, 2001a)
CRBF	Compactly supported Radial Basis Functions	(Wendland, 1995)
DEM	Diffuse Element Method	(Nayroles *et al.*, 1992)
EFG	Element Free Galerkin	(Belytschko *et al.*, 1994b)
FEM	Finite Element Method	(Dhatt & Touzot, 1981)
FPM	Finite Point Method	(Onate *et al.*, 1996a)
GFEM	Generalized Finite Element Method	(Strouboulis *et al.*, 2000)
GMLS	Generalized Moving Least Squares	(Atluri *et al.*, 1999a)
HBPIM	Hybrid Boundary Polynomial Method	(Gu & Liu, 2002)
HBRPIM	Hybrid Boundary Radial Polynomial Method	(Gu & Liu, 2002)
LBIE	Local Boundary Integral Equation	(Atluri & Shen, 2002)
LPIM	Local Polynomial Interpolation Method	(Liu & Gu, 2001a)
LSMM	Least-Squares collocation Meshless Method	(Zhang *et al.*, 2001)
LRPIM	Local Radial Polynomial Interpolation Method	(Liu & Gu, 2001b)
MFEM	Meshless Finite Element Method	(Idelsohn *et al.*, 2003)
MFSo	Method of Fundamental Solutions	(Kupradze, 1964)
MFSp	Method of Finite Spheres	(De & Bathe, 2000)
MLPG	Meshless Local Petrov-Galerkin	(Atluri & Shen, 2002)
MLS	Moving Least Squares	(Lancaster & Salkauskas, 1981)
MLSPH	Moving Least Squares Particle Hydrodynamics	(Dilts, 1999)
MPS	Method of Particular Solutions	(Fox *et al.*, 1967)
MWS	Meshless Weak-Strong	(Liu & Gu, 2003)
NEM	Natural Element Method	(Sukumar, 1998)
PIM	Polynomial Interpolation Method	(Liu & Gu, 2001c)
PPCM	Polynomial Point Collocation Method	(Liu & Gu, 2005)
PUFEM	Partition of Unity Finite Element Method	(Melenk & Babuška, 1996)
PUM	Partition of Unity Method	(Babuška & Melenk, 1997)
RBF	Radial Basis Functions	(Buhmann, 2000)
RKPM	Reproducing Kernel Particle Method	(Liu *et al.*, 1995)
RPCM	Radial Point Collocation Method	(Liu & Gu, 2005)
RPIM	Radial Polynomial Interpolation Method	(Wang & Liu, 2002)
WLS	Weighted Least Squares	(Liu & Gu, 2005)
XFEM	Extended Finite Element Method	(Belytschko *et al.*, 2001)

Table 2.2: *Classification of the existing numerical methods based on the chosen test and shape functions*

		Test function											
		Collocation		Least-Square Residus		Bubnov-Galerkin		Weight functions		Heaviside Step function		Fundamental Solutions	
shape function	FEM					FEM						BEM	
	NEM					MFEM, NEM							
	RBF, CRBF	MLPG2, MFSp collocation			MLPG3		MLPG6, MFSp		MLPG1		MLPG5		LBIE
	Fundamental Solutions	MFSo, MPS											
	PIM, WLS					PIM, WLS			LPIM			BPIM, HBPIM	
	RPIM	MWS-RPIM			MLPG3	RPIM	MLPG6		LRPIM, MWS-RPIM		MLPG5	BRPIM, HBRPIM	LBIE
	MLS, RKPM	MLSPH, FPM, MLPG2, MWS-MLS		LSMM	MLPG3	DEM, EFG	MLPG6, MFSp		MLPG1, MWS-MLS		MLPG5	BNM	LBIE
	PUM				MLPG3	PUFEM, GFEM, XFEM	MLPG6, MFSp						
	hp-clouds	hp-clouds			hp-clouds								
	Hermite-type RPIM	RPCM							LRPIM				
	Hermite-type PIM / WLS, GMLS,	PPCM							LPIM				

No regards to shape functions
Point dependent test functions

Global weak form	Local weak form

Finite Element Method (FEM)

The conventional form of FEM uses the FEM shape functions in a Bubnov-Galerkin scheme: the FEM functions are also used as test functions, and the integral is approximated with quadrature rules on each element (Dhatt & Touzot, 1981). Being the most famous numerical method, FEM has been successfully applied to almost all fields of physics.

Element Free Galerkin (EFG)

Another widespread method is EFG which uses a Bubnov-Galerkin scheme with MLS functions. It was introduced in (Belytschko *et al.*, 1994b) as a more stable version than the Diffuse Element Method (DEM) (Nayroles *et al.*, 1992). In both methods, the MLS nodes are centered on the data points, but the difference is that the quadrature rules for integration are performed on the nodes themselves in DEM while they are performed on a background grid in EFG. The derivatives of the MLS functions are also more accurate in EFG. This method is popular in crack growth and fracture mechanics (Belytschko *et al.*, 1994a; Krysl & Belytschko, 1999; Khazal *et al.*, 2016). It was also applied in other fields of physics such as solid mechanics (Belytschko *et al.*, 1994b; Chen & Guo, 2001) and electromagnetism (Cingoski *et al.*, 1998).

Boundary Element Method (BEM)

BEM is a special case of a Petrov-Galerkin scheme where test and shape functions are only defined on the boundaries by solving the Boundary Integral Equation. The FEM functions are generally used as shape functions (i.e., the boundaries are meshed), and the fundamental solutions of a given PDE are used as test functions. Therefore, the problem's dimensionality is reduced by one, but this can only be applied when fundamental solutions are available for the studied physical problem. BEM has been successfully applied in many fields, such as geosciences (Morra *et al.* (2007), Maerten (2010) Parts 1 and 2).

Method of Fundamental Solutions (MFSo)

MFSo uses the fundamental solutions as shape functions to solve a physical problem (Kupradze, 1964). If these functions are undefined at their center node positions (called source singularities), a technique is to put those nodes outside of the domain of study (Chen, 1995). As the PDE is automatically solved everywhere in the studied area, only the boundary conditions are written. In Table 2.2, we classify MFSo into collocation methods as no integral is computed for solving the PDE, but this is not actually accurate as the strong form is not explicitly written. Note that integrals can still be computed on the boundary conditions. MFSo was applied in many fields of physics (Chen *et al.*, 2008), such as solid mechanics (Poullikkas *et al.*, 2002; Berger & Karageorghis, 2001; Chen *et al.*, 2006), fluid mechanics (Boag *et al.*, 1988) and electromagnetism (Kress *et al.*, 1986).

Meshless Local Petrov-Galerkin (MLPG)

As presented in Section 2.1.2.3, MLPG are methods using a Petrov-Galerkin scheme, generally with meshless shape and test functions, and using the local weak form. Six MLPG techniques are presented in Atluri & Shen (2002) with their names enumerated from MLPG1 to MLPG6. They are differentiated by the test function they use (Table 2.2). A special case is the MLPG4 which is usually called Local Boundary Integral Equation (LBIE) to show the similitude with BEM. Many different applications of MLPG exist in physics (Sladek *et al.*, 2013), such as applications in solid mechanics (Gu & Liu, 2001b), fracture mechanics (Ching & Batra, 2001), and fluid mechanics (Wu *et al.*, 2005b).

Meshless Weak-Strong methods (MWS)

Liu & Gu (2003) introduce a mix of Collocation and Petrov-Galerkin schemes to solve a problem. The idea is to benefit from the computational efficiency of the Collocation scheme while solving their limited ability to enforce Neumann Boundary Conditions. This is done by using a Dirac test function

on all the nodes except for the nodes with NBC where the technique of MLPG1 is used. This has been applied with RPIM and MLS functions on solid and fluid mechanics problems (Liu & Gu, 2005).

2.1.6 Discussion on mesh-based and meshless methods

Controversial notion of meshless

The success of meshless methods in the past decades and the mystery existing around them for non-specialists have created a popular classification: a numerical method is perceived either as meshless or mesh-based. In addition, the rising interest in the industry for meshless sometimes creates the illusion that a method is better for the unique reason that it is considered as meshless. In opposition, using the word *meshless* to describe an approach can also be seen as an unnecessary argument of authority.

From this context, a recurrent question emerges: is the presented method meshless? For most authors in numerical methods, this is not the right question (Idelsohn & Onate, 2006). The main point is to solve efficiently a PDE with its boundary conditions. Therefore, adapted techniques to solve a given application problem are generally sought for regardless of their origin in the scientific communities. Knowing that, we present in the following the notion of a *truly* meshless method, which can be found in the early days of the meshless community.

Notion of a *truly* meshless method

As its name indicates, a *truly* meshless method is a method that never uses a mesh. The test functions, the shape functions, and the integration do not require the structure of a mesh (Atluri *et al.*, 1999b). Although this is true, the discontinuities with complex geometries can be meshed to delimit the boundaries of the domain. Some examples of *truly* meshless methods are all the techniques using a Collocation scheme with meshless shape functions, all the MLPG methods as described in Atluri & Shen (2002) and Bubnov-Galerkin methods using meshless functions and no background grid (e.g., DEM).

Why using a *non-truly* meshless method?

Using meshless shape functions with a background mesh could seem fruitless since meshless functions are generally more computationally demanding than mesh-based ones. If a mesh is created for integration, it seems wise to also use it to create the shape functions. In fact, meshless shape functions generally have a higher interpolation accuracy: Belytschko *et al.* (1996b) and Liu & Gu (2005) show smaller error rates than FEM for a same resolution in function and surface fitting applications.

The other question concerns the computational cost. Liu & Gu (2005) obtain more accurate results and better convergence rates with BG methods using meshless shape functions than FEM for the same resolution and the same order of consistency. On the other hand, FEM claims much faster algorithms. To compare meshless and mesh-based methods, Liu and Gu define the efficiency of a method by the computational cost *vs.* the accuracy. They obtain that BG methods are globally more efficient than FEM, but it depends on the field of application (solid mechanics here) and the complexity of the study.

As presented in Section 2.1.4, meshless techniques offer other possibilities to handle discontinuities and to refine a solution than adapting the mesh geometry and topology. PUFEM, GFEM, XFEM and hp-clouds were first defined with FEM functions for these reasons, but they are not restricted to this scheme anymore. Also, if shape functions can handle discontinuities, the background mesh of a BG scheme do not have to be conformal to structural heterogeneities to perform the integration. Therefore, simple meshes, such as regular ones, are common in BG schemes using meshless shape functions.

A meshing or a sampling problem for discretization

The creation of a complete cover with local meshless supports is a problem of discretization, just like meshing (Figure 2.4). The only difference is that the topology between the nodes is ensured with potentially overlapping areas of influence, and not stored relationships of edges and faces. The studied area can be sampled regularly, on the data, or in any other way that eases the algorithm (e.g.,

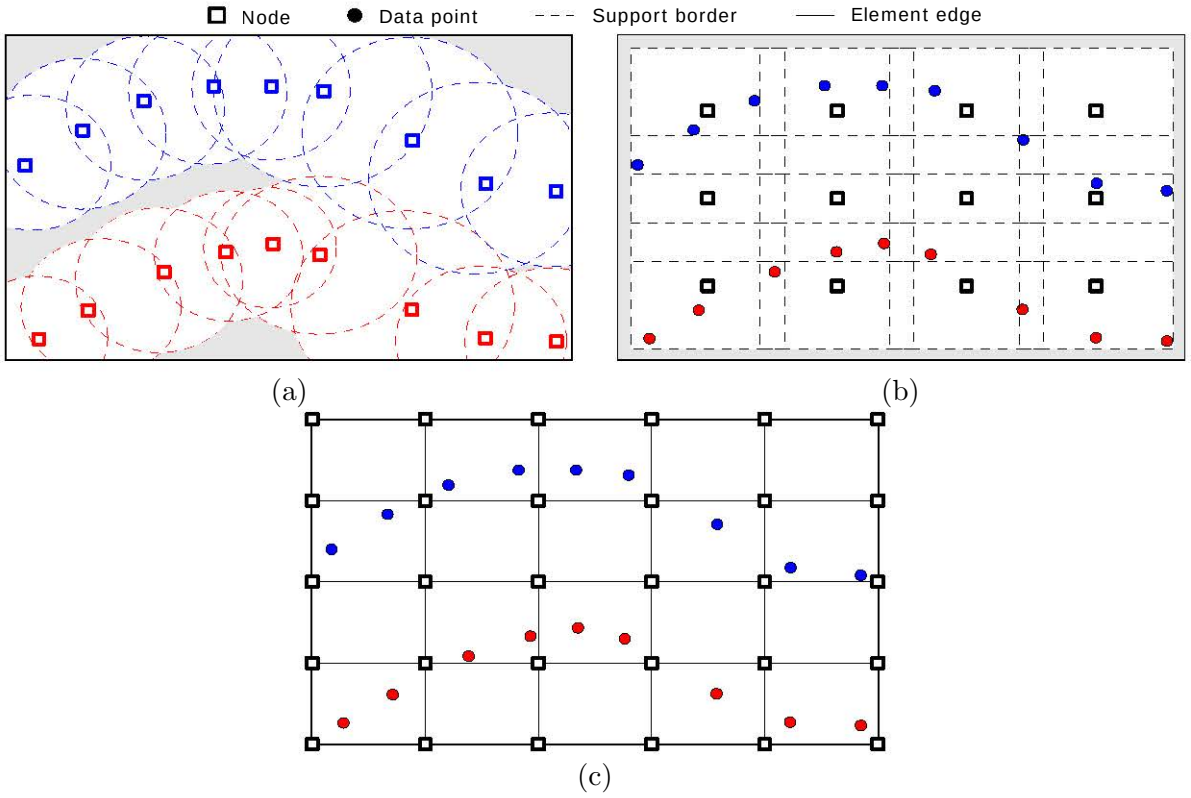


Figure 2.4: *Three ways to structure numerical methods. (a) and (b) Two possible sampling techniques for meshless compact supports, and (c) The regular grid support for FEM shape functions. While ensuring a complete cover in (b) is straightforward as compared to (a) (i.e., where the support lengths are difficult to tune), (b) is getting closer and closer to (c).*

randomly, with trees, or else). When discontinuities are involved in the study, the generation of a potentially complex mesh is avoided, but a problem of sampling remains.

Used terminology in the thesis

The presented methods that do not use Finite Element shape functions are stated as meshless methods. The techniques used to enrich a Finite Element basis functions to reduce the constraints on the mesh are stated as mesh reduction methods. We do not consider these methods as *truly* meshless as the discretization and integration steps are performed on an underlying grid. The goal in this thesis is to investigate new possibilities to solve the structural modeling problem.

2.1.7 Application to implicit structural modeling

2.1.7.1 Positioning DSI and PFM

DSI (Section 1.3.1, p.34) uses the FEM shape functions to define the implicit function u. No PDE or boundary conditions are explicitly written, and therefore no test functions or integration methods are formally defined.

PFM (Section 1.3.2, p.40) uses smooth RBF as shape functions to define the implicit function u. In the specific case of smooth multiquadric RBF and Thin Plate Splines (Section 2.1.3.3), PFM can be considered as a Method of Fundamental Solutions (Section 2.1.5) for the biharmonic equation with in-bound singularity nodes. The technique to handle the discontinuities with jumping drifts (Section 1.3.2.7, p.46) could also be compared to PUMs techniques of discontinuous enrichments of the basis functions (Section 2.1.4.2).

2.1.7.2 Opportunities

The positioning of DSI and PFM tells us that the existence of a PDE for the structural modeling application needs to be discussed, which is the subject of Section 2.2. In the case where the underlying problem of structural modeling is defined as a continuous one, some numerical methods are more adapted than others to solve the problem.

A difficulty with PFM is its dense system of equations which limits the algorithm to a restrictive number of numerical data (Section 1.3.2.10, p.49). This is caused by the chosen interpolation functions which are generally represented by RBF or CRBF with large supports (Section 1.3.2.5, p.43). We therefore consider functions with small compact supports to create sparse systems of equations.

The stratigraphic function is assumed strictly continuous everywhere in the domain of study except at the discontinuity surfaces (Section 1.2.2, p.31). Therefore, the chosen interpolation functions must be compatible (Section 2.1.3.1), so we do not consider the locally defined PIM, WLS and RPIM shape functions.

Polynomial functions have proven to be a good approximation to represent geological structures: piecewise linear mesh elements are typically used in DSI (Section 1.3.1.3, p.35) and a polynomial drift is used in PFM to stabilize the interpolation (Section 1.3.2.3, p.42). Such approaches build an implicit function with a first order consistency (Section 2.1.3.1). We also consider bases functions with at least a first order consistency, so RBF and CRBF are not considered if not enriched with a polynomial term.

FEM, MLS and PUM functions seem particularly adapted as they meet all the previously mentioned conditions. The problem with MLS and PUM functions is to ensure a complete cover, with overlapping subdomains. This may prove difficult with the data irregularity in structural modeling (Section 1.1.4.5, p.26). A sampling method on other positions than data points (Figure 2.4(b)) may be considered with these bases functions. As scalability is a concern, MLS and FEM functions may be preferred over PUM functions which involve much more unknowns in the final system of equations.

Concerning the test functions and the integration, as no boundary conditions are clearly defined in structural modeling, the collocation method seems the most appropriate. If the background sampling is structured enough, a Bubnov-Galerkin scheme with quadrature rules may also be considered.

Finally, the main difficulty encountered by both DSI and PFM is the handling of faults (i.e., discontinuities). Optic criteria are particularly adapted to meshless bases functions with compact supports such as MLS functions (Section 2.1.4.1). The XFEM approach is also adapted to local methods, defining the jump functions on the compact supports of interpolation (i.e., as introduced with the FEM functions in Section 2.1.4.2). Depending on the chosen interpolation functions, these techniques may be considered.

In conclusion, we have narrowed the scope of opportunities to two bases shape functions, two test functions, and two techniques to handle the discontinuities. In the following, a continuous problem is introduced for structural modeling (Section 2.2), and two discretizations of this problem are suggested using two different combinations of the discussed functions and techniques (Chapters 3, p.91 and 4, p.119).

2.2 Application to implicit structural modeling

In this section, we propose a continuous problem for implicit structural modeling (Section 1.2, p.29) to benefit from the numerical methods presented in Section 2.1. After discussing the existence of a Partial Differential Equation (PDE) for structural modeling, the problem is posed as a minimization of generic functionals and it is adapted to classical data constraints and structural discontinuities.

2.2.1 Lack of a strong formulation in implicit structural modeling

Geological structures are the result of multiple phenomena that occurred during a long period of time, and that involved unquantifiable variables. Ideally, reproducing such structures would require to capture the physical principles that have driven all the geological processes over time. Only portions of the current structures and physical properties are accessible: geologists and geophysicists do not have access to the former structures and properties, although they may have hypotheses and models.

Therefore, structural modeling algorithms *only* aim at assessing the geometries of the current geological structures based on the observation.

In implicit structural modeling, the stratigraphic function (Section 1.2.2, p.31) must handle the heterogeneous data (Section 1.1.4, p.26) while reproducing complex geometries. These geometries must follow some geological rules based on analog structures. Finding a PDE solving this modeling problem hence requires to translate empirical rules into a physical behavior. Unfortunately, except for the validity conditions given in Caumon (2009) (Section 1.1.5, p.27), these rules generally involve a non-negligible part of subjectivity from the expert, which is difficult to describe mathematically.

The most common rule in implicit structural modeling is to expect smooth stratigraphic surfaces (as in the Discrete Smooth Interpolation (DSI, Section 1.3.1.4) and in the Potential Field Method (PFM, Section 1.3.2.5)). This may be a controversial simplification, but it presents many practical advantages (Section 1.3.3, p.50). Several PDEs can be proposed to create an implicit function with smooth iso-surfaces. For instance, the harmonic Poisson equation could be suggested as in unique surface construction (Section 1.5.3.1, p.55), but it requires a vector field of the aimed structures as input, which may be as difficult to create as the implicit function.

A PFM scheme using smooth multiquadric RBF or Thin Plate Splines intrinsically solves the biharmonic equation (Section 2.1.3.3). The strong form of the PDE is solved without boundary conditions; the data points may be considered as in-bound boundary conditions (as in Briggs (1974)), but this is not standard. This approach is convenient as boundary conditions (on the borders of the domain of study and the discontinuities) are unknown in most geoscience problems. For instance, inverse methods are used to estimate geomechanical constraints on boundary surfaces such as faults (Maerten (2010) Part 2 for the slip and stress) and domain boundaries (Mazuyer (2018) for the stress). In structural modeling as well, the boundaries should not have any influence on the structures and be treated as free borders. The problem is that a PDE should be defined together with boundary conditions to ensure a unique solution.

Posing a modeling problem as a PDE without boundary conditions seems mathematically incorrect. We therefore suggest to use the concept of energy that does not require boundary conditions and can penalize the fitting of the data constraints. This naturally leads to a weak formulation by using the Euler-Lagrange equations of the calculus of variations, which can be solved using the numerical methods presented in Section 2.1. Appendix B (p.166) discusses the equivalence between the PDE and the energy approaches in the specific case of the biharmonic equation and the bending energy.

2.2.2 The proposed generic framework

The modeling problem is posed as a minimization of a set of functionals as

$$\min_u \left(J(u)\right) = \min_u \left(J_E(u) + J_{data}(u)\right), \tag{2.49}$$

with J_{data} the set of functionals associated to the data constraints, J_E the set of functionals of continuous energies, and u the implicit function defined in the domain of study Ω (i.e., $u = u(\boldsymbol{x})$, $\forall \boldsymbol{x} \in \Omega$).

In this formalism, several types of data constraints and several energies can be used all together (i.e., the number of functionals is arbitrary). The next section deals with the standard setting used in this thesis, which is defined with the data points value constraints and the bending energy.

2.2.3 The standard framework

2.2.3.1 A spatial regression of data points

Data points value (DPV) constraints

The DPV constraints concern the sets $\boldsymbol{D}(H)$ of data points $\boldsymbol{p}$ associated to a horizon H and an already known iso-value α_H. This constraint is presented in Section 1.3.1.2 and its equation (Equation (1.5), p.35) is repreated here:

$$\forall H,\ \forall \boldsymbol{p} \in \boldsymbol{D}(H), \quad u(\boldsymbol{p}) = \alpha_H. \tag{2.50}$$

In the standard setting, the global set of data $\boldsymbol{D}$ only involves DPV constraints (i.e., $\boldsymbol{D} = \cup \boldsymbol{D}(H)$, $\forall H$). Also, we suggest for simplicity to assume implicitly the association between a data point $\boldsymbol{p}$ and its

horizon H. It is then assumed that the enforced iso-value on a data point $\alpha_{\boldsymbol{p}}$ is the iso-value of its horizon α_H. The DPV constraint is then written as

$$\forall \boldsymbol{p} \in \boldsymbol{D}, \quad u(\boldsymbol{p}) = \alpha_{\boldsymbol{p}}. \tag{2.51}$$

Functional of the DPV constraints

Honoring all these constraints at once is equivalent to a spatial regression of data points with their associated values, minimizing the functional J_{DPV} written as

$$J_{DPV}(u) = \frac{1}{2} \sum_{\boldsymbol{p} \in \boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \, (u(\boldsymbol{p}) - \alpha_{\boldsymbol{p}})^2, \tag{2.52}$$

where $\lambda_{\boldsymbol{p}}$ are the constant weights associated to each DPV constraint.

2.2.3.2 The bending energy

Definition

The bending energy (or thin plate energy) (Dubrule, 1984; Wahba, 1990) stands as a way to control the global curvature of a thin plate being bended to fit spatially distributed constraints (e.g., tractions or fix point positions). Depending on its rigidity, the plate may resist to the efforts it endures. A rigid material hence creates a smooth plate averaging the constraints, and a soft material creates a plate closely accommodating all the constraints. In implicit structural modeling, this energy smooths globally the stratigraphic function and its iso-surfaces. The relation between thin plate curvature and smooth stratigraphic function is illustrated and further discussed in Section 5.3.1 (p.137).

We suggest this energy as the referring energy in the standard formulation of implicit structural modeling as it is a well known smoothing energy. The assumption of smoothness being made, the same advantages and limits of smoothing as presented in the existing methods are expected (Section 1.3.3, p.50).

Functional of the bending energy

The functional of the bending energy J_ϵ corresponds to

$$J_\epsilon(u) = \frac{1}{2} \int_\Omega \lambda_\epsilon^2 \left(\partial_{ij}^2 u\right)^2 d\Omega, \tag{2.53}$$

with λ_ϵ the penalization function of the bending energy assumed constant for all terms over the domain Ω, with $\partial_{ij}^2 = \frac{\partial^2}{\partial ij}$, and with a sum assumed on i and j for each coordinate axis. This is equivalent to

$$J_\epsilon(u) = \begin{cases} \frac{1}{2} \int_\Omega \lambda_\epsilon^2 \left(\partial_{xx}^2 u\right)^2 d\Omega & in\ 1D, \\ \frac{1}{2} \int_\Omega \lambda_\epsilon^2 \left(\left(\partial_{xx}^2 u\right)^2 + \left(\partial_{yy}^2 u\right)^2 + 2 \left(\partial_{xy}^2 u\right)^2 \right) d\Omega & in\ 2D, \\ \frac{1}{2} \int_\Omega \lambda_\epsilon^2 \left(\left(\partial_{xx}^2 u\right)^2 + \left(\partial_{yy}^2 u\right)^2 + \left(\partial_{zz}^2 u\right)^2 + 2 \left(\partial_{xy}^2 u\right)^2 + 2 \left(\partial_{xz}^2 u\right)^2 + 2 \left(\partial_{yz}^2 u\right)^2 \right) d\Omega & in\ 3D. \end{cases} \tag{2.54}$$

2.2.3.3 A spatial regression of data points penalized by the bending energy

Standard minimization problem

In the standard setting of the proposed framework, the DPV functional J_{DPV} and the bending energy functional J_ϵ are minimized as

$$\begin{aligned} \min_u \left(J(u)\right) &= \min_u \left(J_{DPV}(u) + J_\epsilon(u)\right) \\ &= \min_u \left(\frac{1}{2} \int_\Omega \lambda_\epsilon^2 \left(\partial_{ij}^2 u\right)^2 d\Omega + \frac{1}{2} \sum_{\boldsymbol{p} \in \boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \, (u(\boldsymbol{p}) - \alpha_{\boldsymbol{p}})^2 \right). \end{aligned} \tag{2.55}$$

The constants $\lambda_{\boldsymbol{p}}$ and λ_ϵ control the influence of each constraint over the minimization problem. Globally increasing the $\lambda_{\boldsymbol{p}}$ values as compared to λ_ϵ will theoretically improve the fitting between iso-surfaces and data points. In opposition, increasing the λ_ϵ value as compared to the $\lambda_{\boldsymbol{p}}$ values will theoretically smooth the iso-surfaces and deteriorate the fit to the data points. This concept will be tested and illustrated with the numerical methods proposed in this thesis (Chapters 3, p.91 and 4, p.119).

Weak formulation using the Euler-Lagrange equations

Applying the Euler-Lagrange equations on the standard minimization problem (Equation (2.55)) gives a weak formulation as defined in Section 2.1.1.1. Solving the implicit structural modeling problem is then equivalent to finding a function u satisfying

$$\forall v, \quad \int_\Omega \lambda_\epsilon^2 \ \left(\partial_{ij}^2 u \ \partial_{ij}^2 v\right) d\Omega + \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ u(\boldsymbol{p}) \ v(\boldsymbol{p}) = \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ \alpha_{\boldsymbol{p}} \ v(\boldsymbol{p}), \tag{2.56}$$

where v is an arbitrary test function.

This weak formulation involves both continuous and discrete operators, as well as a discretely known right hand function (i.e., $\int_\Omega v(\boldsymbol{x}) \ f(\boldsymbol{x}) d\Omega = \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ \alpha_{\boldsymbol{p}} \ v(\boldsymbol{p})$). The details of the Euler-Lagrange developments on the data and energy terms are given in Appendix C (p.167).

2.2.4 Formulation with generic bases functions and domain discretization

2.2.4.1 Test and shape functions

As presented in Section 2.1.1.2, the implicit function u is taken in the space U (Equation (2.5), p.60) and the test function v is taken in the space V (Equation (2.6), p.60). Replacing the discrete functions u and v in the weak formulation of the problem (Equation (2.56)) gives the linear system

$$\left(\int_\Omega \lambda_\epsilon^2 \left(\partial_{ij}^2 \boldsymbol{\Psi} \cdot \partial_{ij}^2 \boldsymbol{\Phi}^T\right) d\Omega + \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ \left(\boldsymbol{\Psi}(\boldsymbol{p}) \cdot \boldsymbol{\Phi}^T(\boldsymbol{p})\right) \right) \cdot \boldsymbol{U} = \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ \alpha_{\boldsymbol{p}} \ \boldsymbol{\Psi}(\boldsymbol{p}), \tag{2.57}$$

with $\boldsymbol{\Phi} = \{\Phi_1, \ldots, \Phi_{N_U}\}$ the basis shape functions, $\boldsymbol{\Psi} = \{\Psi_1, \ldots, \Psi_{N_V}\}$ the basis test functions, and $\boldsymbol{U} = \{u_1, \ldots, u_{N_U}\}$ the unknowns of the problem. The details of the development are given in Appendix D (p.168).

2.2.4.2 Domain discretization

As presented in Section 2.1.1.2, depending on the chosen shape and test functions, the domain Ω may be subdivided into $\boldsymbol{\Omega}$ subdomains. In the case where the subdomains do not overlap, and their union covers exactly the domain Ω, the linear system (2.57) can be discretized as

$$\left(\sum_{\omega\in\boldsymbol{\Omega}} \int_\omega \lambda_\epsilon^2 \left(\partial_{ij}^2 \boldsymbol{\Psi} \cdot \partial_{ij}^2 \boldsymbol{\Phi}^T\right) d\omega + \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ \left(\boldsymbol{\Psi}(\boldsymbol{p}) \cdot \boldsymbol{\Phi}^T(\boldsymbol{p})\right) \right) \cdot \boldsymbol{U} = \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ \alpha_{\boldsymbol{p}} \ \boldsymbol{\Psi}(\boldsymbol{p}). \tag{2.58}$$

2.2.4.3 General system

The general system (2.58) can be written on all the subdomains as

$$\left\{ \left(\sum_{\omega\in\boldsymbol{\Omega}} \int_\omega \lambda_\epsilon^2 \left(\partial_{ij}^2 \Psi_k(\boldsymbol{x}) \cdot \partial_{ij}^2 \Phi_l(\boldsymbol{x})\right) d\omega + \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ \left(\Psi_k(\boldsymbol{p}) \cdot \Phi_l(\boldsymbol{p})\right) \right) \cdot \boldsymbol{U} = \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ \alpha_{\boldsymbol{p}} \ \Psi_k(\boldsymbol{p}), \quad \forall \Psi_k \in \boldsymbol{\Psi}, \ \forall \Phi_l \in \boldsymbol{\Phi} \right. , \tag{2.59}$$

which is equivalent to the general matrix form given in Equation (2.10) (i.e., $\boldsymbol{K} \cdot \boldsymbol{U} = \boldsymbol{F}$, p.61).

2.2.4.4 Least squares equivalence in a Bubnov-Galerkin scheme

Although it is not a requirement of the proposed framework, the Bubnov-Galerkin scheme (Section 2.1.2.2) is a special interest in this thesis. As a reminder, the Bubnov-Galerkin scheme takes the test function v in the same space U as the solution function u (i.e., $\boldsymbol{\Psi} \equiv \boldsymbol{\Phi}$). In this case, the general system (2.59) is a square system.

This system can also be written as a least squares system. The equivalence is explicit in the case of a quadrature of the integral terms with one point, as developed in Appendix E (p.169). The least squares system in the case of a nodal basis shape functions is written

$$\begin{cases} \lambda_\epsilon \sqrt{\nu_{\overline{\boldsymbol{p}}}} & \left(\partial^2_{ij} \boldsymbol{\Phi}(\overline{\boldsymbol{p}})^T \cdot \boldsymbol{U}\right) & = 0, & \forall \overline{\boldsymbol{p}} \in \boldsymbol{N}, \\ \lambda_{\boldsymbol{p}} & \left(\boldsymbol{\Phi}(\boldsymbol{p})^T \cdot \boldsymbol{U}\right) & = \lambda_{\boldsymbol{p}}\, \alpha_{\boldsymbol{p}}, & \forall \boldsymbol{p} \in \boldsymbol{D}, \end{cases} \tag{2.60}$$

with $\nu_{\overline{\boldsymbol{p}}}$ the volume of the subdomain $\Omega_{\overline{\boldsymbol{p}}}$ associated to the node $\overline{\boldsymbol{p}}$, and $\boldsymbol{N}$ the set of nodes which contains as many elements as the set of unknowns (i.e., $\#\boldsymbol{N} \equiv N_U$). Here, the sum is not assumed on i and j, so one row is written for each set of coordinates in adequacy with the dimensional space.

This formulation is convenient as each modeling constraint (i.e., data constraint and local approximation of the energy term) represents a separate row in the system. Adding a new DPV constraint, for instance, is equivalent to writing Equation (2.50) (with u in the space U) as an additional row in system (2.60).

2.2.5 Other data constraints

Theoretically, all the data constraints handled in DSI (Mallet (2014), Section 14.2) or PFM (Hillier *et al.*, 2014) can be used in the presented framework. The associated functional to each type of data is added to the global data functional J_{data} in the minimization problem (Equation (5.1)).

The principal types of constraints for structural modeling and their associated functionals are presented in this section. The approach developed above and in the Appendices C (p.167) to E (p.169) can be applied to these functionals. Also, the presented equations of constraints can be added as rows in the least squares system (2.60).

2.2.5.1 Increment data points (IDP)

The IDP constraints concern the sets $\boldsymbol{D}(I)$ of data points $\boldsymbol{p}$ associated to an increment horizon I for which the expected iso-value is unknown. This constraint is presented in Section 1.3.2.2 and its equation (Equation (1.13), p.42) is repeated here (with adapted notations):

$$\forall I,\ \forall (\boldsymbol{p}_1, \boldsymbol{p}_2) \in \boldsymbol{D}(I) \quad u(\boldsymbol{p}_1) - u(\boldsymbol{p}_2) = 0. \tag{2.61}$$

The functional J_{IDP} is written by fixing a reference point $\boldsymbol{p}_I$ per horizon I as

$$J_{IDP}(u) = \frac{1}{2} \sum_I \sum_{(\boldsymbol{p}, \boldsymbol{p}_I) \in \boldsymbol{D}(I)} \lambda^2_{\boldsymbol{p}} \left(u(\boldsymbol{p}) - u(\boldsymbol{p}_I)\right)^2. \tag{2.62}$$

2.2.5.2 Normal gradient data (NGD)

The NGD constraints concern the set $\boldsymbol{D}(G)$ of points $\boldsymbol{p}$ respectively representing the tail of a vector $\boldsymbol{g}$ perpendicular to the stratigraphy and with a norm $||\boldsymbol{g}||$. The shape functions must be at least once derivable at each point $\boldsymbol{p} \in \boldsymbol{D}(G)$. This constraint is presented in Section 1.3.2.2 and its equation (Equation (1.14), p.42) is repeated here:

$$\forall \boldsymbol{p} \in \boldsymbol{D}(G), \quad \nabla u(\boldsymbol{p}) = \boldsymbol{g}. \tag{2.63}$$

In matrix form, this is written (with $\partial_i = \frac{\partial}{\partial i}$) as

$$\begin{array}{lccc} & \textit{in } 1D & \textit{in } 2D & \textit{in } 3D \\ \forall \boldsymbol{p} \in \boldsymbol{D}(G), & \begin{bmatrix} \partial_x u(p_x) \end{bmatrix} = \begin{bmatrix} g_x \end{bmatrix}, & \begin{bmatrix} \partial_x u(p_x) \\ \partial_y u(p_y) \end{bmatrix} = \begin{bmatrix} g_x \\ g_y \end{bmatrix}, & \begin{bmatrix} \partial_x u(p_x) \\ \partial_y u(p_y) \\ \partial_z u(p_z) \end{bmatrix} = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix}. \end{array} \tag{2.64}$$

The functional J_{NGD} is written as

$$J_{NGD}(u) = \frac{1}{2} \sum_{\boldsymbol{p} \in \boldsymbol{D}(G)} \lambda_{\boldsymbol{p}}^2 \left(\nabla u(\boldsymbol{p}) - \boldsymbol{g} \right)^2 . \tag{2.65}$$

2.2.5.3 Tangent data (TD)

The TD constraints concern the set $\boldsymbol{D}(T)$ of points $\boldsymbol{p}$ respectively representing the tail of a vector $\boldsymbol{t}$ tangential to the stratigraphy. The shape functions must be at least once derivable at each point $\boldsymbol{p} \in \boldsymbol{D}(T)$. This constraint is also known as lineation constraint (Hillier *et al.*, 2014) or isoline constraint (Caumon *et al.*, 2013). It enforces the iso-surface passing by the point $\boldsymbol{p}$ to be parallel to the vector $\boldsymbol{t}$. In other terms, the gradient of the implicit function ∇u is constrained to be perpendicular to the tangent vector $\boldsymbol{t}$:

$$\nabla u(\boldsymbol{p}) \cdot \boldsymbol{t} = 0. \tag{2.66}$$

Contrarily to the normal gradient data, the norm $||\boldsymbol{t}||$ does not influence the norm of the implicit function's gradient, but it does influence the system of equations as a weighting factor. In 3D, two tangent vectors in the stratigraphic plane are necessary for this constraint (i.e., involving two equations).

The functional J_{TD} is written as

$$J_{TD}(u) = \frac{1}{2} \sum_{\boldsymbol{p} \in \boldsymbol{D}(T)} \lambda_{\boldsymbol{p}}^2 \left(\nabla u(\boldsymbol{p}) \cdot \boldsymbol{t} \right)^2 . \tag{2.67}$$

2.2.5.4 Polygonal lines (PL) and surfaces (PS)

The PL and PS constraints concern the set $\boldsymbol{D}(\Gamma)$ of any type of polygonal lines or surfaces. It enforces an iso-surface in a 2D (*resp.* 3D) problem to superpose a polygonal line (*resp.* surface) Γ. In the following, we develop the concept for polygonal lines, but it also applies to polygonal surfaces.

If the iso-value α_Γ of the expected 2D iso-surface is known (as in a DPV constraint), the polygonal line value (PLV) constraint corresponds to

$$\forall \Gamma, \quad \int_\Gamma \left(u(\boldsymbol{x}) - \alpha_\Gamma \right) d\Gamma = 0. \tag{2.68}$$

In the case where this iso-value is unknown (as in an IDP constraint), the polygonal line increment (PLI) constraint corresponds to

$$\forall \Gamma, \quad \int_\Gamma \left(u(\boldsymbol{x}) - u(\boldsymbol{x}_\Gamma) \right) d\Gamma = 0, \tag{2.69}$$

where $\boldsymbol{x}_\Gamma$ is a point indicating the reference iso-value of the line Γ. The point $\boldsymbol{x}_\Gamma$ may or may not be chosen within a given line Γ (e.g., a horizon defined with increment data may involve several lines and points).

The functionals J_{PLV} and J_{PLI} are respectively written as

$$J_{PLV}(u) = \frac{1}{2} \sum_\Gamma \lambda_\Gamma^2 \left(\int_\Gamma \left(u(\boldsymbol{x}) - \alpha_\Gamma \right)^2 d\Gamma \right)^2 , \tag{2.70}$$

and

$$J_{PLI}(u) = \frac{1}{2} \sum_\Gamma \lambda_\Gamma^2 \left(\int_\Gamma \left(u(\boldsymbol{x}) - u(\boldsymbol{x}_\Gamma) \right)^2 d\Gamma \right)^2 . \tag{2.71}$$

All of these integrals can be approximated with quadrature rules per polygonal element, reducing these two constraints to an equivalent set of DPV and IDP constraints. However, the polygonal elements give more information than just their position in space, they also give their local orientation. Therefore, it is also possible to locally extract tangent and normal vectors from these polygonal elements leading to additional TD constraints and, if the elements are polarized, NGD constraints. The polarization must be in the same direction as other gradient data (generally in the younging direction of the stratigraphic layers). Using both TD and NGD constraints obtained from a line or a surface is redundant.

2.2.5.5 Hard data points (HDP)

As presented in Sections 1.3.1.5 (p.37) and 1.3.2.6 (p.44), the different reliabilities in the numerical data (Section 1.1.4, p.26) are handled by distinguishing two types of data: the soft and hard data. In general, special treatments are only applied to hard data to enforce the corresponding constraints perfectly while the soft data are constrained as presented previously. This section deals with possible techniques to enforce hard data in the case of DPV constraints for simplicity, but the presented principles can be adapted to the other constraints.

Tuning the data weights

Modifying the weight values λ_ϵ and $\lambda_{\boldsymbol{p}}$ in the minimization problem (Equation (2.55)) controls the smoothness of the created iso-surfaces and the fit to the data points. In the case of soft and hard data points, a common approach is to give an average weight λ^{soft} to all soft data $\boldsymbol{D}(soft)$, and a greater weight λ^{hard} to all hard data $\boldsymbol{D}(hard)$ (Mallet, 2002). Theoretically, the fit to hard data should be better than the fit to soft data. In practice, deciding the value of λ^{hard}, as compared to the two weights λ^{soft} and λ_ϵ, is not trivial: if taken too small, hard data may not be honored, and if taken too large, the system (Equation (2.59)) may become ill conditioned. In addition, hard data may be honored with a negligible error, but not exactly.

Using the substitution

This technique can only be applied with nodal shape functions having the Kronecker Delta Property (KΔP, i.e., $u(\overline{\boldsymbol{p}}) = u_{\overline{\boldsymbol{p}}}$, Section 2.1.3.1). In this case, a node $\overline{\boldsymbol{p}}$ associated with a shape function $\Phi_{\overline{\boldsymbol{p}}}$ and a coefficient $u_{\overline{\boldsymbol{p}}}$ is added at the position of the hard data point $\boldsymbol{p}_{hard}$ (i.e., $\overline{\boldsymbol{p}} = \boldsymbol{p}_{hard}$). Then, the associated nodal coefficient $u_{\overline{\boldsymbol{p}}}$ is substituted by the DPV constraint's iso-value α_{hard} in the system (2.59).

Consequently, the coefficient $u_{\overline{\boldsymbol{p}}}$ is no longer considered an unknown to the problem as it is replaced by the value α_{hard}. The dimensions of the system are changed accordingly and the equations involving this coefficient are modified by transferring the corresponding terms in the right-hand member of the system. This is a typical technique employed in FEM (Dhatt & Touzot (1981), Section 5.3.2) and DSI (Frank *et al.*, 2007). The issue is that it requires to update the mesh which may generate ill-shaped elements.

Using Lagrange multipliers

This technique can be applied using any type of shape functions in a Bubnov-Galerkin scheme. It may be applicable to other schemes (i.e., collocation or Petrov-Galerkin) but the author did not find any example.

The Lagrange multipliers consider each weight $\lambda_{\boldsymbol{p}}$ associated to a hard datum as an unknown to the problem. The objective is to automatically determine each of these weights to perfectly honor their constraints. Let some hard DPV constraints be added to the modeling problem at the positions $\boldsymbol{p}$ with expected iso-values $\alpha_{\boldsymbol{p}}$. The general system (2.59) becomes a saddle problem (Brezzi, 1974):

$$\begin{bmatrix} \boldsymbol{K} & \boldsymbol{\Theta} \\ \boldsymbol{\Theta}^T & 0 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{\alpha} \end{bmatrix}, \tag{2.72}$$

where $\boldsymbol{K}$, $\boldsymbol{U}$ and $\boldsymbol{F}$ are defined by system (2.59) with $\Psi = \Phi$ and $N_V = N_U$, $\boldsymbol{\Theta}$ is the matrix gathering the vertical vectors of shape functions $\boldsymbol{\Phi}$ applied to each hard data point $\boldsymbol{p} \in \boldsymbol{D}(hard)$ and $\boldsymbol{\lambda}$ and $\boldsymbol{\alpha}$ are the sets of associated unknown weights $\lambda_{\boldsymbol{p}}^{hard}$ and expected iso-values $\alpha_{\boldsymbol{p}}^{hard}$. This square system $(\#\boldsymbol{N} + \#\boldsymbol{D}(hard), \#\boldsymbol{N} + \#\boldsymbol{D}(hard))$ also includes soft data constraints, but these are considered in the matrices $\boldsymbol{K}$ and $\boldsymbol{F}$ as their weights are given as inputs to the problem (i.e., not considered as unknowns). The details of the demonstration obtaining this system are given in Appendix G (p.171).

Although the Lagrange multipliers can be seen as an automatic technique to tune the data weights, and therefore not honoring exactly the hard data constraints, it is considered as an accurate method (Belytschko *et al.*, 1996b). The problem is that a saddle point system is not positive definite and possesses zeros on the diagonal. Therefore, available solvers for system (2.72) may not be the most efficient ones.

2.2.5.6 Inequality data constraints (IqD)

The IqD constraints can concern any types of data constraints presented above, but where the equalities are replaced by inequalities. For instance, IqD point constraints (IqDP, Section 1.4.2.1, p.51) can indicate that a point $\boldsymbol{p}$ belongs to a layer by enforcing its iso-value to be smaller (*resp.* bigger) than the top (*resp.* bottom) iso-surface of the layer (Frank *et al.*, 2007). If a data point position $\boldsymbol{p}$ is expected to have a smaller iso-value than a constant $\alpha_{\boldsymbol{p}}$, then the IqDP constraint corresponds to

$$u(\boldsymbol{p}) \leq \alpha_{\boldsymbol{p}}. \tag{2.73}$$

This is particularly handy for wellbore and outcrop observations where it is easier to link an interpreted data to a layer than a horizon. In Chilès *et al.* (2004) for example, such constraints are enforced at the bottom of the wellbores, where horizons are not reached.

Augmented Lagrangian method

The main difficulty with IqD constraints is that the framework has to be adapted to incorporate inequalities in the system. A common approach is to use the Augmented Lagrangian method (Nocedal & Wright, 2006). It sequentially solves linear systems with equality constraints, converging to a solution satisfying the inequality constraints. During the iterations, the inequality constraints are checked and activated with an increased penalty weight if not honored. The modification of the penalty weights follows certain rules and is called *augmentation*.

In the case of IqDP constraints, the associated functional J_{IqDP} defined with the Augmented Lagrangian method is written on the set $\boldsymbol{D}(Iq)$ of inequality points as

$$J_{IqDP}(u) = \sum_{\boldsymbol{p} \in \boldsymbol{D}(Iq)} \frac{1}{2\ \rho_{\boldsymbol{p}}} \left(\lambda_{\boldsymbol{p}} + \rho_{\boldsymbol{p}}(u(\boldsymbol{p}) - \alpha_{\boldsymbol{p}})\right)^2, \tag{2.74}$$

with $\lambda_{\boldsymbol{p}}$ and $\rho_{\boldsymbol{p}}$ the *augmentation* weights of the constraints. Following this section's developments on this functional leads to a linear problem equivalent to system (2.59). In this case, the Augmented Lagrangian method reduces to an iterative solve of the obtained system while augmenting the weights $\lambda_{\boldsymbol{p}}$ and $\rho_{\boldsymbol{p}}$ of the *violated* constraints (i.e., every point for which Equation (2.73) is not verified). The *augmentation* at the step $(n+1)$ is written with the solution vector $\boldsymbol{U}^{(n)}$ of the step n and with a chosen incremental factor a as

$$\lambda_{\boldsymbol{p}}^{(n+1)} = \lambda_{\boldsymbol{p}}^{(n)} + \rho_{\boldsymbol{p}}^{(n)} \left(\boldsymbol{\Phi}^T(\boldsymbol{p}) \cdot \boldsymbol{U}^{(n)} - \alpha_{\boldsymbol{p}}\right), \tag{2.75}$$

$$\rho_{\boldsymbol{p}}^{(n+1)} = a\ \rho_{\boldsymbol{p}}^{(n)}. \tag{2.76}$$

Getting rid of the weighting system with inequalities

The inherent weighting system on the constraints in the proposed framework is intuitive. However, it has no physical meaning and often requires to resolve a problem several times before getting a satisfactory result. Such an issue is solved for the hard data constraints where the weights are computed by Lagrange multipliers. Unfortunately, the weights of the soft data cannot be computed in the same manner.

The Augmented Lagrangian method can be seen as a variant to the Lagrange multipliers in the sense that it automatically finds the best weights for each inequality constraint to be respected. Therefore, a solution is to replace all the soft constraints by inequality constraints, and use the Augmented Lagrangian method.

For instance, a DPV constraint can be enforced by two inequality constraints as

$$\forall H,\ \forall \boldsymbol{p} \in H, \qquad \begin{array}{rcl} u(\boldsymbol{p}) - \alpha_H & \leq & +\epsilon_{\boldsymbol{p}}, \\ u(\boldsymbol{p}) - \alpha_H & \geq & -\epsilon_{\boldsymbol{p}}, \end{array} \tag{2.77}$$

with α_H the iso-value of the horizon H, and $[\alpha_H - \epsilon_{\boldsymbol{p}}, \alpha_H + \epsilon_{\boldsymbol{p}}]$ the interval of iso-values allowed for $u(\boldsymbol{p})$.

The value $\epsilon_{\boldsymbol{p}}$ can be chosen differently for each data point or horizon, and it can be scaled on the implicit function's trend (i.e., the norm of the gradient $||\nabla u||$) to be expressed as a distance (e.g., in meters). This trend can be deduced by preprocessing (Section 1.3.1.2, p.35), or normalized with other settings of data constraints (Section 1.3.2.2, p.41). With this approach, the reliability of the data, generally expressed in distances, can directly replace $\epsilon_{\boldsymbol{p}}$ and its influence on the constraints can therefore have a physical meaning. This approach is close to a PFM scheme where the nugget is scaled on the implicit function's trend.

2.2.5.7 Concept of minimum requirement of a data setting

Some sets of data constraints may not be sufficient to obtain a solution to the modeling problem. For instance, using only IDP constraints (i.e., $J_{data} \equiv J_{IDP}$) does not ensure a unique solution: the data points in each horizon are enforced to have the same iso-value, but no reference is given, so a solution could be the null constant field. A minimum requirement setting is thus defined as a set of data constraints ensuring the unicity of the solution for a given set of weights λ_ϵ and $\lambda_{\boldsymbol{p}}$.

Standard data setting

As presented in Section 2.2.2, the standard minimization problem is written with DPV constraints in this thesis. The minimum requirement setting is a set of at least two (*resp.* three) data points in 2D (*resp.* 3D) with the same iso-value, and at least one other data point with a different iso-value.

Other data settings

As presented in Section 1.3.2.2 (p.41), PFM generally uses (i) one reference DPV constraint, (ii) the remaining points are sorted by horizons and treated as IDP constraints, and (iii) the gradient vectors are normalized and used as NGD constraints. The minimum requirement setting is a set with at least the reference DPV constraint and one NGD constraint.

A variant of this data setting can be given by two DPV constraints with different iso-values, and the remaining points as IDP constraints. The minimum requirement setting uses the two reference DPV constraints and at least one IDP constraint (i.e., involving two points).

Many other data settings are possible, but their minimum requirement generally reduces to the three presented settings. All the data settings presented here can also be employed in both DSI and PFM.

2.2.6 Unconformities and faults treated as discontinuities

The discontinuities represent one of the main challenging problems in structural modeling. DSI creates meshes conformal to the faults (Section 1.3.1.6, p.38), and PFM defines fault zones with polynomials and jump functions (Section 1.3.2.7, p.46), but both approaches become challenging when dealing with complex fault networks and may require heavy user interactions. Stratigraphic unconformities may be addressed by several implicit functions (Section 1.2.2.2, p.32), which requires to separate each stratigraphic sequences beforehand.

For simplicity, we propose to address both faults and stratigraphic unconformities as inputs (Section 1.2.2.5, p.32) to the proposed framework and to treat them as discontinuities as presented in Section 2.1.4. They thus are considered as boundaries to the modeling problem and have to be handled with special techniques adapted to the chosen discretization.

2.2.7 Relation with existing methods

2.2.7.1 Discrete Smooth Interpolation

The proposed framework is close to DSI in the sense that the problem is posed as a minimization of a set of functionals (Equation (1.10), p.37). The only difference is that the regularization term (i.e., the penalization energy or, in DSI, the global roughness) is a continuous functional in our framework. A continuous global roughness factor is mentioned in DSI (e.g., formulated as the Laplacian energy, Mallet (1997); Levy & Mallet (1999)), but its formal discretization is never developed.

As the problem is discrete in DSI, there is no notion of integral approximation. This is the main reason why an empirical normalization on the smoothing equations is needed to ensure the convergence of a DSI method when refining the resolution of the mesh (Section 1.3.1.9, p.39). In the least squares version of the proposed framework (Section 2.2.4.4), the volumes ν_s are naturally deduced from the discretization of the continuous energy. These terms normalize the influence of the number of smoothing equations regardless of the discretization and its resolution. The continuous approach therefore gives theoretical arguments to normalize the weighting system of discrete least squares approaches such as DSI. Also, a possible normalization on the number of data constraints is presented in Appendix F (p.170). Unfortunately, the weighting system remains non-physical, so λ_ϵ and $\lambda_{\boldsymbol{p}}$ must be tuned for each model.

Concerning the global roughness, the best-known applied formulation can be related to the bending energy. A continuous version of the constant gradient constraint (1.7) (p. 36) would enforce

$$\forall \boldsymbol{x} \in \Omega,\ \forall \boldsymbol{v}, \quad \nabla u(\boldsymbol{x}) = \nabla u(\boldsymbol{x} + \boldsymbol{v}), \tag{2.78}$$

with $\boldsymbol{v}$ an arbitrary vector in the study's dimensional space. A first order development of $\nabla u(\boldsymbol{x} + \boldsymbol{v})$ is written as

$$\nabla u(\boldsymbol{x} + \boldsymbol{v}) = \nabla u(\boldsymbol{x}) + \nabla^2 u(\boldsymbol{x}) \cdot \boldsymbol{v}, \tag{2.79}$$

with $\nabla^2 = \boldsymbol{H}$ the Hessian operator. By combining Equations (2.78) and (2.79), we obtain

$$\boldsymbol{H}(u) \cdot \boldsymbol{v} = 0 \Leftrightarrow v_i \partial^2_{ij} u = 0, \tag{2.80}$$

with a sum assumed on i for each coordinate of space. In 2D, this notation gives

$$\begin{cases} v_x \partial^2_{xx} u + v_y \partial^2_{xy} u &= 0 \\ v_x \partial^2_{xy} u + v_y \partial^2_{yy} u &= 0 \end{cases}. \tag{2.81}$$

The bending energy, which enforces each second derivative to be null, hence implies the continuous version of the constant gradient constraint, but they are not equivalent.

2.2.7.2 Potential Field Method

PFM naturally minimizes the bending energy when using the Thin Plate Splines (TPS) (Duchon, 1977) as interpolants (Section 2.1.3.3). The proposed standard minimization problem (Equation (2.55)) is thus close to this form of PFM at the two exceptions that (i) the bending energy is minimized explicitly with a penalization weight, and (ii) the discretization is not reduced to an interpolation scheme.

When PFM is defined with another type of interpolant (e.g., the cubic covariance), it is unclear to the author if a corresponding set of energy functionals exists. The opposite is also true: it is unclear to the author if the Green's functions of any set of energy functionals can be found. Therefore, the proposed framework may not always be comparable with PFM. However, in surface construction, the global minimization problem given by Equation (5.1) is often presented before being reduced to an interpolation scheme with smooth RBF. This approach is not reduced to the proposed standard minimization problem (Equation (2.55)) treated with TPS (as in Turk & O'brien (1999)), it can also concern smoother problems treated with more complex RBF (as in Dinh *et al.* (2001)). Therefore, a review of existing fundamental solutions to different physical energies may benefit both approaches.

Another difference is that the NGD constraints must be written in a Hermite-Birkhoff interpolation scheme in PFM (Section 1.3.2.4, p.42). This is not a requirement of the proposed framework, but it is possible to use Hermite-type shape functions (Section 2.1.3.6) if desired.

Résumé du Chapitre 3

Dans ce troisième chapitre, nous proposons une première discrétisation de notre problème continu.

Les moindres carrés glissants (Moving Least Squares, MLS) sont utilisés comme fonctions de forme. Ces fonctions sont construites sur des nœuds discrétisés régulièrement dans l'espace ; ces nœuds ne sont donc pas confondus avec les points de données. Cela permet de définir une taille commune à tous les domaines d'influence des MLS tout en garantissant une couverture complète du domaine d'étude, c'est-à-dire que tout point du domaine est influencé par une ou plusieurs fonctions d'interpolation.

Les fonctions de forme sont également utilisées comme fonctions de test. Le domaine est discrétisé en sous-domaines réguliers, et les intégrales du problème continu sont approximées par quadrature à un point centrée respectivement sur chaque nœud de construction des fonctions MLS. Les discontinuités sont gérées avec le critère de visibilité qui empêche les fonctions MLS d'influencer l'interpolation de part et d'autre d'une faille en utilisant des tests d'intersection entre segments et surface de faille. Le problème est ensuite résolu au sens des moindres carrés et la fonction implicite obtenue est évaluée et visualisée sur une grille Cartésienne. Les surfaces correspondant aux iso-valeurs des horizons sont ensuite extraites par des algorithmes d'extraction linéaire sur triangles et tétraèdres (marching triangle et marching tetrahedra).

La méthode est illustrée sur des modèles synthétiques 2D et 3D, et l'ensemble des contraintes définies dans le Chapitre 2 sont testées en 2D. Les limites du critère de visibilité sont également illustrées et la méthode de transparence est proposée comme solution alternative. Le chapitre finit sur une étude de sensibilité de la méthode réalisée sur une coupe géologique 2D regroupant des plis, des failles, et une érosion. Les résultats permettent d'avoir une idée précise du comportement de la méthode en fonction des paramètres choisis et de la qualité des données. Les avantages apportés par cette méthode résident dans la simplicité avec laquelle les failles et autres discontinuités sont gérées, et dans sa flexibilité pour échantillonner des géométries complexes.

Chapter 3

Moving Least Squares discretization

3.1 Motivations

In the previous chapters, we discussed the limits of existing methods in implicit structural modeling and showed that the opportunities of improvement offered by these methods are restricted by their assumption of discretization (Chapter 1, p.23). This allowed us to consider posing the modeling problem with continuous equations to benefit from other numerical methods (Chapter 2, p.59). Here, a first discretization of the proposed framework is suggested. With this approach, we intend to inherit from most of the advantages of both the Discrete Smooth Interpolation (DSI) and the Potential Field Method (PFM) while mitigating their shortcomings. In Section 1.3.1 (p.34), DSI is presented as a method producing smooth surfaces for any data set as long as a good quality mesh is used. The sparsity of the system, which is dependent on the mesh and its topology, is attractive. In Section 1.3.2 (p.40), PFM is also presented as a method producing smooth results, but it should be used on a limited number of data constraints. Although optimization techniques are available, we intend to create a method that is inherently scalable. However, the fact that PFM does not require a mesh to create the implicit function is attractive. Therefore, we introduce an algorithm to build implicit functions using locally defined Moving Least Squares shape functions. Such an approach has already been suggested in Maerten (2018), but the method is not deduced from the discretization of a continuous problem for structural modeling, and no specific strategies of kernel sampling adapted to the MLS functions are given. Here, the functions are centered on regularly sampled nodes to ensure the definition of the implicit function everywhere in the domain. The corresponding ranges of influence (i.e., support sizes) are scaled on the nodal spacing, which generates a sparse system of equations. The creation and storage of a complex mesh is avoided, but the problem of node discretization remains. The discontinuities are handled with optic criteria (Section 2.1.4.2, p.72), limiting user interactions even in complex cases. These decisions on the employed shape functions and the techniques for the discontinuities are discussed in Section 2.1.7.2 (p.79). The presented method is the subject of the published paper Renaudeau *et al.* (2019b). This chapter is an extended version of this paper.

3.2 The proposed method

3.2.1 The modeling problem using Moving Least Squares functions

As presented in Section 2.2.3.3 (p.81), the standard modeling problem is posed as a regression of data points value (DPV) constraints penalized by the bending energy as

$$\min_u \left(J(u)\right) \quad = \min_u \left(\frac{1}{2} \int_\Omega \lambda_\epsilon^2 \left(\partial_{ij}^2 u\right)^2 d\Omega + \frac{1}{2} \sum_{\boldsymbol{p} \in \boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \left(u(\boldsymbol{p}) - \alpha_{\boldsymbol{p}}\right)^2 \right), \tag{3.1}$$

where the iso-values $\alpha_{\boldsymbol{p}}$ are determined with a pre-processing.

The implicit function u is taken in the space U defined with Moving Least Squares functions as

$$U = \{u(\boldsymbol{x}) = \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}} \Phi_{\overline{\boldsymbol{p}}}^{MLS}(\boldsymbol{x}) \; u_{\overline{\boldsymbol{p}}} = \boldsymbol{\Phi}(\boldsymbol{x})^T \cdot \boldsymbol{U} \mid \boldsymbol{x} \in \Omega\}, \tag{3.2}$$

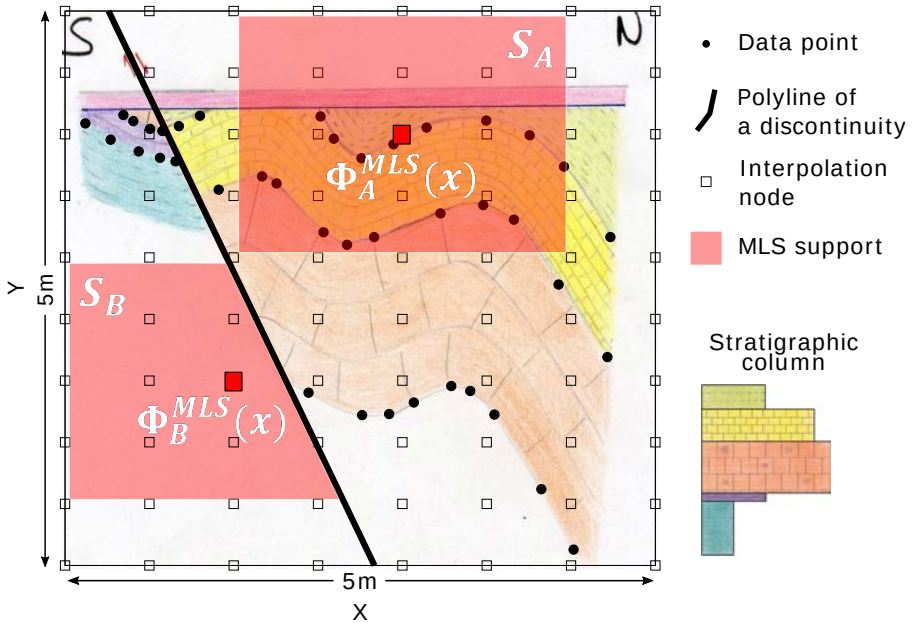


Figure 3.1: *Schematic representation of the proposed modeling method using MLS shape functions on a synthetic 2D cross section of a folded outcrop with one normal fault (from Figure 1.9).*

with $\boldsymbol{N}$ the set of nodes supporting the interpolation and $\boldsymbol{U}$ the problem's unknowns.

In this section, the details of the method solving this problem is presented, involving the nodal distribution, the definition of the moving least squares shape functions and their default parameters, the discretization of the continuous energy, and how the structural discontinuities are handled. The approach is illustrated in Figure 3.1 on a handmade interpreted cross section.

3.2.2 Node discretization

The chosen shape functions are locally defined on the set of nodes $\boldsymbol{N}$. For simplicity, we propose to discretize these nodes regularly in the domain of study Ω. This discretization is equivalent to a Cartesian grid, but the grid is not stored, only its cell corner points (i.e., the nodes). As the shape functions $\Phi_{\overline{\boldsymbol{p}}}^{MLS}$ are linearly independent, the implicit function is uniquely defined by the coefficients $u_{\overline{\boldsymbol{p}}}$. Each function $\Phi_{\overline{\boldsymbol{p}}}^{MLS}$ and each coefficient $u_{\overline{\boldsymbol{p}}}$ have an influence in a local support $S_{\overline{\boldsymbol{p}}}$. An example of regularly distributed interpolation nodes and two local supports S_A and S_B are illustrated in Figure 3.1.

3.2.3 Moving Least Squares (MLS) shape functions

The MLS shape functions are defined in Section 2.1.3.4 (Equation (3.3), p.92) together with their second order derivatives (Equation (3.5), p.92). These equations are repeated here. The shape functions are written

$$\forall \overline{\boldsymbol{p}} \in \boldsymbol{N},\ \forall \boldsymbol{x} \in \Omega, \quad \Phi_{\overline{\boldsymbol{p}}}^{MLS}(\boldsymbol{x}) = w_{\overline{\boldsymbol{p}}}(\boldsymbol{x})\mathcal{P}^T(\boldsymbol{x}) \cdot [\boldsymbol{A}(\boldsymbol{x})]^{-1} \cdot \mathcal{P}(\overline{\boldsymbol{p}}) = \boldsymbol{\Gamma}(\boldsymbol{x}) \cdot \boldsymbol{B}_{\overline{\boldsymbol{p}}}(\boldsymbol{x}), \tag{3.3}$$

where $\boldsymbol{A}$ is the moment matrix defined as

$$\boldsymbol{A}(\boldsymbol{x}) = \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}} w_{\overline{\boldsymbol{p}}}(\boldsymbol{x})\mathcal{P}(\overline{\boldsymbol{p}}) \cdot \mathcal{P}^T(\overline{\boldsymbol{p}}), \tag{3.4}$$

and with $\mathcal{P}$ a complete polynomial basis, $\boldsymbol{\Gamma}$ the solution of $\boldsymbol{A} \cdot \boldsymbol{\Gamma} = \mathcal{P}$, $\boldsymbol{B}_{\overline{\boldsymbol{p}}} = w_{\overline{\boldsymbol{p}}}(\boldsymbol{x})\mathcal{P}(\overline{\boldsymbol{p}})$ and $w_{\overline{\boldsymbol{p}}}$ the weight functions. The second derivatives are written

$$\partial_{ij}^2 \Phi_{\overline{\boldsymbol{p}}}^{MLS}(\boldsymbol{x}) = \boldsymbol{\partial}_{ij}^2 \Gamma^T \cdot \boldsymbol{B}_{\overline{\boldsymbol{p}}} + \boldsymbol{\partial}_i \Gamma^T \cdot \partial_j \boldsymbol{B}_{\overline{\boldsymbol{p}}} + \boldsymbol{\partial}_j \Gamma^T \cdot \partial_i \boldsymbol{B}_{\overline{\boldsymbol{p}}} + \boldsymbol{\Gamma}^T \cdot \partial_{ij}^2 \boldsymbol{B}_{\overline{\boldsymbol{p}}}, \tag{3.5}$$

where $\partial_i\Gamma$ is solution of

$$\boldsymbol{A}\cdot\boldsymbol{\partial}_i\Gamma = \partial_i\mathcal{P} - \partial_i\boldsymbol{A}\cdot\boldsymbol{\Gamma}, \tag{3.6}$$

and $\partial^2_{ij}\Gamma$ is solution of

$$\boldsymbol{A}\cdot\boldsymbol{\partial}^2_{ij}\Gamma = \partial^2_{ij}\mathcal{P} - (\partial_i\boldsymbol{A}\cdot\partial_j\boldsymbol{\Gamma} + \partial_j\boldsymbol{A}\cdot\boldsymbol{\partial}_i\Gamma + \partial^2_{ij}\boldsymbol{A}\cdot\boldsymbol{\Gamma}). \tag{3.7}$$

Some details of the MLS functions construction are given in Appendix A (p.165) and a geometrical illustration of an MLS interpolation in 1D is given in Figure 2.1 (p.67). Other details on their construction can be found in Fries & Matthias (2004), and Nguyen *et al.* (2008) and useful empirical values for their parameters can be found in Liu & Gu (2005).

3.2.3.1 Default parameters

Weight functions

The fourth order splines w^{FS} with cubic supports are used. Each weight function w is centered on a node $\overline{\boldsymbol{p}}(\overline{p}_x,\overline{p}_y,\overline{p}_z)$ and written

$$w_{\overline{\boldsymbol{p}}}(\boldsymbol{x}) = w^{FS}\left(\frac{|x-\overline{p}_x|}{\rho_x}\right)\ w^{FS}\left(\frac{|y-\overline{p}_y|}{\rho_y}\right), \tag{3.8}$$

in 2D, and

$$w_{\overline{\boldsymbol{p}}}(\boldsymbol{x}) = w^{FS}\left(\frac{|x-\overline{p}_x|}{\rho_x}\right)\ w^{FS}\left(\frac{|y-\overline{p}_y|}{\rho_y}\right)\ w^{FS}\left(\frac{|z-\overline{p}_z|}{\rho_z}\right), \tag{3.9}$$

in 3D, with $\rho(\rho_x,\rho_y,\rho_z)$ the dilatation parameter, and

$$\forall r_i\geq 0,\quad w^{FS}(r_i) = \begin{cases} 1-6\left(\frac{r_i}{\rho_i}\right)^2+8\left(\frac{r_i}{\rho_i}\right)^3-3\left(\frac{r_i}{\rho_i}\right)^4 & if\quad r_i\leq\rho_i \\ 0 & if\quad r_i>\rho_i \end{cases}, \tag{3.10}$$

with r_i and ρ_i the distance and dilatation parameter in the axis i. The created weight functions are C^2 (Fries & Matthias, 2004)).

Polynomial order

The global polynomial order $order(\mathcal{P})$ of the MLS shape functions is fixed to 1. The number of monomials $\#\mathcal{P}$ in the corresponding complete polynomial basis is 2 (*resp.* 3 and 4) in 1D (*resp.* 2D and 3D) (Section 2.1.3.1, p.64).

Dilatation parameter

The dilatation parameters $\boldsymbol{\rho}_{\overline{\boldsymbol{p}}}(\rho_x,\rho_y,\rho_z)$ control the size of the support $S_{\overline{\boldsymbol{p}}}$ associated to the node $\overline{\boldsymbol{p}}$. A support $S_{\overline{\boldsymbol{p}}}$ is the support of both the weight function $w_{\overline{\boldsymbol{p}}}$ and the shape function $\Phi_{\overline{\boldsymbol{p}}}$. Two supports S_A and S_B are illustrated in Figure 3.1. A global and adimensional dilatation parameter ρ is used to define constant dilatation parameters scaled on the nodal spacing in each axis i as

$$\forall\overline{\boldsymbol{p}}\in\boldsymbol{N},\quad \boldsymbol{\rho}_{\overline{\boldsymbol{p}}} = \left[\rho\ \frac{L_i(\Omega)}{(\#\boldsymbol{N}(i)-1)}\ \middle|\ \forall i\right], \tag{3.11}$$

where $L_i(\Omega)$ is the length of the domain Ω and $\#\boldsymbol{N}(i)$ is the number of interpolation nodes in the i axis. With the cubic supports, the regular sampling, and this normalization, the relation between a node $\overline{\boldsymbol{p}}$ and its neighbors $\boldsymbol{N}(\overline{\boldsymbol{p}})$ is controlled by the parameter ρ. This parameter is fixed to 1.99.

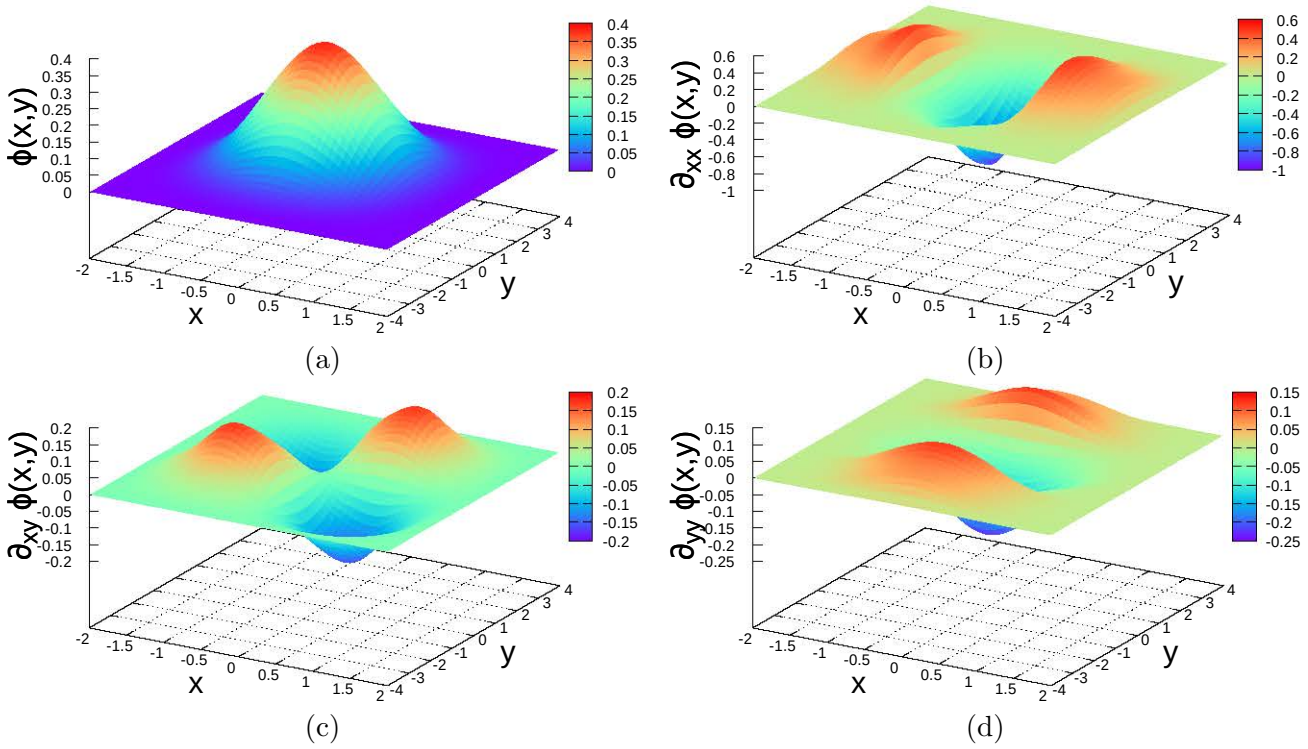


Figure 3.2: *2D plots of the MLS function and its second order derivatives on a center node of a* (9×9) *grid with* $\rho = 1.99$.

Derivability

The derivability of the MLS shape functions is guided by the chosen weight functions; here, it is C^2. On the illustration of the 1D MLS interpolation in Section 2.1.3 (Figure 2.1, p.67), the function u is strictly linear in areas where only two supports are influencing the interpolation. It is unlikely that a steady variation like this could be guided by highly continuous weight functions. This is due to the first order polynomial (i.e., $\#\boldsymbol{\mathcal{P}} = 2$) and the local number of neighboring nodes (i.e., $\#\boldsymbol{N}(\boldsymbol{x}) = 2$). This is a specific case: the number of monomials is equal to the number of neighbors, so the local system to determine the coefficients (system (A.7), p.166) is square, hence the weight functions have no influence on its solution. If the number of neighbors is bigger than the number of monomials, which is generally the case in the proposed setting, the MLS shape functions are as derivable as their weight functions.

3.2.3.2 Plot of the MLS functions and second order derivatives

The MLS functions and their derivatives are dependent on the dilatation parameter $\boldsymbol{\rho}_{\overline{\boldsymbol{p}}}$ and the neighboring nodes. If supported by regularly sampled nodes, they become identical one from another when they are away from the borders (i.e., with a gap of at least $\boldsymbol{\rho}_{\overline{\boldsymbol{p}}}$, Figure H.2(a), p.172). An example of an MLS function and its second derivatives are represented in Figure 3.2. These are the functions associated to the center node $\overline{\boldsymbol{p}}$ of a grid of (9×9) nodes and using an adimentional ρ parameter of 1.99. To show the normalization effect of ρ, the grid has a spacing twice bigger in the y axis than in the x axis.

Other plots of the weight and MLS functions together with their derivatives using the presented default setting are given in Appendix H (p.172). Similar plots can also be found in Liu & Gu (2005) and Nguyen *et al.* (2008).

3.2.4 Test functions and domain discretization

As the shape functions are regularly discretized over the entire domain of study Ω, using the underlying grid to perform the quadrature approximation of the integration term as in the Element Free Galerkin

method is attractive. Therefore, we suggest to follow the EFG formulation with a Bubnov-Galerkin scheme and a domain discretization on the underlying grid.

The basis test functions is the same as the basis shape functions (i.e., $\boldsymbol{\Psi} \equiv \boldsymbol{\Phi}$). The domain Ω is regularly divided into subdomains $\omega_{\overline{\boldsymbol{p}}}$ respectively centered on a node $\overline{\boldsymbol{p}} \in \boldsymbol{N}$ (i.e., $\Omega \subset \cup\omega_{\overline{\boldsymbol{p}}}$, $\forall \overline{\boldsymbol{p}} \in \boldsymbol{N}$) and having the same volume ν (i.e., the volume of a cell of the regular sampling). These subdomains are not the MLS supports $S_{\overline{\boldsymbol{p}}}$ as they do not overlap. The integration term is approximated as constant in each subdomain $\omega_{\overline{\boldsymbol{p}}}$ and evaluated at the center node $\overline{\boldsymbol{p}}$ (i.e., quadrature with one point, Appendix E, p.169). The integration grid is thus implicit and never stored.

As presented in Section 2.2.4.4 (p.83), this approach leads to a system equivalent to a least squares system with each constraint written as a row (i.e., system (2.60)), which is convenient for implementation.

3.2.5 Handling the discontinuities

Both faults and stratigraphic unconformities are first modeled with explicit methods (Section 1.2.2.3, p.32), and used as inputs to the presented method. We then suggest to address these discontinuity surfaces with optic criteria (Section 2.1.4.1, p.72).

For simplicity, we propose to use the visibility criterion as the standard optic criterion. Discontinuous jumps in the implicit function are thus introduced by local intersection tests between segments and discontinuity objects (e.g., polylines in 2D or triangulated surfaces in 3D). This approach has the potential to reduce the user interactions to handle structural discontinuities as compared to other modeling methods, such as DSI and PFM. As an example, the support S_B affected by the visibility criterion is illustrated in Figure 3.1. Other illustrations of the effects of the different optic criteria on the supports $S_{\overline{\boldsymbol{p}}}$ are given in Figure 2.2 (p.72) and in Fries & Matthias (2004).

3.2.6 Example of a system of equations

In 2D for instance, the least squares system corresponding to the minimization problem of Equation (3.1) with the presented discretization can be written as

$$\left\{\begin{array}{rcll} \lambda_\epsilon\ \sqrt{\nu} & \left(\partial^2_{xx}\boldsymbol{\Phi}(\overline{\boldsymbol{p}})^T \cdot \boldsymbol{U}\right) & = 0, & \forall \overline{\boldsymbol{p}} \in \boldsymbol{N}, \\ \lambda_\epsilon\ \sqrt{\nu} & \left(\partial^2_{yy}\boldsymbol{\Phi}(\overline{\boldsymbol{p}})^T \cdot \boldsymbol{U}\right) & = 0, & \forall \overline{\boldsymbol{p}} \in \boldsymbol{N}, \\ \lambda_\epsilon\ \sqrt{2\nu} & \left(\partial^2_{xy}\boldsymbol{\Phi}(\overline{\boldsymbol{p}})^T \cdot \boldsymbol{U}\right) & = 0, & \forall \overline{\boldsymbol{p}} \in \boldsymbol{N}, \\ \lambda_{\boldsymbol{p}} & \left(\boldsymbol{\Phi}(\boldsymbol{p})^T \cdot \boldsymbol{U}\right) & = \lambda_{\boldsymbol{p}}\ \alpha_{\boldsymbol{p}}, & \forall \boldsymbol{p} \in \boldsymbol{D}, \end{array}\right. \tag{3.12}$$

We solve this system with an LU decomposition solver for sparse matrices.

3.2.7 Extraction of the iso-surfaces

After solving system (3.12), the obtained coefficients $u_{\overline{\boldsymbol{p}}}$ can be used to evaluate the implicit function u in space. For visualization, the domain Ω is discretized on a grid, and the function is evaluated on the grid's corner points (i.e., the set $\boldsymbol{N}_v$ of visualization points). The iso-surfaces are then extracted linearly on each grid element with marching triangles in 2D and marching tetrahedra in 3D. As the mesh is not conformal to the discontinuities, these techniques create stair-step effects in the extracted surfaces as discussed in Section 1.2.1.4 (p.31). This is particularly visible in the 3D results. In 2D, the resolution of visualization is chosen fine enough to have the stair-steps under the fault lines. As in PFM, this is the main limit to our method as further extraction techniques must be investigated to create sealed models.

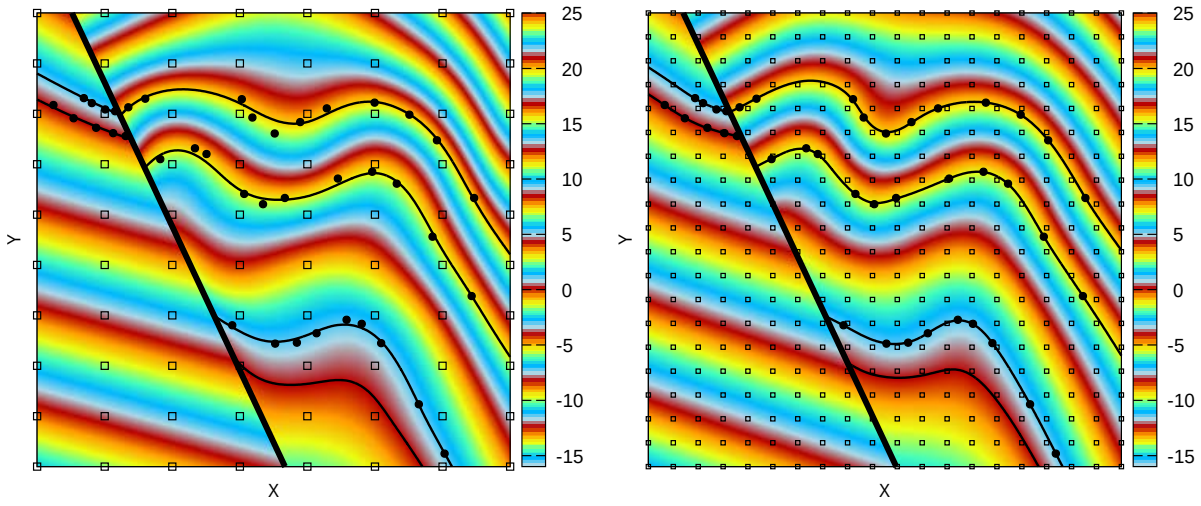


Figure 3.3: *Results computed on the numerical data presented in Figure 3.1 with two different nodal resolutions. (a) Low resolution* (8×10)*. (b) Higher resolution* (20×20)*.*
Iso-values from bottom to top: $[0, 1, 7, 11.5]$, $\lambda_\epsilon = 5$, $\lambda_{\boldsymbol{p}} = 1$.

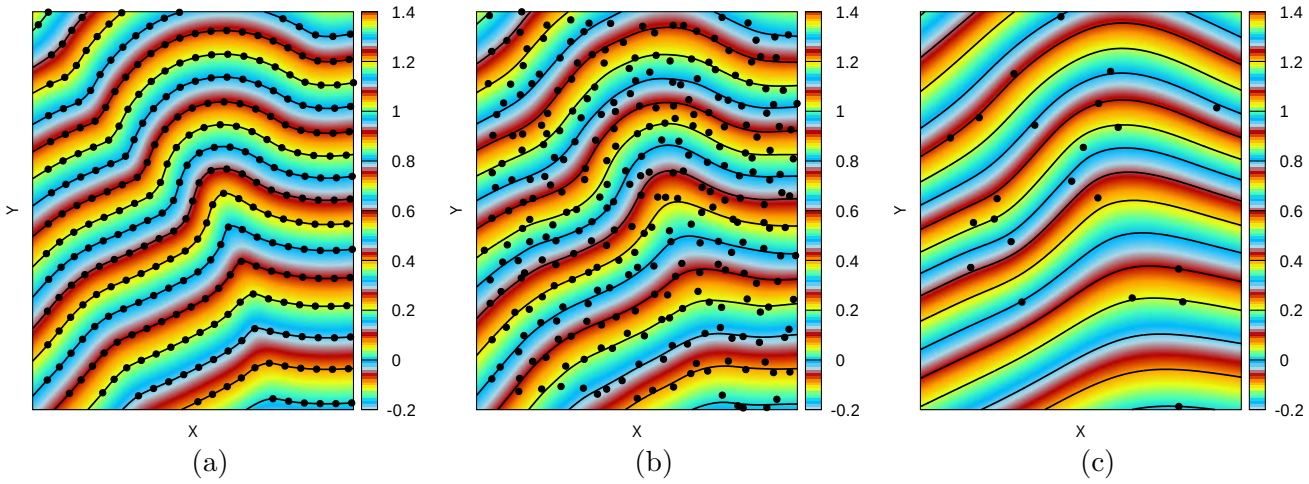


Figure 3.4: *Results on a synthetic model of an isopaque fold with different qualities of data samplings. (a) The data points are densely and regularly sampled (319 data points). (b) The data points are densely sampled and noisy (310 data points). (c) The data points are sparsely sampled and noisy (22 data points).*
Iso-values from bottom to top: $[0, 0.07, 0.14, 0.20, 0.26, 0.32, 0.38, 0.43, 0.48, 0.53, 0.58, 0.64, 0.70, 0.74, 0.80, 0.87, 0.92, 1]$, $\lambda_\epsilon = 20$, $\lambda_{\boldsymbol{p}} = 1$.

3.3 Some results

The proposed method is tested on a few 2D and 3D examples. All the 2D results presented in this section are visualized on a fine grid (i.e., 300×300) in order to have high quality images. These have been tested with this thesis's implementation of DSI and PFM in Section 1.3 (p.34), whose illustrations can be used for comparison.

The computing times of the 2D examples are approximately the same: less than a second to create the implicit function (between $0.1s$ and $0.7s$), and around four seconds to print the function on the visualization grid. The computing times and visualization resolutions for the 3D examples are given explicitly in their description in the text. Please note that this is a research code and that these computing times are not involving any optimization of the code. All the models are run on a laptop with Intel Core i7-4940 3GHz with 32 Gb of RAM, and running Windows 7 enterprise 64 bits.

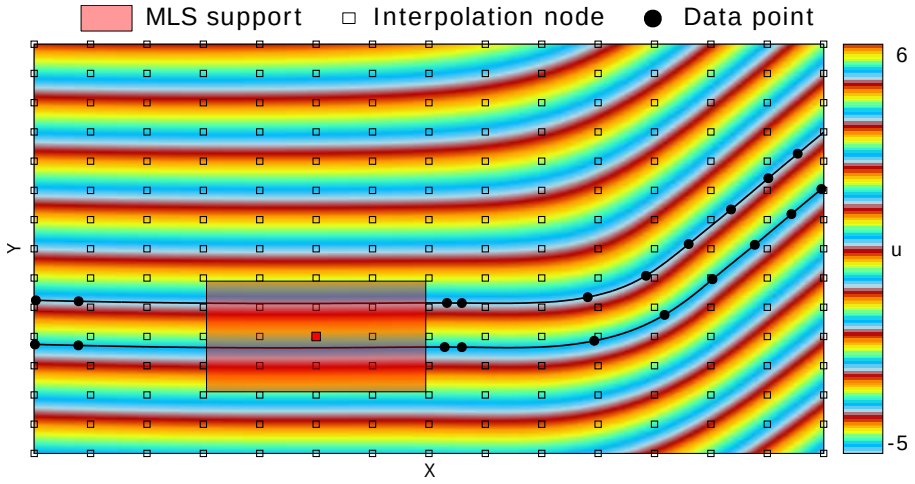


Figure 3.5: *Synthetic cross section with a gap of missing data points. The interpolation nodes are discretized on a* (15×15) *grid with an adimentional parameter* ρ *of* 1.99. *An example of MLS support is illustrated to give an idea of the used size of support.*
Iso-values from bottom to top: $[0, 1]$, $\lambda_\epsilon = 1$, $\lambda_{\boldsymbol{p}} = 1$.

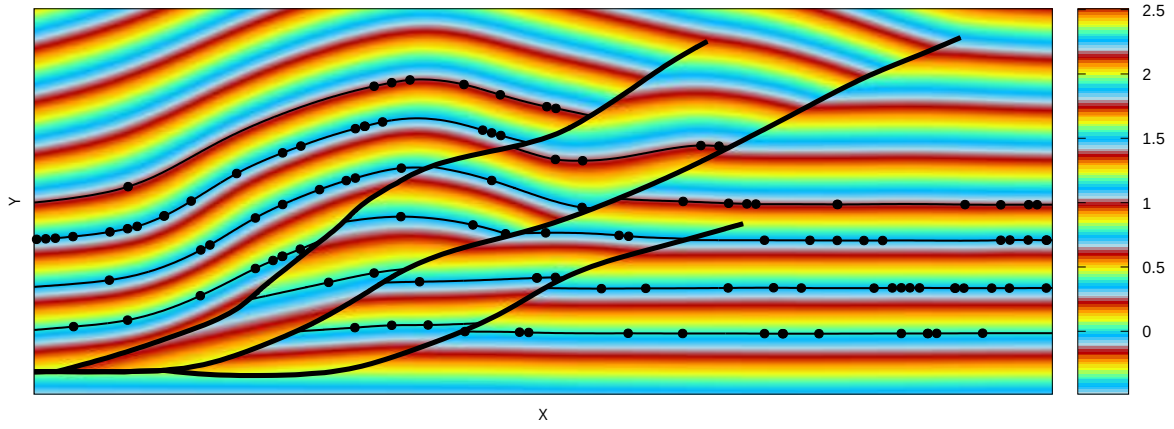


Figure 3.6: *Synthetic cross section with three discontinuities intersecting with small angles. The visibility criterion is used with a nodal discretization of* (50×50).
Iso-values from bottom to top: $[0, 0.34, 0.71, 1]$, $\lambda_\epsilon = 1$, $\lambda_{\boldsymbol{p}} = 1$.

3.3.1 Standard setting

In Figure 3.3, the standard method is tested on the cross section of Figure 3.1 with two different nodal discretizations: the same resolution as in Figure 3.1 is used in Figure 3.3(a), and a higher resolution is used in Figure 3.3(b). Although the same weights λ_ϵ and $\lambda_{\boldsymbol{p}}$ are used, the data are better fitted with more nodes.

The method's capability to handle both noise and sparsity is shown in Figure 3.4. The densely and regularly sampled data points of Figure 3.4(a) are perturbed in (b) and both perturbed and decimated in (c). Those perturbations and decimations are performed randomly with uniform laws further detailed in the sensitivity tests (Sections 3.4.3.1 and 3.4.3.2). The proposed algorithm smoothly filters the noise and interpolates between sparse data points.

The algorithm also handles areas of missing data as shown in Figure 3.5. For this model, the nodal resolution and the chosen dilatation parameter create supports smaller than the ones used in the PFM example of Figure 1.19 (p.44), but no abnormal high curvature is created in the results here. This example thus illustrates that supporting the interpolation on background nodes rather than data points enables to use small ranges of influence while preserving the stability of the results. The system is sparser, but in this case it is also much bigger (i.e., (695×225) here, versus (20×20) with PFM).

Figure 3.6 illustrates the capability of the method to handle finite discontinuities that intersect with

a small angle. With a fine nodal discretization, the visibility criterion is sufficient to obtain satisfactory results in the vicinities of tips and intersections of discontinuities. The surfaces of discontinuities are handled with intersection tests between the faults and the MLS functions, which can be performed during the evaluation of the system of equations. In comparison to DSI and PFM, this approach avoids the pre-evaluation steps of creating a conformal mesh or volumetric fault zones.

Finally, a 3D version of the standard method is tested on two synthetic models in Figure 3.7. In Figures 3.7(a), (c) and (e), the model includes folds, thickness variations, two faults, and an unconformity (blue). The inputs count 3 triangulated surfaces and 8815 data points. The implicit function was computed with $(40 \times 40 \times 30)$ nodes in 25 seconds, and printed on a $(50 \times 50 \times 50)$ cube in 1 minute. In Figures 3.7(b), (d) and (f), the model includes one large scaled fold and faults with small angle intersections. The inputs count 15 triangulated surfaces and 18021 data points. The implicit function was computed with $(80 \times 80 \times 30)$ nodes in 1 minute and 15 seconds and printed on a $(100 \times 100 \times 100)$ cube in 2 minutes and 40 seconds. These models were run on a laptop with Intel Core i7-4940 3GHz with 32 Gb of RAM, and running Windows 7 enterprise 64 bits.

3.3.2 Other data constraints

The standard setting of the method is changed to include all the other constraints presented in Section 2.2.5 (p.83). The remaining parameters are kept as defined in Section 3.2. All the constraints presented here have already been published for DSI and PFM (Section 1.3, p.34), we only demonstrate that the proposed framework can do just as much.

In Figure 3.8, the following constraints are used all together: data points value (DPV), increment data points (IDP), polygonal lines increments (PLI), normal gradient data (NGD) and tangent data (TD). From Figure 3.8(a) to (f), each type of constraint is either modified or erased to appreciate their respective influences on the results. This model is the same a presented in Figure 1.17 (p.41) for PFM's constraints, but using tangent and line data in addition; note that PFM (and DSI) also handle such constraints.

As the MLS functions do not have the Kronecker Delta Property, the best solution to enforce hard data constraints is to use the Lagrange multipliers as discussed in Section 2.2.5.5 (p.85). In Figure 3.9(b), two hard data points value (HDPV) constraints are enforced with this method and the results can be compared with Figure 3.9(a) where only soft DPV constraints are used.

The inequality constraints are implemented with the Augmented Lagrangian to avoid the non-physical weighting system (Section 2.2.5.6, p.86). In Figure 3.10, the regularly sampled data points from model (a) are perturbed in models (b) to (f) with a maximum displacement in a radius of one meter. The interval of allowed iso-values away from the data points (i.e., $[\alpha_{\boldsymbol{p}} - \epsilon_{\boldsymbol{p}}, \alpha_{\boldsymbol{p}} + \epsilon_{\boldsymbol{p}}]$) is scaled on the average gradient's norm of a previously computed model with DPV constraints, which is why $\epsilon_{\boldsymbol{p}}$ is expressed in meters. It could have been computed in other ways as discussed in Section 2.2.5.6 (p.86), and better results might have been obtained with a locally computed gradient norm. To ease the observations, a grid with a spacing of a meter in each axis is represented.

When the data are not perturbed, a small distance $\epsilon_{\boldsymbol{p}}$ can be employed to fit perfectly all data points (Figure 3.10(a)). Otherwise, the bigger $\epsilon_{\boldsymbol{p}}$, the smoother the results. When $\epsilon_{\boldsymbol{p}}$ is smaller than the noise displacement, the results are deteriorated and create stratigraphic inconsistencies (Figures 3.10(e) and (f)). This example shows that the uncertainty of the numerical data (around $1m$ here) can be directly used as $\epsilon_{\boldsymbol{p}}$ value to obtain satisfactory results with this formalism. Although it is simplified, the system is still dependent to the penalty weights: if $\lambda_{\boldsymbol{p}}$ is chosen too big as compared to λ_ϵ for a given model, then a deteriorated model like (f) may be obtained as a first (and final) guess in the Augmented Lagrangian method. This is why a small $\lambda_{\boldsymbol{p}}$ is chosen in Figure 3.10. Also, as this algorithm is iterative, the system of equations is solved multiple times, which increases the computing times. For instance, the results from Figure 3.10(c) to (f) where obtained after 5 to 15 iterations which required $1.5s$ to $4.5s$ to compute the implicit function (instead of $0.3s$ using DPV constraints).

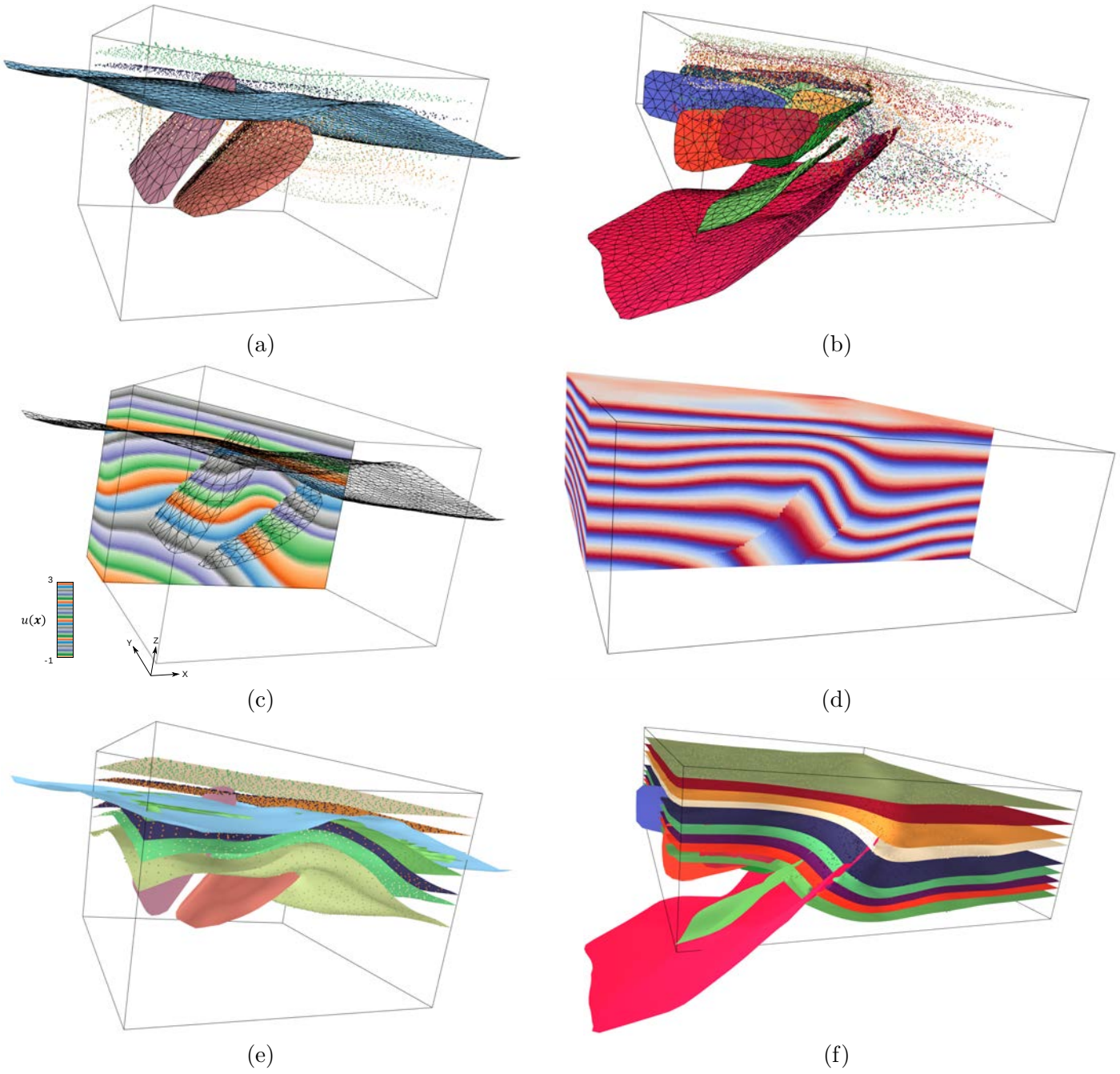


Figure 3.7: *Two synthetic models computed in 3D. (a) and (b) The data points and discontinuities as triangulated surfaces. (c) and (d) The two computed implicit functions. (e) and (f) The two computed models after iso-surfaces extraction.*
$\lambda_\epsilon = 1$, $\lambda_{\boldsymbol{p}} = 1$.
(Data sets provided by Laurent Maerten, Schlumberger)

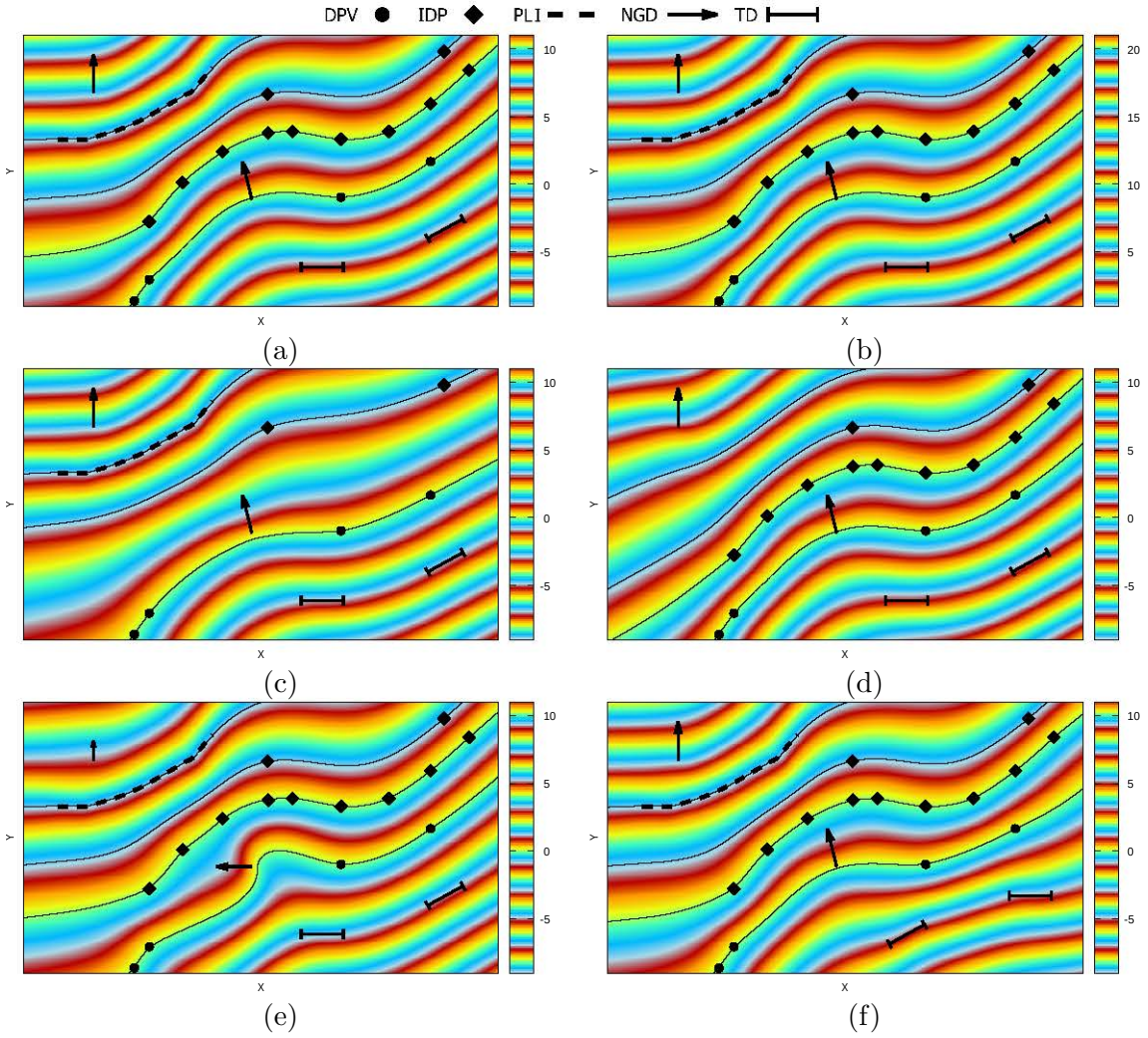


Figure 3.8: *Influence of different types of constraints on the results. (a) The reference model including all the constraints with a reference DPV constraint* $\alpha_0 = 0$. *(b) The DPV constraint is changed to* $\alpha_0 = 10$. *(c) A horizon of IDP constraints is erased. (d) The PLI constraint is erased. (e) The norm of one of the NGD constraints is divided by two, and the direction of the other is changed. (f) The directions of the TD constraints are changed.*
The reference NGD constraints have a norm of 1. $\lambda_\epsilon = 1$, $\lambda_{\boldsymbol{p}} = 1$.

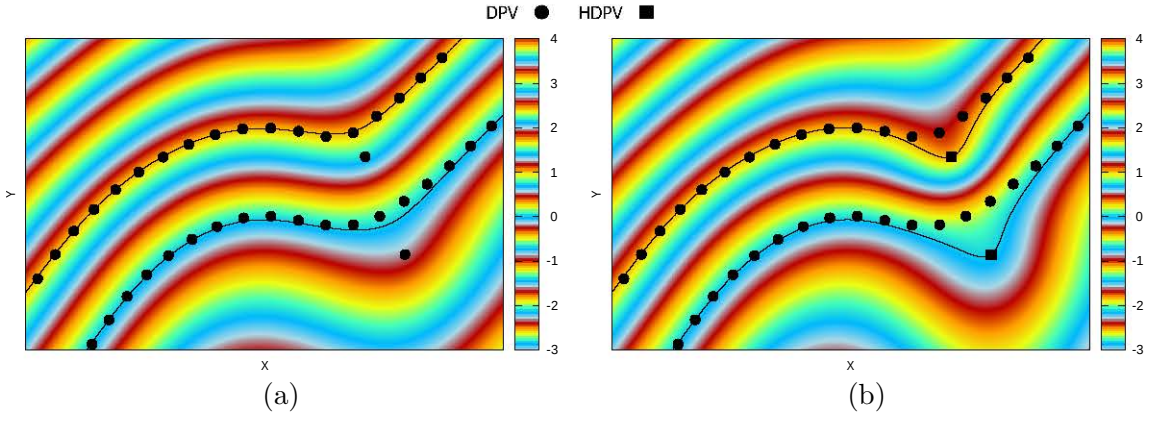


Figure 3.9: *Influence of hard data points on the results when enforced with Lagrange multipliers.*
Iso-values from bottom to top: $[0, 1]$, $\lambda_\epsilon = 1$, $\lambda_{\boldsymbol{p}}^{DPV} = 1$.

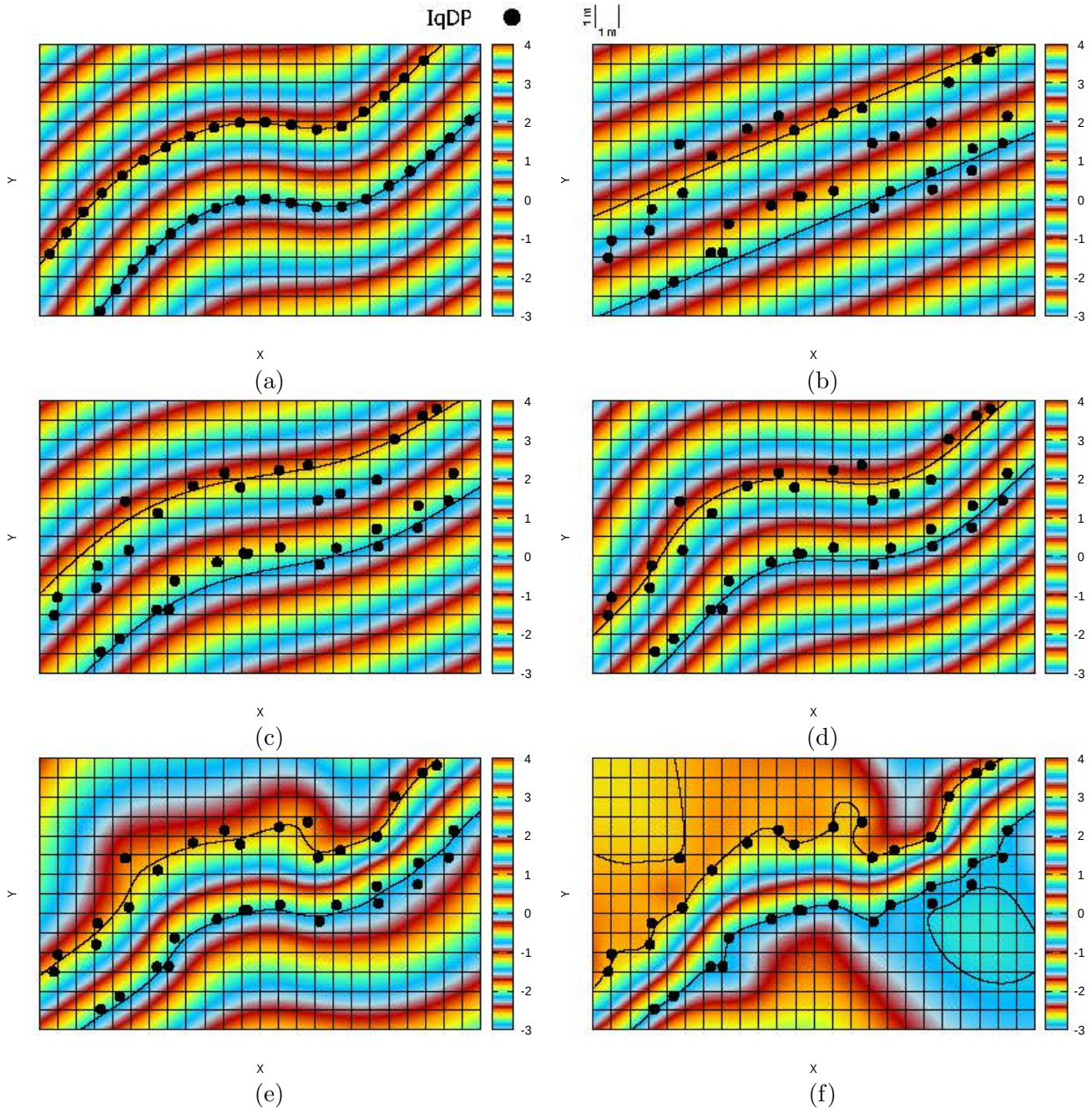


Figure 3.10: *Influence of different tolerances of data distance* $\epsilon_{\boldsymbol{p}}$ *when using inequality constraints. The grid gives both the scale and the distribution of interpolation nodes:* (25×15). *(a)* $\epsilon_{\boldsymbol{p}} = 0.1m$. *(b)* $\epsilon_{\boldsymbol{p}} = 5m$. *(c)* $\epsilon_{\boldsymbol{p}} = 2m$. *(d)* $\epsilon_{\boldsymbol{p}} = 1m$. *(e)* $\epsilon_{\boldsymbol{p}} = 0.5m$. *(f)* $\epsilon_{\boldsymbol{p}} = 0.1m$. *Iso-values from bottom to top:* $[0, 1]$, $\lambda_\epsilon = 1$, $\lambda_{\boldsymbol{p}} = 0.01$.

3.3.3 Other optic criteria

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. For both issues, the visibility criterion can be replaced by two different forms of transparency method.

The tip issue is illustrated in Figure 3.11(a): abnormal discontinuities in the stratigraphic function are observed when a coarse nodal distribution is used. The discontinuities are clearly marked thanks to the high resolution of the visualization grid and some additional iso-surfaces extracted from the implicit function. In Figure 3.11(b), the same nodal distribution is used but with a transparency

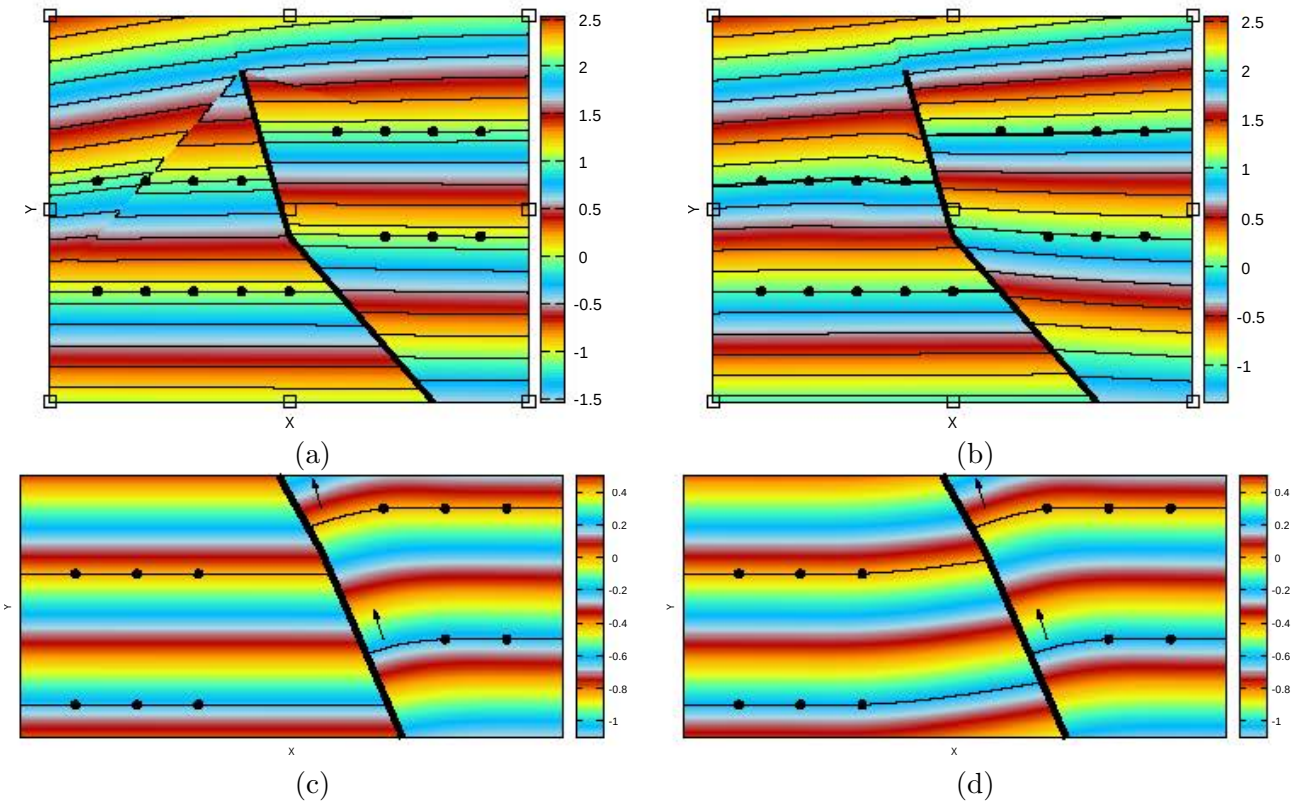


Figure 3.11: *The limits of the visibility criterion and the influence of the transparency method. (a) The visibility criterion creates unexpected discontinuities away from the fault tip. (b) The transparency method reduces the discontinuities. (c) The visibility criterion imposes two fault blocks to be independent one from another. (d) The transparency method ensures a discontinuous jump between the two fault blocks while preserving a mutual dependency on the structures.*
Iso-values from bottom to top: $[0, 1]$, $\lambda_\epsilon = 1$, $\lambda_{\boldsymbol{p}} = 1$.
(a) $\gamma = 2$ *and* $r_{tot} = 0.3L_F$ *with* L_F *the total length of the fault. (b)* $\tau = 0.001$.

method which equation (Equation (2.47), p.72) is repeated here:

$$r_{\overline{\boldsymbol{p}}}^{trpcy} = r_{\overline{\boldsymbol{p}}} + \rho_{\overline{\boldsymbol{p}}} \left(\frac{r_{tip}}{r_{tot}} \right)^{\gamma}, \ \gamma \geq 2, \tag{3.13}$$

with $r_{\overline{\boldsymbol{p}}}^{trpcy}$ the re-evaluated distance with the transparency, γ an empirically chosen parameter, and $\rho_{\overline{\boldsymbol{p}}}$ the dilatation parameter. The abnormal discontinuities are efficiently reduced, but not totally erased. The visibility criterion is generally sufficient to handle discontinuities with a relatively fine nodal discretization, but the transparency method may help at reducing the tip effect with a moderate nodal discretization.

Figure 3.11(c) illustrates the isolation effect caused by the visibility criterion: the interpolation in the two fault blocks are independent one from another. In Figure 3.11(d), the interdependency is established with the transparency method: the slope imposed by the gradient data also affects the other side of the fault. The criterion is here written as a reduction of the influence of any neighboring node $\overline{\boldsymbol{p}}$ on the other side of the fault as compared to a point $\boldsymbol{x}$ as

$$w_{\overline{\boldsymbol{p}}}(\boldsymbol{x}) = \tau \ w(\frac{|x - \overline{\boldsymbol{p}}_x|}{\rho_x}) \ w(\frac{|y - \overline{\boldsymbol{p}}_y|}{\rho_y}), \tag{3.14}$$

with $\tau \in [0, 1]$ the transparency coefficient. Equation (3.14) is an adaptation of the classical transparency method (Equation (3.13)) for a homogeneous transparency over all the discontinuity and not only in the vicinity of its tip. This reduction is written on the weight functions but could have been

Symbol	Meaning	Default value
$\#\boldsymbol{N}$	Total number of interpolation nodes	$2,500$
$\#\boldsymbol{N}_v$	Total number of visualization points	$10,000$
$\#\boldsymbol{D}$	Total number of data points	989
ρ	Dilatation parameter (adimensional)	1.99
$order(\mathcal{P})$	Polynomial order	1
$\#\mathcal{P}$	Number of monomials in the polynomial basis	3
$\alpha_{\boldsymbol{p}}$	Expected iso-values	Bottom to top horizons $[0, 0.13, 0.21, 0.32, 0.44, 0.51, 0.6, 0.73, 0.86, 1, 2, 2.09, 2.18, 2.27, 2.38, 2.5]$
λ_ϵ	Constant weight on energy equations	1
$\lambda_{\boldsymbol{p}}$	Constant weight on DPV equations	1

Table 3.1: *Parameters of the proposed method and their default values for the sensitivity tests.*

written on the distances as in Equation (3.13). As for γ, the coefficient τ must be empirically tuned and the relation between its value and the results is not intuitive. Further studies in this field could help normalizing γ and τ depending on the application.

3.4 Sensitivity tests

3.4.1 Reference model for the sensitivity tests

The tests are performed on a synthetic cross section of an eroded, faulted and folded domain in two dimensions (Figure 3.12). As in Figure 3.1, Figure 3.12(a) shows the used numerical data and illustrates the supports of the interpolation. Figure 3.12(b) shows the computed implicit function with extracted horizons. Equations (3.2) and (2.55) (p.81) are also reminded.

All the parameters used to create Figure 3.12(b) and their values are given in Table 3.1. The interpolation nodes used in the computation are actually more numerous than illustrated in Figure 3.12(a) 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, which reflects the anisotropy of the studied structures. The number of visualization points is purposely bigger than the number of nodes and data points (i.e., a grid of (300×300) is used for visualization, and (100×100) for tests and comparisons) 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.

Figure 3.12(b) is used as reference model for sensitivity analysis in the following. The parameters and their values are commented and tested separately. The values given in Table 3.1 are systematically used as default values for all parameters, only the tested parameters values are changed in each sensitivity test. The results are compared with a metric described below.

3.4.2 Model distance and data distance

Two types of distances are suggested in this thesis 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 $\boldsymbol{p} \in \boldsymbol{N}_v$. The reference values are the reference model's implicit function values (Figure 3.12(b)) evaluated at the corresponding visualization points $u_{ref}(\boldsymbol{p})$. The distance D_{model} is thus evaluated as

$$D_{model} = \frac{1}{\#\boldsymbol{N}_v} \sum_{\boldsymbol{p} \in \boldsymbol{N}_v} \frac{|u(\boldsymbol{p}) - u_{ref}(\boldsymbol{p})|}{||\overline{\boldsymbol{g}}||_{ref}}. \tag{3.15}$$

with $||\overline{\boldsymbol{g}}||_{ref}$ the average gradient norm of the reference model.

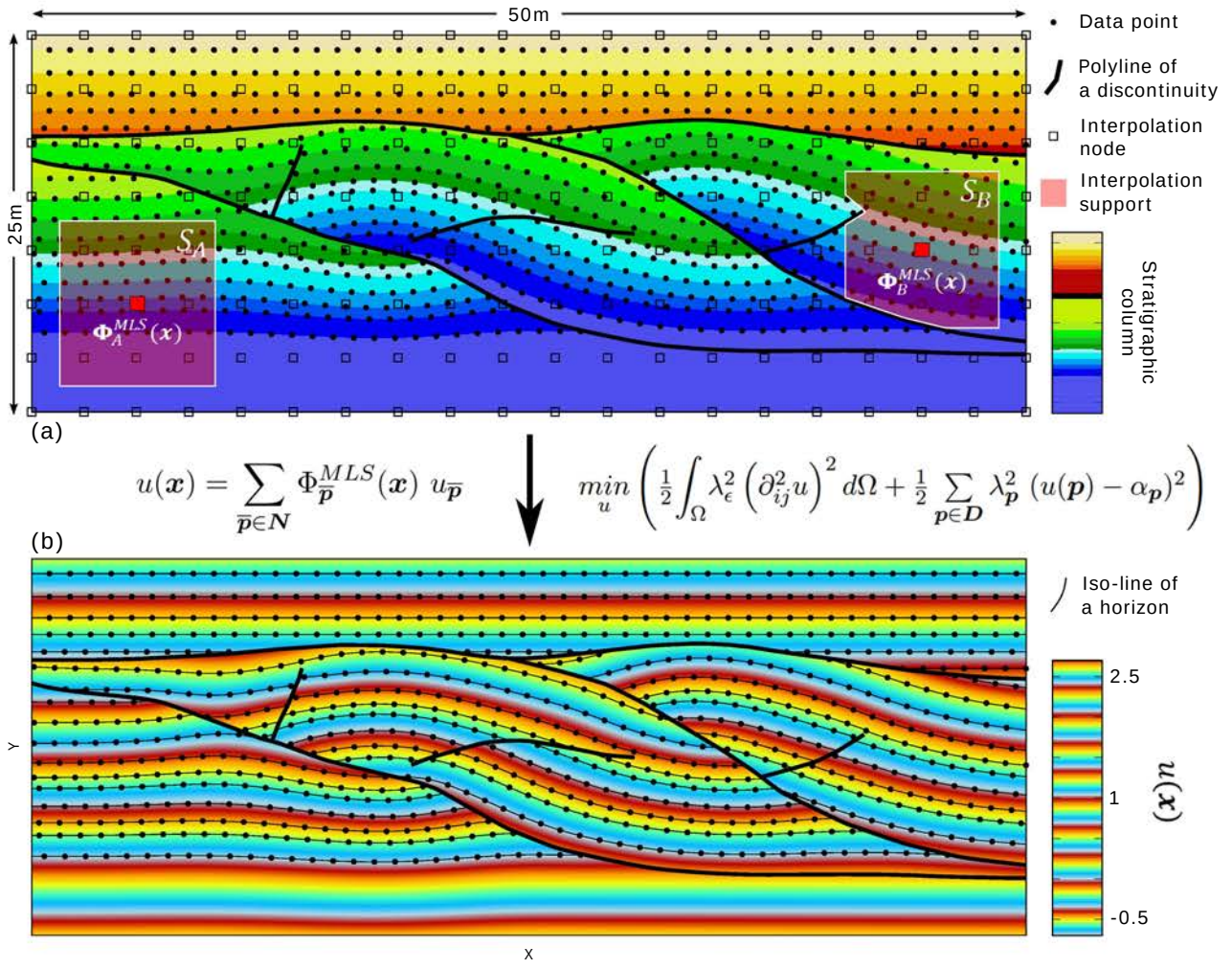


Figure 3.12: *Schematic representation of the proposed workflow. (a) A cross section densely and regularly interpreted with data points and fault and unconformity segments in 2D. (b) Computed implicit function u obtained with the parameter values of Table 3.1. Between (a) and (b) The two main equations of the method: the implicit function definition and the problem to solve. The nodal sampling to compute (b) is (50×50) although fewer nodes are illustrated for visibility.*

In D_{data}, the tested values are the implicit function values evaluated at the data points positions $\boldsymbol{p} \in \boldsymbol{D}$ of the currently tested model (e.g., sparse, noisy, depending on the application). The reference values are the expected data values $\alpha_{\boldsymbol{p}}$ (i.e., of the DPV constraints). The distance D_{data} is thus evaluated as

$$D_{data} = \frac{1}{\#\boldsymbol{D}} \sum_{\boldsymbol{p} \in \boldsymbol{D}} \frac{|u(\boldsymbol{p}) - \alpha_{\boldsymbol{p}}|}{||\overline{\boldsymbol{g}}||_{ref}}. \tag{3.16}$$

As the errors are normalized by the reference gradient norm $||\overline{\boldsymbol{g}}||_{ref}$, they are actual distances in meters comparable to the domain's dimensions (i.e., $(50m \times 25m)$), 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 $1m$ indicates that a value evaluated at a visualization point exists, on average, at one meter in the reference model; a D_{data} of $1m$ indicates that the average distance between each data point and the corresponding iso-line is equal to one meter.

3.4.3 Sensitivity to data quality

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 and quality.

3.4.3.1 Data sparsity

Depending on the types of field samples (e.g., seismic and wells) and the quality of the rock exposure, the interpreted data points for structural modeling may be more or less clustered and sparse (Section 1.1.4, p.26). Figure 3.13(a) shows the accuracy of the method for different degrees of random decimation of the reference data points (Figure 3.12(a)).

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. When a random parameter is involved, 100 simulations 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 the online resource of Renaudeau *et al.* (2019b), but not represented for visibility reasons.

In Figure 3.13(a), it can be observed that the fewer the data points, the worse D_{model}. On the contrary, D_{data} is more or less 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 (Section 2.2.5.7, p.87). Figure 3.13(b) shows a result with a data decimation value of 98%, which gave, in this random case, 14 data points. In the circled areas of missing data, the folds are smoothed, but the remaining of the model is well reconstructed. This figure also gives an idea of what a model with a few decimeters of error looks like as compared to the reference model.

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, a special attention should be given to acquire and interpret data on high curvature areas and areas of thickness variation.

3.4.3.2 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 (Figure 3.12(a)). 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 chosen using a uniform distribution between 0 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 3.14 shows the resulting models for three different values of the smoothness parameter λ_ϵ. As mentioned in Section 2.2.5.6 (p.86), it is difficult to have an *a priori* knowledge of a proper λ_ϵ value to use.

Figure 3.15 shows how the error evolves when the intensity of noise and the smoothing level change. For an intensity of noise fixed to 1 meter (Figure 3.15(a)), the best D_{model} value is obtained for a λ_ϵ around 30 (illustrated by Figure 3.14(b)). Below this range, the noisy data points are better represented (i.e., D_{data} decreases), which drives away the results from the reference model (i.e., D_{model} increases, illustrated by Figure 3.14(a)). Above this range, the structures start to be smoothed, which deteriorates the fitting (i.e., D_{data} and D_{model} slowly increase, illustrated by Figure 3.14(c)). For a λ_ϵ fixed to 30 (Figure 3.15(b)), both D_{data} and D_{model} increase together with the intensity of noise. As the noise increases, the data points represent less and less the geological structures, hence the observed trend. Although it is dependent on the data distribution, we observed that satisfactory results are generally obtained with a λ_ϵ between 1 and 100 for most data sets with a $\lambda_{\boldsymbol{p}}$ between 1 and 10.

Figure 3.16 is an example of the use of different types of data constraints to reproduce the reference model. The data points (DPV constraints, Section 2.2.3.1, p.80) were obtained using a decimation of

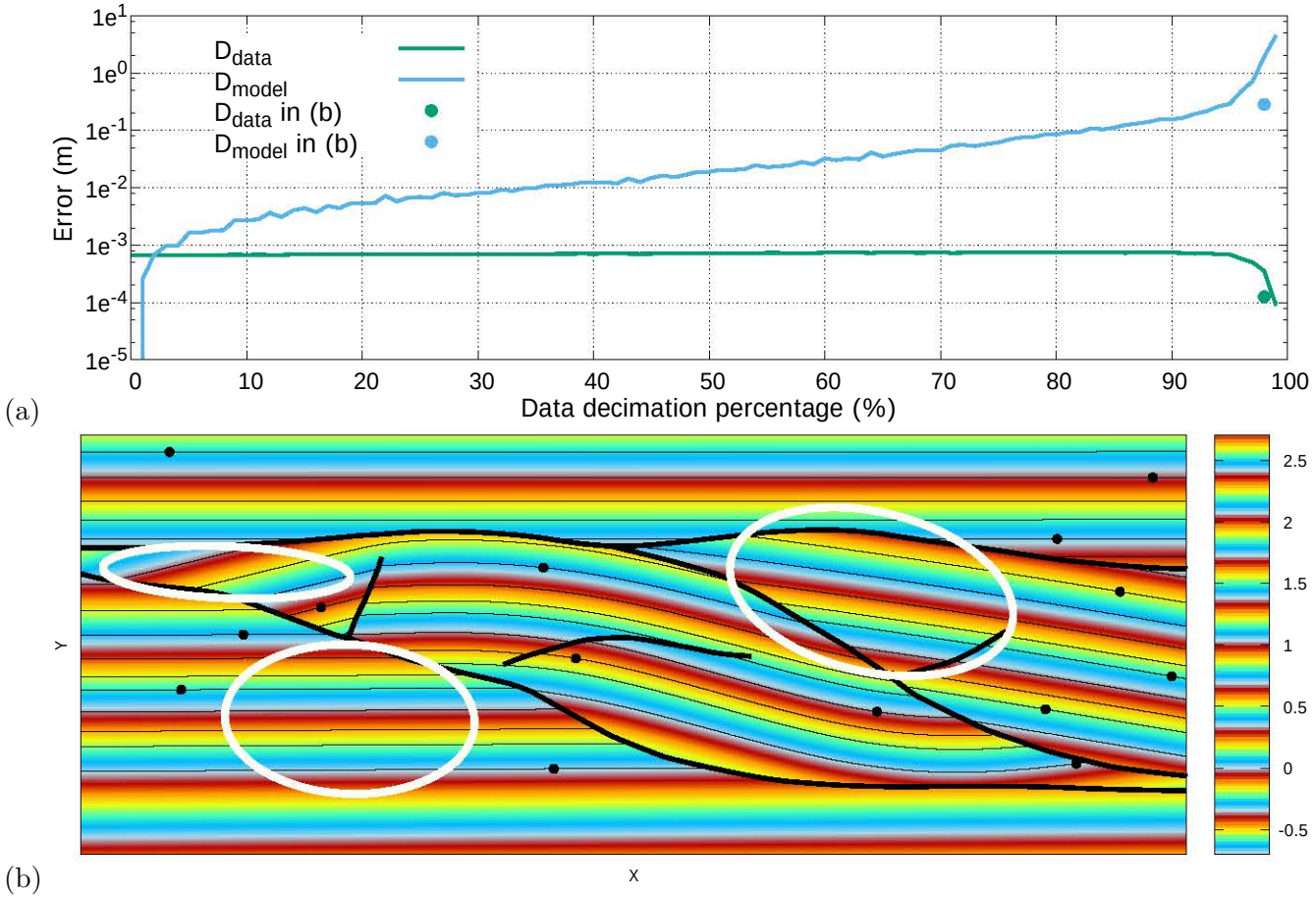


Figure 3.13: *Sensitivity of the proposed method to irregular data points. (a) Distances to the reference model and the decimated data (Section 3.4.2) with a varying random decimation in the data points. (b) A resulting implicit function obtained with a decimation value of* 98% *(i.e.,* 14 *data points). The white circles indicate where geological structures were lost as compared to Figure 3.12(b).*

98% and an intensity of noise of 1 meter, generating 13 noisy points. The hard data points (HDPV constraints, Section 2.2.5.5 (p.85)) were obtained using a decimation of 99%, generating 5 points positioned as in the reference data set. Here, the Lagrange multipliers were not used but the weights on the hard data were tuned. The 5 normal gradient data vectors (NGD constraints, Section 2.2.5.2, p.83) were extracted on the implicit function from the reference model to control the structures 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., up to $\approx$ 395 data points, from Figure 3.13(a)).

3.4.4 Sensitivity to the method's parameters

3.4.4.1 Normalizing the smoothing equations

In a least squares approach such as DSI, the number of equations has an influence on the results (Section 1.3.1.9, p.39). For a given data set, the number of equations changes with different nodal resolutions. Therefore, it may be necessary to tune the penalization weights λ_ϵ and $\lambda_{\boldsymbol{p}}$ for any different number of nodes $\#\boldsymbol{N}$ used for computation.

However, the terms of volumes ν in system (3.12), obtained during the discretization stage, normalize this dependency to $\#\boldsymbol{N}$. This is empirically proven in Figure 3.17 where D_{model} and D_{data} are evaluated on the perturbed data points of Figure 3.14 for different sets of regularly discretized nodes. Two cases are evaluated: (i) with the weight λ_ϵ scaled on ν, and (ii) with a fixed weight λ_ϵ. It can be

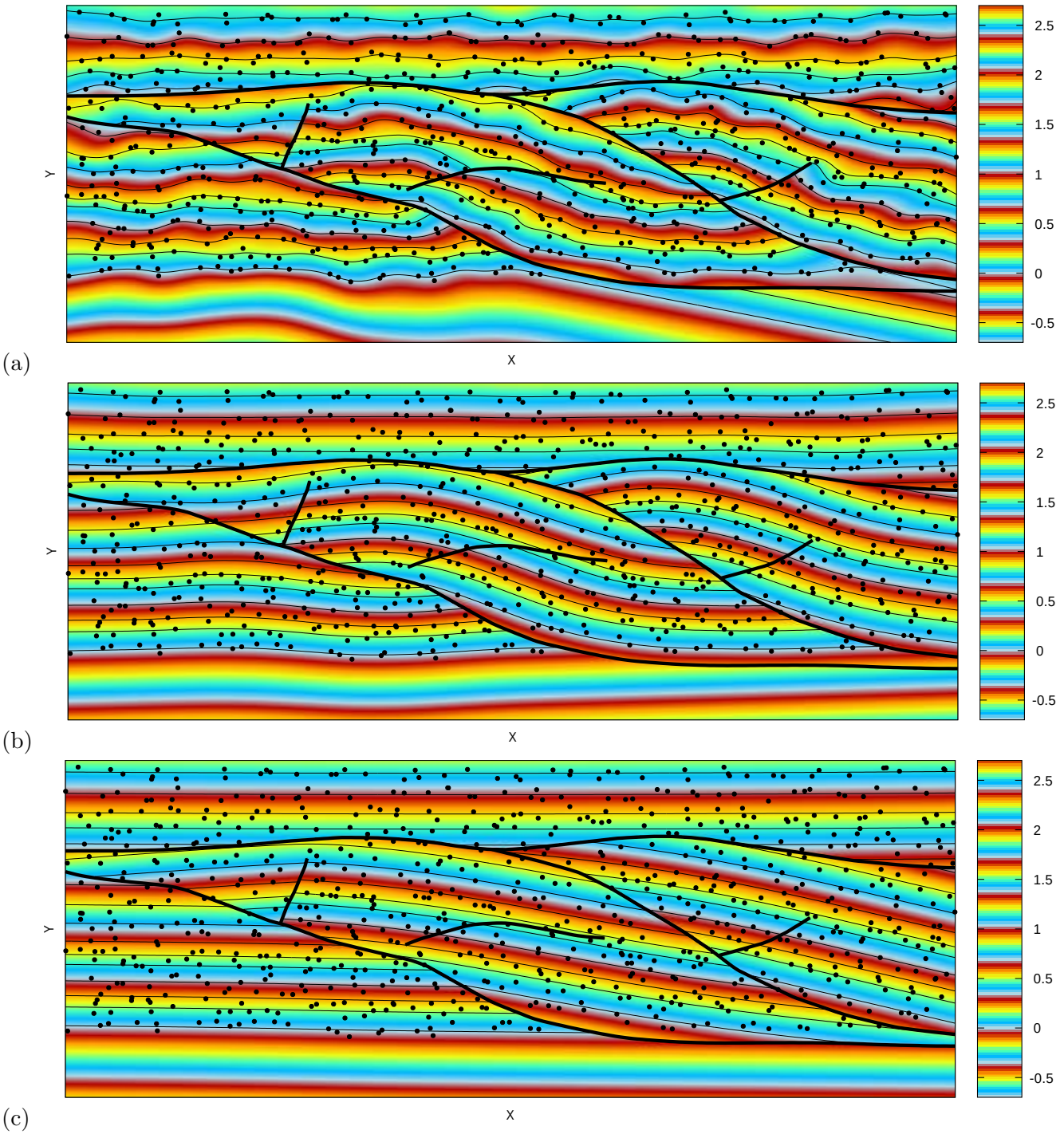


Figure 3.14: *Resulting implicit function when using the same set of perturbed data points (intensity of noise is* $1m$*), but with different energy weights* λ_ϵ*. (a)* $\lambda_\epsilon = 1$*. (b)* $\lambda_\epsilon = 30$*. (c)* $\lambda_\epsilon = 400$*.*

observed in Figure 3.17 that D^{ν}_{model} and D^{ν}_{data} converge while D^{fixed}_{model} and D^{fixed}_{data} diverge. This figure also shows that the algorithm takes a few seconds to produce a result on a thousand data (975) using thousands of nodes, and that the computation time is linearly dependent on the number of nodes. These numbers include the evaluation on the (100×100) visualization grid and the computation of the distances.

An equivalent principle should also be applied on the $\#\boldsymbol{D}$ data equations as presented in Appendix F (p.170). In practice, the tests on data sparsity using the random decimation show that the dependency to the data equations is, in this study, negligible (see D_{data} in Figure 3.13(a)).

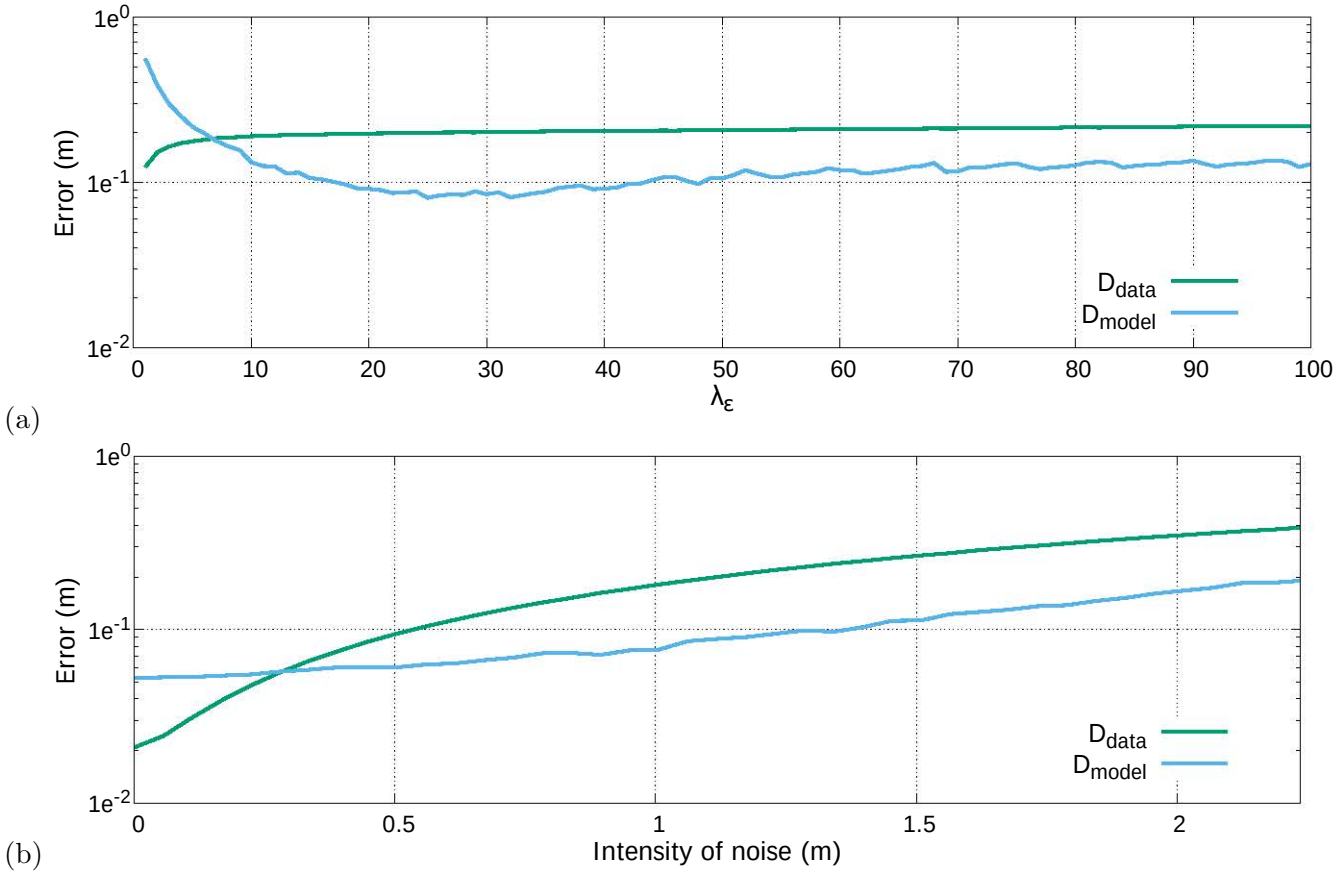


Figure 3.15: *Sensitivity of the proposed method to the noise and the energy weight* λ_ϵ. *(a) The intensity of noise is fixed to* $1m$, *and* λ_ϵ *varies. (b) The weight* λ_ϵ *is fixed to* 30 *and the intensity of noise varies.*

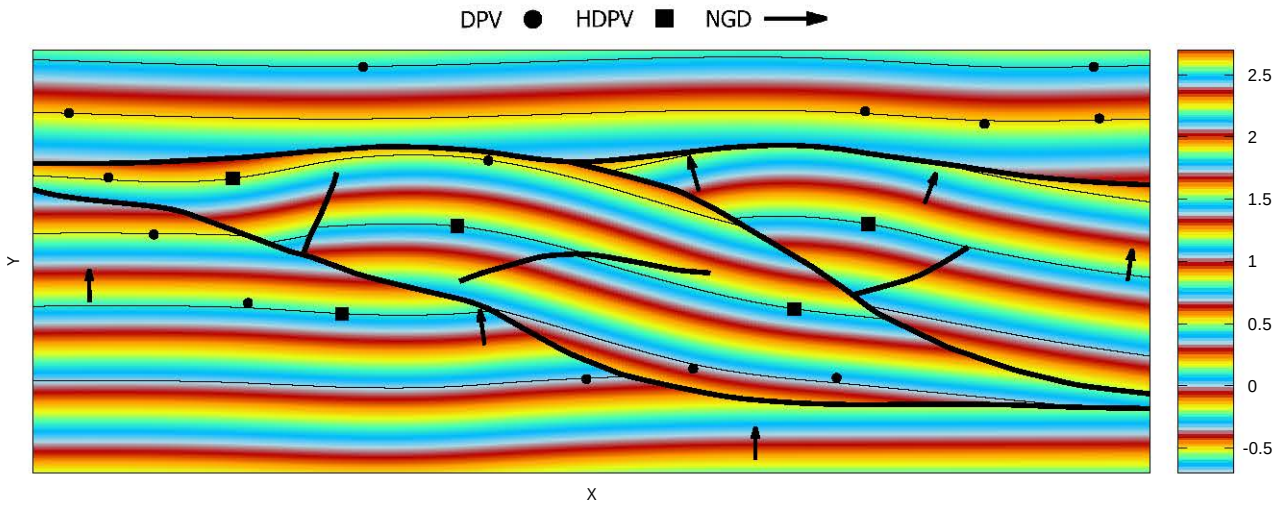


Figure 3.16: *A resulting implicit function using sparse and heterogeneous data with noise in the soft data (intensity of noise:* $1m$*) and a few hard and normal gradient data.* $\lambda_{\boldsymbol{p}}^{HDPV} = 10$, $\lambda_{\boldsymbol{p}}^{DPV} = 1$, $\lambda_{\boldsymbol{p}}^{NGD} = 10$, $\lambda_\epsilon = 30$. $D_{data} = 8.3e^{-3}m$ *and* $D_{model} = 2.9e^{-2}m$.

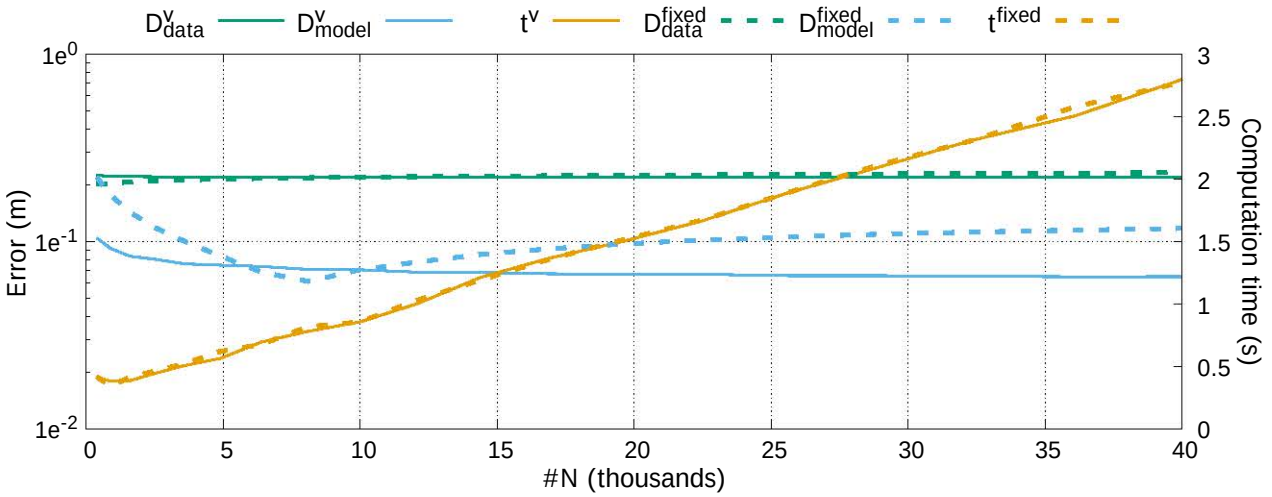


Figure 3.17: *Sensitivity of the proposed method to the number of computation nodes $\#\boldsymbol{N}$ on the data points from Figure 3.14. The distances D^{ν}_{model}, D^{ν}_{data} and computation times t^{ν} were computed with a λ_ϵ dependent on the discretization (i.e., $\lambda_\epsilon = 30\sqrt{\nu}$). The distances D^{fixed}_{model}, D^{fixed}_{data} and computation times t^{fixed} were computed with a λ_ϵ independent of the discretization (i.e., fixed to $\lambda_\epsilon = 95 \Leftrightarrow 30\sqrt{\nu}$ when $\#\boldsymbol{N} = 10.000$).*

3.4.4.2 Complexity and stability of the moving least squares functions

Polynomial order of the MLS functions

The polynomial order $order(\mathcal{P})$ defines, together with the dimensional space (e.g., two here), the number of monomials $\#\mathcal{P}$ in the polynomial basis. This number $\#\mathcal{P}$ describes the dimensions of the moment matrix $\boldsymbol{A}$ (i.e., $dim(\boldsymbol{A}) = [\#\mathcal{P} \times \#\mathcal{P}]$). It thus has an influence on the complexity of the method and must be chosen small enough to avoid unnecessary computational costs. Figure 3.18(a) 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 (Figure 3.18(b)), with a higher computational complexity. The order $order(\mathcal{P})$ is therefore fixed to 1 in the presented method. Although purely empirical, first order MLS functions seem to reproduce any complex geological structures.

Local supports and cover problem

An MLS shape function $\Phi_{\overline{\boldsymbol{p}}}(\boldsymbol{x})$ is only defined within the support $S_{\overline{\boldsymbol{p}}}$ of its weight function $w_{\overline{\boldsymbol{p}}}$. By definition, $S_{\overline{\boldsymbol{p}}}$ and the support of $\Phi_{\overline{\boldsymbol{p}}}(\boldsymbol{x})$ are the same. The node $\overline{\boldsymbol{p}}$ has an influence on the points existing within the support $S_{\overline{\boldsymbol{p}}}$. If a point $\boldsymbol{x}$ is influenced by a node $\overline{\boldsymbol{p}}$, then $\overline{\boldsymbol{p}}$ is said to be a neighbor of $\boldsymbol{x}$.

This restricted influence of the nodes $\overline{\boldsymbol{p}}$ in space represents the main advantage of the MLS functions as each constraint involves a small number of neighbors (i.e., the system (3.12) is sparse). Unfortunately, it may also lead to singularities. Let the intersection of all supports $S_{\overline{\boldsymbol{p}}}$ be denominated as the cover. When the entire domain Ω is included in the cover, it is said to be complete (Section 2.1.3.1, p.64). If the cover is incomplete, then the implicit function $u(\boldsymbol{x})$ is undefined in the uncovered areas. In addition, the number of linearly independent neighboring nodes required around a position $\boldsymbol{x}$ should be equal at least to the number of monomials $\#\mathcal{P}$ in the MLS polynomial basis to have a non-singular matrix $\boldsymbol{A}$.

Therefore, the distribution of the nodes $\overline{\boldsymbol{p}}$ and the size of the supports $S_{\overline{\boldsymbol{p}}}$, controlled by the dilatation parameter ρ, must be defined carefully to avoid singularities. This situation is analogous to ordinary and universal kriging when the neighborhood size is too small.

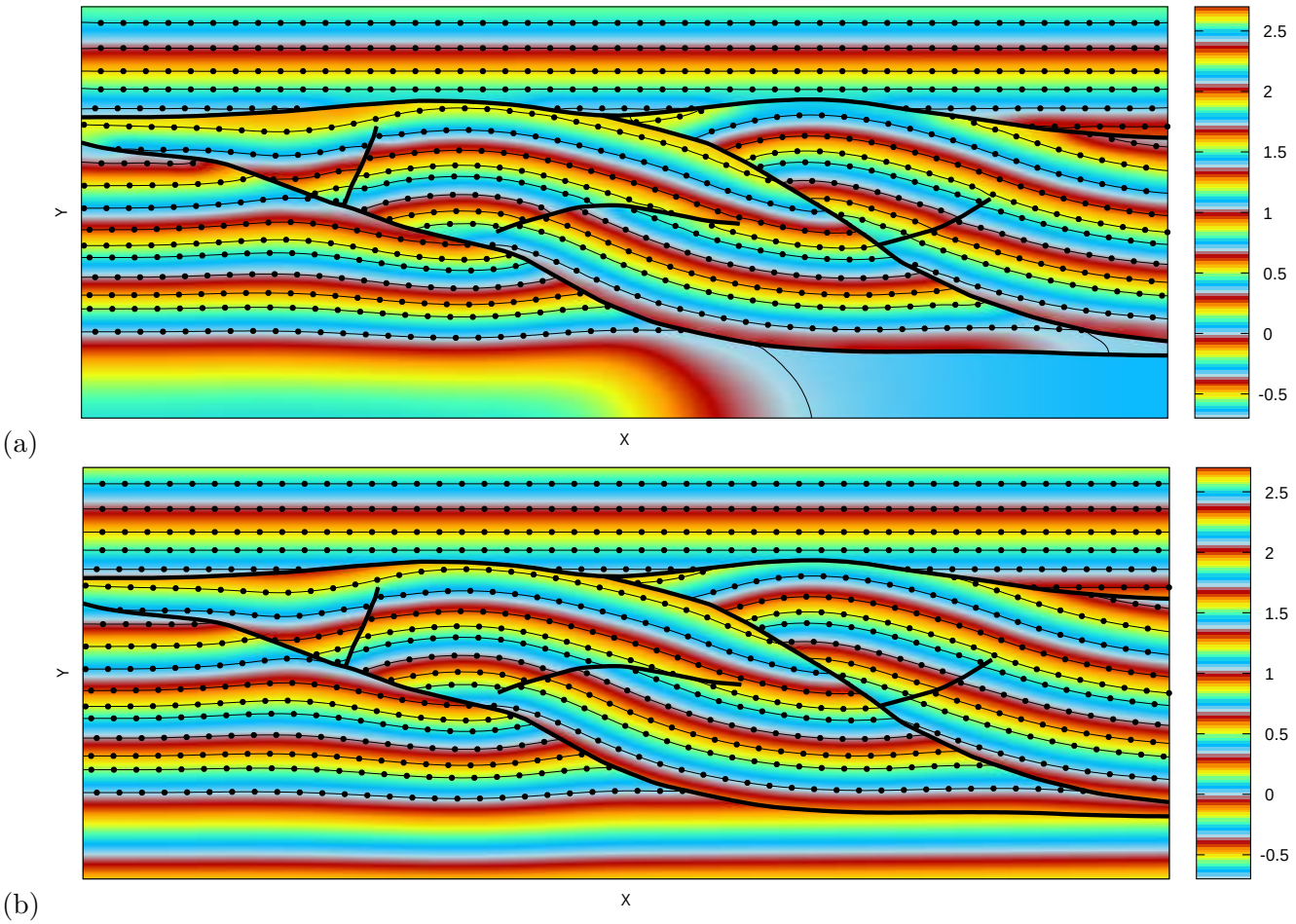


Figure 3.18: *Resulting implicit function with different polynomial orders order($\mathcal{P}$) for the MLS shape functions and adapted dilatation values ρ. (a) $d = 0$ and $\rho = 1.99$. (b) $d = 2$ and $\rho = 2.99$.*

Theoretical relationship between the dilatation parameter and the number of neighbors

The cover problem (Section 3.4.4.2) is tackled with squared supports (Equation (2.27), p.66)) and by making the dilatation parameter ρ proportional to the regular spacing of interpolation nodes (Section 3.2.3.1). Figure 3.19(a) illustrates the relationship between ρ and the resulting support of an MLS function centered on a given interpolation node $\overline{\boldsymbol{p}}$. Close to a discontinuity with the visibility criterion (Section 2.1.4.1, p.72) or close to a border, the support may cover less neighboring nodes than in regular cases. The dilatation parameter ρ has thus a direct impact on the maximum number of neighboring interpolation nodes n_{node}^{max} around a given node $\overline{\boldsymbol{p}}$, following

$$n_{node}^{max} = (2\lfloor\rho\rfloor + 1)^2, \tag{3.17}$$

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

As ρ is constant for all nodes, if an MLS support centered on a node $\overline{\boldsymbol{p}}$ covers a point $\boldsymbol{x}$, then an imaginary MLS support centered on $\boldsymbol{x}$ covers $\overline{\boldsymbol{p}}$. Figure 3.19(b) illustrates the relationship between ρ and the support of an MLS function centered on a data point $\boldsymbol{x}$, defining the neighboring nodes influencing this datum. 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 $\boldsymbol{x}$ is

$$n_{data}^{max} = (2\lfloor\rho + 0.5\rfloor)^2. \tag{3.18}$$

Equation (3.18) 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.

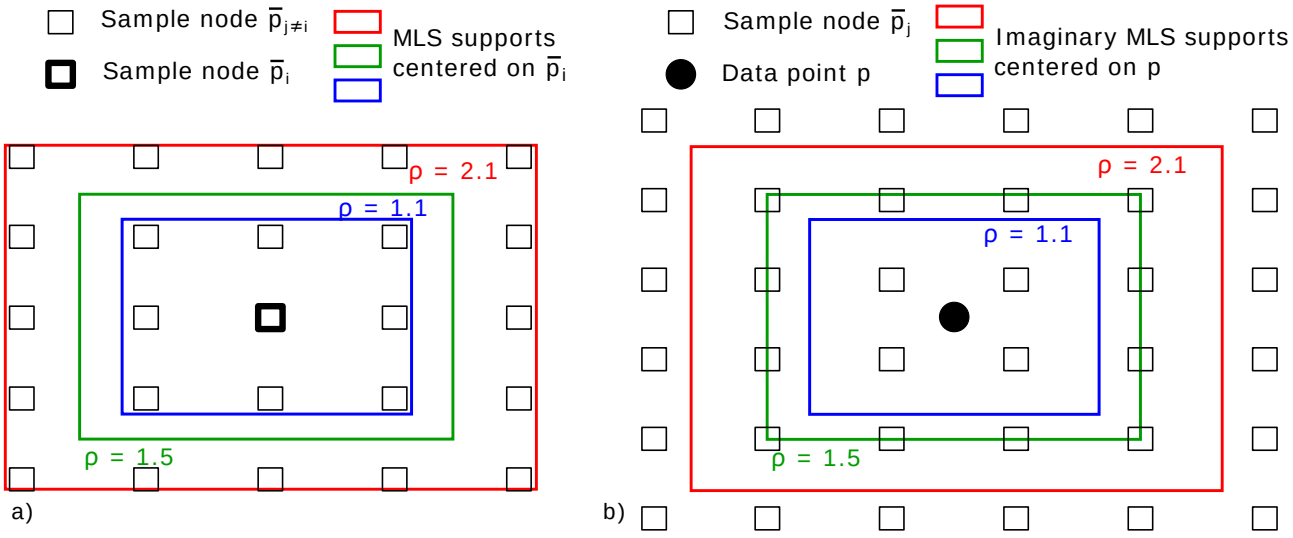


Figure 3.19: *Illustration of the relationship between the dilatation parameter ρ, proportional to the regular spacing, and the MLS functions supports. (a) The supports for three different ρ values centered on a node $\overline{\boldsymbol{p}}_i$. (b) The imaginary supports for the three same ρ values centered on a data or visualization point $\boldsymbol{x}$.*

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 (3.12). In addition, both of them must at least be greater than the number of monomials $\#\mathcal{P}$, which is given by the dimensional space and the polynomial order $order(\mathcal{P})$ (Section 3.4.4.2).

Practical influence of the support size on the method

Even if the theory defines a minimum value for the dilatation parameter ρ as compared to the MLS parameter $\#\mathcal{P}$ (Section 3.4.4.2), this minimum value is not necessarily reliable when discontinuities are present. This is illustrated in Figure 3.20 where the influence of ρ on the method is given for a polynomial order of 1 (i.e., $m = 3$ in two dimensions).

If the moment matrix $\boldsymbol{A}$ is singular at a position $\boldsymbol{x}$, the implicit function u cannot be evaluated. In this case, the concerned point (i.e., data or visualization point) is avoided during the computation of distances from Section 3.4.2 and flagged as undefined. Such anomalies are not explicitly represented on Figure 3.20 for visibility reasons, but their numbers per model can be found in the online resource of Renaudeau *et al.* (2019b).

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 $\boldsymbol{A}$ (i.e., $n_{data}^{max} = 4 > 3$, $n_{node}^{max} = 9 > 3$). This is related to the visibility criterion (Section 2.1.4.1, p.72): it reduces the number of neighbors by cutting the supports near the discontinuities, and therefore modifying the perfect cases drawn in Figure 3.19. In this example, the number n_{data}^{max} is too close to $\#\mathcal{P}$ and the number of actual neighbors is likely to drop under $\#\mathcal{P}$ if a data point is close to a discontinuity. No other undefined points are found in the other models of Figure 3.20.

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., $D_{model} = 0$ around a ρ value of 1.99) is caused by the exact equivalence with the reference parameters (Table 3.1, default values).

Concerning the computational time t, it increases incrementally with ρ. The computational cost of the method is therefore dependent on n_{node}^{max} as each unit of ρ defines a different number of neighboring nodes (Equation (3.17)). It is also dependent on n_{data}^{max} (Equation (3.18)), though little observable on Figure 3.20. 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.

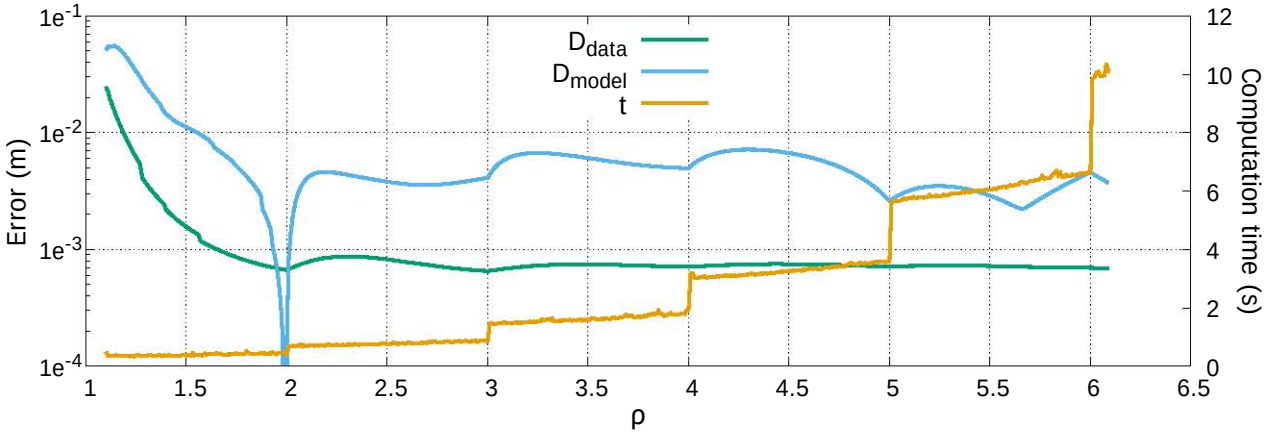


Figure 3.20: *Influence of the support size ρ on the method when ρ is proportional to the regular spacing and with a polynomial order $order(\mathcal{P})$ of* 1.

In conclusion of this study, 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 3.1) 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 of 2.99 was used for a polynomial order $order(\mathcal{P})$ of 2 in Figure 3.18(b).

3.4.4.3 Regular or irregular sampling

Distributing the interpolation nodes regularly (Section 3.2.2, p.92) is not a requirement of the proposed method, but it has several advantages. In Figure 3.21, 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_x and ρ_y cannot be specified relatively to the interpolation node spacing (e.g., not as in Equation (3.11)). Therefore, they are fixed for all simulations, regardless of the number of nodes $\#\boldsymbol{N}$ and the sampling technique. In this application, ρ_x is fixed to 3.5 meters, and ρ_y is fixed to 1.75 meters, so that each support $S_{\overline{\boldsymbol{p}}}$ of interpolation covers $\approx 0.5\%$ of the domain Ω.

The model distance D_{model} and data distance D_{data} (Section 3.4.2) are represented respectively by D^{reg}_{model} and D^{reg}_{data} for regular sampling, and D^{rand}_{model} and D^{rand}_{data} 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 the online resource of Renaudeau *et al.* (2019b).

Figure 3.21 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) for the same number of nodes, regular sampling always gets models closer to the reference model and the data set than the average random sampling; and (iv) the computational efficiencies of the two methods are equivalent. Not observable in the figure but in the online resource of Renaudeau *et al.* (2019b): the two methods can generate undefined points for small numbers of nodes $\#\boldsymbol{N}$ (up to 900; small as 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_x and ρ_y, both sampling techniques obtain close results as long as a minimum number of interpolation nodes $\#\boldsymbol{N}$ is used. The main difference is that the number of nodes $\#\boldsymbol{N}$ and the dilatation parameters ρ_x and ρ_y can be theoretically correlated to avoid singularities with regular sampling (Section 3.4.4.2), which is not the case with random sampling. In practice, this correlation also avoids unnecessary large supports of interpolation and thus reduce the computation time with regular sampling (Section 3.4.4.2). In Figure 3.17 for instance, it takes 0.9 seconds to generate a model using $10,000$ nodes and a dilatation parameter ρ of 1.99 scaled on the nodal spacing (Equation (3.11)). If performed on non-perturbed data points, the evaluated distances are $D_{data} = 5.14e^{-4}m$ and $D_{model} = 1.56e^{-3}m$.

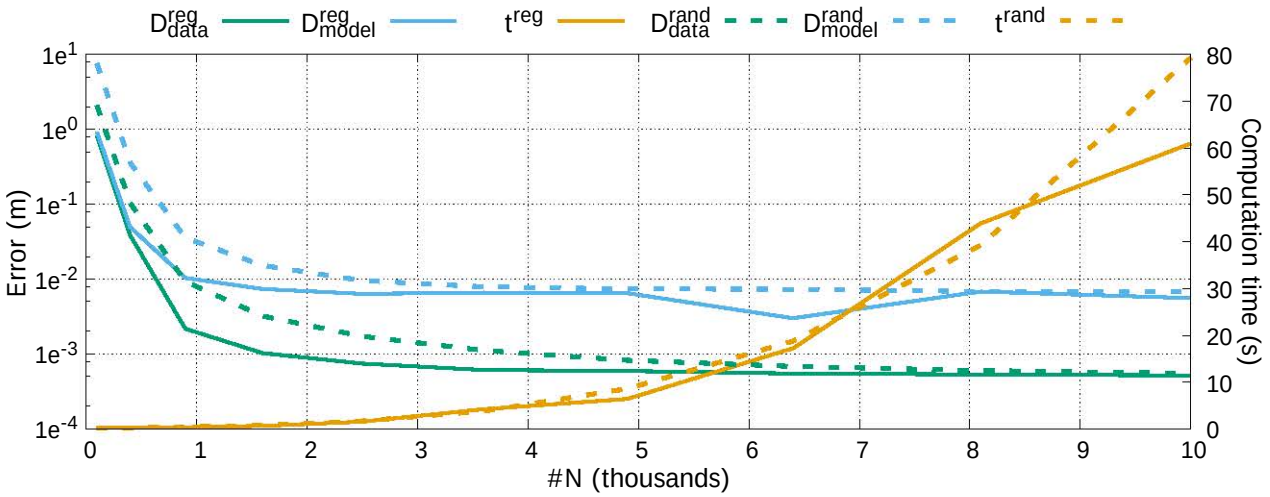


Figure 3.21: *Influence of the number of interpolation nodes $\#\boldsymbol{N}$ on the presented method with regular and random sampling. The distances and the computation time are evaluated for both random and regular sampling separately.*

3.4.4.4 Complex geometries of structural discontinuities

Lack of neighboring nodes

Although the visibility criterion (Section 2.1.4.1, p.72) is criticized for stability reasons (Belytschko *et al.* (1996a), Section 2.1.4.1, p.72), 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 may produce areas not covered by the MLS functions and/or areas with singularities in the interpolation.

In Figure 3.22(a), 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 $\boldsymbol{A}$ singular in the concerned area (Section 3.4.4.2). In Figure 3.22(b), in addition to singularities, the generated implicit function bends abnormally away from the intersection. The unevenly distributed nodes also have deteriorated the evaluation of the second derivatives which has impacted the solution coefficients $u_{\overline{\boldsymbol{p}}}$ attached to the concerned nodes $\overline{\boldsymbol{p}}$.

Those results were generated with the default parameters (Table 3.1) and smaller numbers of interpolation nodes $\#\boldsymbol{N}$. 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. 3.20 and 3.21 show that both of these solutions are possible but costly.

Changing the polynomial order

An alternative strategy to address the lack of neighbors could be to locally reduce the polynomial order $order(\mathcal{P})$: if the number of interpolation neighbors of a point is smaller than the chosen number $\#\mathcal{P}$, $order(\mathcal{P})$ can be decreased accordingly.

Unfortunately, this solution is not applicable as it does not solve situations with no neighbors (Figure 3.22(a)) or with unevenly distributed neighbors (Figure 3.22(b)). Also, decreasing $order(\mathcal{P})$ to the zeroth order, if necessary, is not adapted to a structural modeling application (Section 3.4.4.2, Figure 3.18).

Generation of new nodes

In Section 3.4.4.3, 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 $\boldsymbol{x}$, the evaluation of the MLS functions and their derivatives

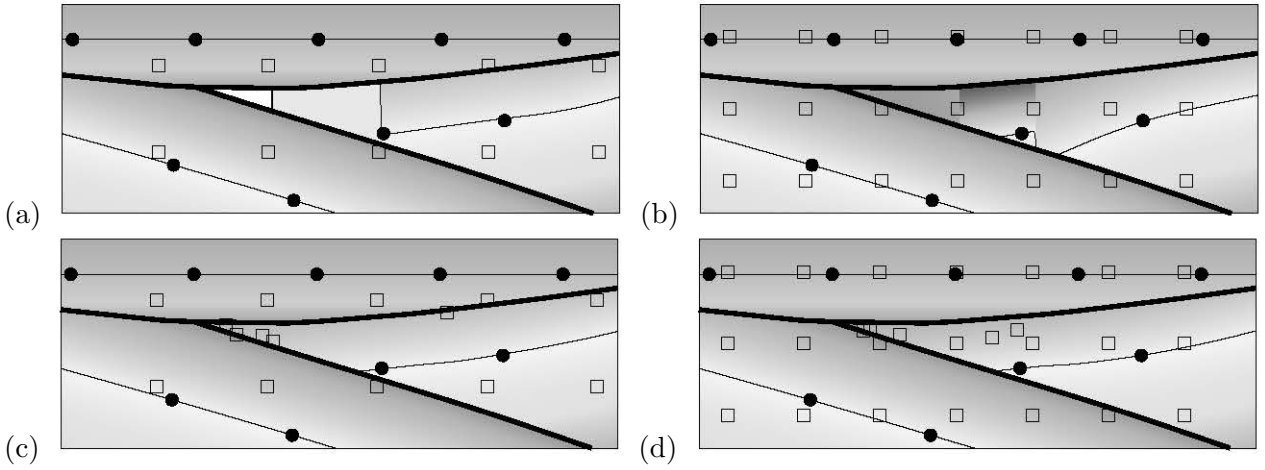


Figure 3.22: *Limits to the visibility criterion and a proposed correction. (a) Case of singularities and undefined values at visualization point positions when neighboring interpolation nodes are missing. (b) Case of badly evaluated nodal coefficients $u_{\overline{\boldsymbol{p}}}$ when neighboring interpolation nodes are unevenly distributed. (c) and (d) Corrections to the limits observed in (a) and (b) by randomly generating new neighbor nodes.*

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 $\boldsymbol{x}$ (interpolation node, data point or visualization point) is close to a discontinuity, we suggest the following procedure:

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

Figures 3.22(c) and (d) show how this technique solves the two problems exposed in Section 3.4.4.4 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 $\boldsymbol{x}$. This could increase the chances to obtain a stable interpolation in those areas with a small n_{ref} number, which is not guaranteed here. Adding interpolation nodes changes the problem's dimensions and its density (system (3.12)), but this technique does not change the computational efficiency of the algorithm because the modifications are local.

3.5 Perspectives and optimizations

3.5.1 Finite difference method

In this proposed discretization of this thesis's framework, the smoothing equations reduce to each second derivative ∂^2_{ij} evaluated at each computation node, and written in a separate row of system (3.12). With a regular sampling, approximating these equations by finite differences seems straightforward. The difficulty is that the MLS functions do not have the KΔP. Therefore, approximating the second derivatives by finite differences requires the evaluation of the implicit function at each neighboring node in each equation, which involves much more coefficients than the finite difference method expressed on a Cartesian grid.

For instance, the second derivative in the x axis is written

$$\partial^2_{xx} u(\overline{\boldsymbol{p}}) \approx \frac{u(\overline{\boldsymbol{p}}_x + \Delta_x, \overline{\boldsymbol{p}}_y) - 2u(\overline{\boldsymbol{p}}_x, \overline{\boldsymbol{p}}_y) + u(\overline{\boldsymbol{p}}_x - \Delta_x, \overline{\boldsymbol{p}}_y)}{\Delta_x^2}, \tag{3.19}$$

with Δ_x the nodal spacing in the x axis. When the shape functions $\Phi_{\overline{\boldsymbol{p}}}$ do not have the KΔP, the finite differences cannot be written with the neighboring coefficients $u_{\overline{\boldsymbol{p}}+\Delta_x}$ and $u_{\overline{\boldsymbol{p}}-\Delta_x}$ as

$$\partial^2_{xx} u(\overline{\boldsymbol{p}}) \neq \frac{u_{\overline{\boldsymbol{p}}+\Delta_x} - 2u_{\overline{\boldsymbol{p}}} + u_{\overline{\boldsymbol{p}}-\Delta_x}}{\Delta_x^2}. \tag{3.20}$$

In practice, we made the approximation that the MLS functions do have the KΔP (only for the finite difference method). This way, only the neighboring coefficients are involved in the approximation of the second derivatives of u. When applied on the reference data (Figure 3.12) with (50×50) nodes, the evaluated distances are $D_{model} = 9.06e^{-3}m$ and $D_{data} = 7.99e^{-4}m$. In 2D, the number of nodes $\#\boldsymbol{N}$, and thus the number of smoothing equations, is relatively low so the gains on the computational efficiency are moderate: both the computation and the evaluation on the (100×100) visualization grid took $0.45s$ with finite differences instead of $0.52s$ with the conventional derivatives. In 3D, where the number of nodes is much bigger, the gains are consequent. For instance, comparable implicit functions are computed in $10s$ instead of $25s$ with the data of Figure 3.7(a), and in $18s$ instead of $75s$ with the data of Figure 3.7(b).

3.5.2 Other possible optimizations

Once an adimentional dilatation parameter value ρ is chosen, 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, Section 2.1.4.1, p.72) have the same pattern of neighbors (Figure 3.19(a)). The MLS second derivatives evaluated at these nodes, or their finite difference approximations, are therefore equal one to another. 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 the borders and the discontinuities, the MLS functions are identical one to another (Section 3.2.3.2). Each MLS function $\Phi_{\overline{\boldsymbol{p}}}$ could thus be approximated by studying their evolution depending on the position $\boldsymbol{x}$ in the vicinity of their center node $\overline{\boldsymbol{p}}$. For instance, it can be observed in Appendix H (p.172) (Figures H.3(a) and H.5(a)) that for any $\boldsymbol{x}$ around the center node $\overline{\boldsymbol{p}} = \{0, 0\}$, the weight function $w_{\overline{\boldsymbol{p}}}$ is approximately 0.4 times the MLS function $\Phi_{\overline{\boldsymbol{p}}}$ when the dilatation ρ is of 1.99. Although not perfectly equivalent, this normalized function is less computationally demanding than the MLS function and may replace it during the evaluation of data and visualization points. The previous comments for patterns on 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 (3.12) are independent one from another. When considering a node or a data point, the set of neighbors is defined with the chosen support size and the proximity to the 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 (Section 2.1.4.1, p.72) does not require any heavy preprocessing on the sampling, but only intersection tests between segments (and triangles in three dimensions). Therefore, each equation in system (3.12) can be written in parallel.

In the current version of the code, these intersection tests on triangles in 3D are performed in repetitive loops to ensure that no triangle is missed regardless of the nodal distribution and of the resolution of the triangulated surfaces. As this is a research code, we did not judge the optimization of these tests as necessary, but it greatly slows the computation as well as the evaluation on the visualization grid.

3.5.3 Adaptive sampling resolution with a varying dilatation parameter

The aim is to avoid the homogeneous fine resolution of regular sampling everywhere in the domain Ω. Local details could still be captured with fine nodal resolutions while the number of nodes could

be reduced in areas with little complexity. For this, we could consider to use a distribution with a varying resolution in space, and adapt the dilatation parameter accordingly.

In the Element-Free Galerkin method (EFG), several techniques of node distributions for the MLS functions have been proposed. These include, for instance, regular and random samplings (Belytschko *et al.*, 1994b), uniform samplings based on the geometry of the studied object (Beissel & Belytschko, 1996; Yavari *et al.*, 2001), re-samplings close to complex features and discontinuities (Belytschko *et al.*, 1996b), and samplings using density maps and other spacing functions (Müller *et al.*, 2004). In the latter case, the related supports of interpolation may be defined as spherical supports (i.e., Equation (2.26), p.66) with a varying dilatation parameter based on the density maps.

The difficulty with such an approach is that MLS functions are not only constructed on their center node, but also on their neighboring nodes. Therefore, constructing a function on an isolated node does not only require to increase the size of this node's support, but also of its neighbors. Performant sampling techniques varying smoothly the density of nodes in space while ensuring local uniform distributions should be considered (Slak & KOSEC, 2019).

Résumé du Chapitre 4

Dans ce quatrième chapitre, nous présentons une seconde discrétisation du problème continu proposé pour la modélisation structurale.

Le domaine est discrétisé sur une grille Cartésienne sur laquelle sont communément construites les fonctions de forme et les fonctions de test. Pour cela, nous utilisons les fonctions bilinéaires issues de la méthode des éléments finis construites sur des quadrilatères en 2D et des hexaèdres en 3D. Comme ces fonctions ne sont pas suffisamment dérivables pour écrire les équations du problème continu sur les sommets de cette grille, nous les approximons par différences finies. Enfin, nous introduisons le concept de nœud fantôme pour la gestion des discontinuités (i.e., failles et discordances).

Les nœuds fantômes sont créés par duplication des sommets de la grille dans les cellules comportant au moins une surface de discontinuité. Ces nœuds additionnels sont associés à des degrés de liberté supplémentaires. La fonction implicite est alors évaluée avec différents degrés de liberté de part et d'autre de la discontinuité (i.e., certains associés aux sommets de la grille et certains associés aux nœuds fantômes). La décision concernant les degrés de liberté à utiliser lors de l'évaluation repose sur des tests d'intersections entre segments et surfaces de discontinuité. Les schémas de différences finies sont également adaptés à ce concept pour garantir l'utilisation de tous les degrés de liberté dans le système d'équations.

La méthode est illustrée sur des modèles synthétiques 2D et comparée à la méthode présentée dans le Chapitre 3. À des nombres de degrés de liberté comparables, cette méthode produit des résultats moins continus que celle utilisant les moindres carrés glissants, mais elle les produit plus rapidement. Cette méthode peut également être adaptée à tout autre type de maillages sur lesquels les équations utilisée continues peuvent être approximées.

Chapter 4

Cartesian grid discretization with ghost nodes

4.1 Motivations

In the previous chapter (p.91), we have suggested a first discretization using Moving Least Squares functions for this thesis's continuous framework of implicit structural modeling (defined in Chapter 2, p.59). In this chapter, we propose a second discretization of this framework with two objectives: (i) show that the same problem can be solved with a different set of numerical methods, and (ii) present an alternative for mesh-based methods to handle structural discontinuities.

This work is part of a cooperation project with Modeste Irakarama and Gautier Laurent (RING, Lorraine University). The initial idea was to create a fast algorithm to compute stratigraphic functions. The chosen method is presented in Irakarama *et al.* (2018a,b). It can be classified as a Discrete Smooth Interpolation technique: it discretizes the interpolation on a Cartesian grid with FEM shape functions and penalizes the data point equations by a smoothing factor in the least squares sense. This factor is written in 1D as the finite difference operator of second derivatives. In higher dimensions, this 1D scheme is written in all directions of the Cartesian grid (i.e., along the axes and the diagonals of the grid's cells). As the neighboring nodes are involved in the finite difference operators, the smoothing equations are not written on the border nodes. The evaluation of the stratigraphic function is also avoided within the cells cut by discontinuities, and the smoothing factor is not written on the corner nodes of those cells.

We adapted this approach within the framework of this thesis (Section 2.2, p.79). The first reason for this was to have a second discretization of the continuous problem for comparison. As compared to the above Cartesian grid method, this discretization includes volumetric weights in the least squares system, obtained during the integral discretization. It also explicitly uses the finite difference operators of second cross derivatives to write the bending energy equations. The second reason was to solve a limit of the above method where the data points contained in cells cut by discontinuities are not considered. To consider all data points, the resolution of the grid may then be controlled by the smallest distance between data points and discontinuity surfaces. As the grid resolution should better be scaled on the structures curvature, the method proposed hereafter handles stratigraphic discontinuities with meshless techniques (Section 2.1.4, p.72) to evaluate the function close to the discontinuities even with a relatively coarse resolution.

The presented method is the subject of the published paper Renaudeau *et al.* (2019a). This chapter is an extended version of this paper.

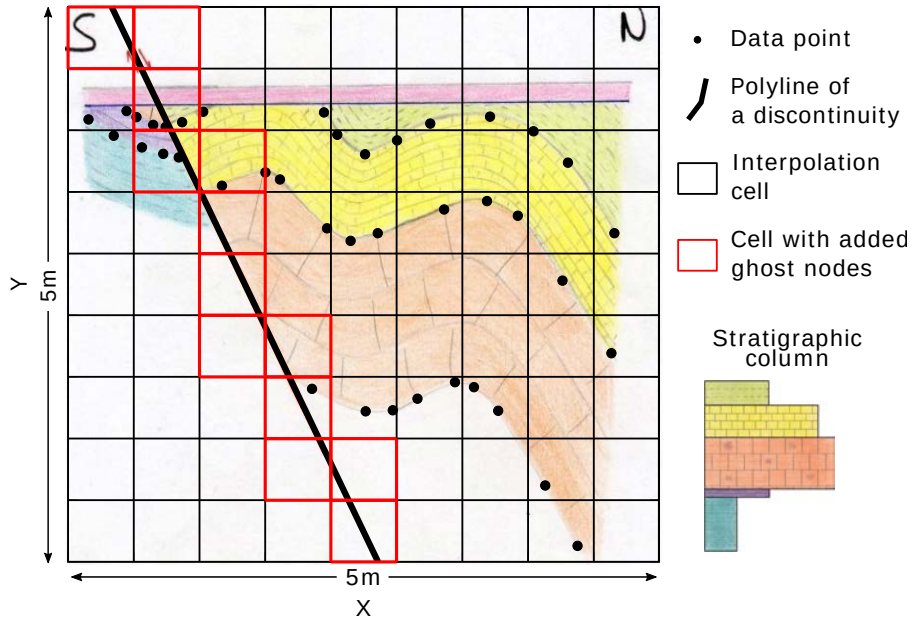


Figure 4.1: *Schematic representation of the proposed modeling method using FEM shape functions and ghost nodes on a synthetic 2D cross section of a folded outcrop with one normal fault (interpreted model from Figure 1.9).*

4.2 The proposed method

4.2.1 The modeling problem using a Cartesian grid with ghost nodes

As presented in Section 2.2.3.3 (p.81), the standard modeling problem is posed as a regression of data points value (DPV) constraints penalized by the bending energy as

$$\min_u \left(J(u)\right) \quad = \min_u \left(\frac{1}{2} \int_\Omega \lambda_\epsilon^2 \left(\partial_{ij}^2 u\right)^2 d\Omega + \frac{1}{2} \sum_{\boldsymbol{p} \in \boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \left(u(\boldsymbol{p}) - \alpha_{\boldsymbol{p}}\right)^2 \right), \tag{4.1}$$

where the iso-values $\alpha_{\boldsymbol{p}}$ are determined with a pre-processing.

The domain Ω is discretized on a Cartesian grid for interpolation. The implicit function u is defined with the FEM shape functions constructed on the grid's elements. These functions are thus associated to the grid's corner points, called nodes in the following. Let $\boldsymbol{N}$ be the set of grid nodes, the implicit function u is taken in the space U as

$$U = \{u(\boldsymbol{x}) = \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}} \Phi_{\overline{\boldsymbol{p}}}^{FEM}(\boldsymbol{x})\ u_{\overline{\boldsymbol{p}}} = \boldsymbol{\Phi}(\boldsymbol{x})^T \cdot \boldsymbol{U} \mid \boldsymbol{x} \in \Omega\}, \tag{4.2}$$

with $\boldsymbol{U}$ the problem's unknowns.

In this section, the details of the method solving this problem in 2D is presented, involving the definition of the FEM shape functions, the discretization of the continuous energy, and the use of ghost nodes to handle the structural discontinuities. The approach is illustrated in Figure 4.1 on a handmade interpreted cross section.

4.2.2 Bilinear quadilateral shape functions

The FEM shape functions $\Phi_{\overline{\boldsymbol{p}}}^{FEM}$ are constructed on the grid cells as isoparametric quadrilateral elements with four nodes (i.e., the corner points of the cell) (Dhatt & Touzot, 1981). For any position $\boldsymbol{p}$ in Ω, the containing cell c is found. The position $\boldsymbol{p}(p_x, p_y)$ and the four cell nodes $\overline{\boldsymbol{p}} \in \boldsymbol{N}(c)$ are substituted in the cell space as the position $\boldsymbol{\xi}(\xi, \eta)$ and the four nodes $\boldsymbol{\xi}_1(-1,-1)$, $\boldsymbol{\xi}_2(1,-1)$,

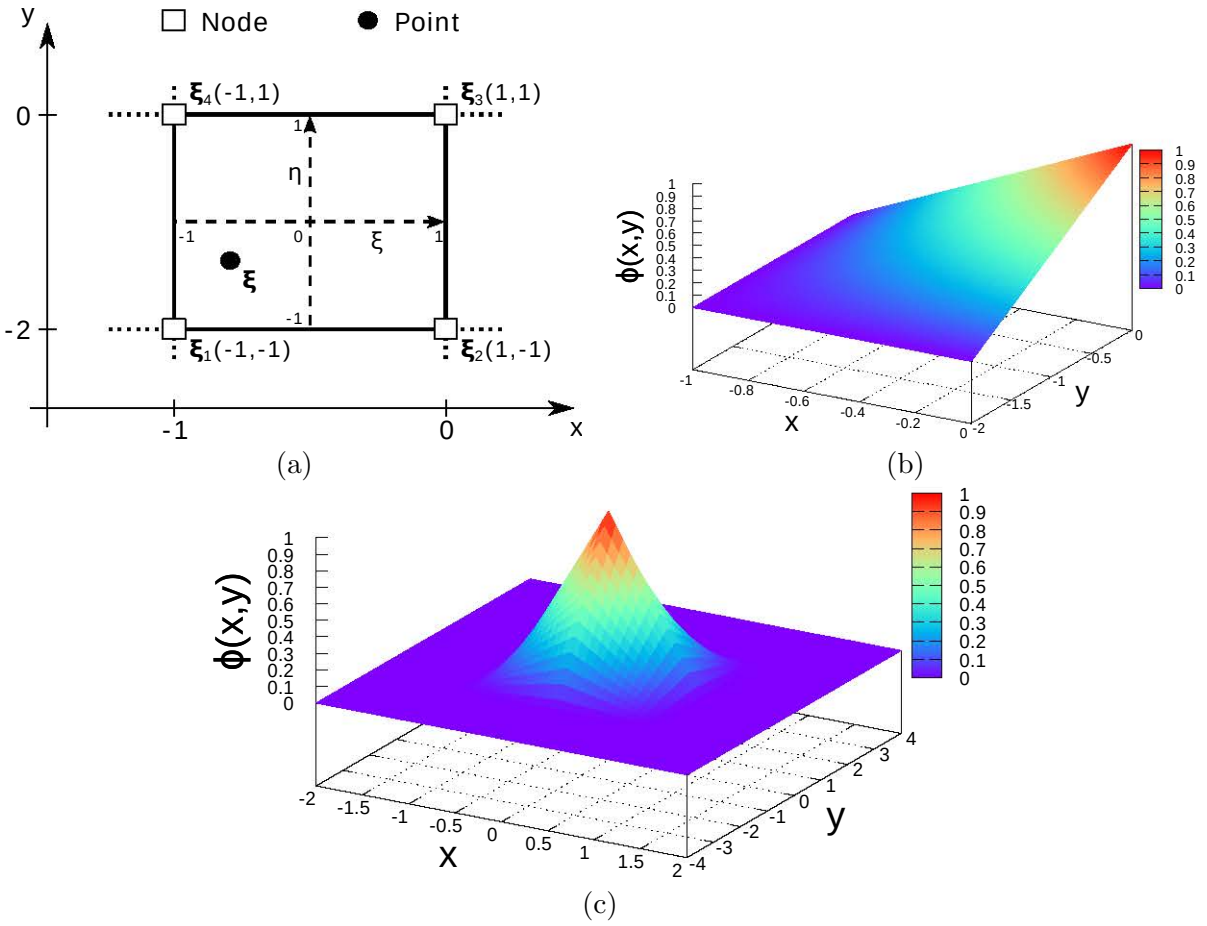


Figure 4.2: *Illustrations and 2D plots of the bilinear quadrilateral shape function on a Cartesian grid. (a) Illustration of the substitution from the Cartesian space to the cell's space. (b) Plot of the interpolation function $\mathcal{N}_3$ in the cell represented in (a). (c) Plot of the FEM shape function of the node $\overline{\boldsymbol{p}}(0,0)$ in a (5×5) grid.*

$\boldsymbol{\xi}_3(1,1)$ and $\boldsymbol{\xi}_4(-1,1)$ as illustrated in Figure 4.2(a). The interpolation functions associated to each node in this space are defined as

$$\begin{aligned} \mathcal{N}_1(\boldsymbol{\xi}) &= \frac{1}{4}(1-\xi)(1-\eta), \quad \mathcal{N}_2(\boldsymbol{\xi}) = \frac{1}{4}(1+\xi)(1-\eta), \\ \mathcal{N}_3(\boldsymbol{\xi}) &= \frac{1}{4}(1+\xi)(1+\eta), \quad \mathcal{N}_4(\boldsymbol{\xi}) = \frac{1}{4}(1-\xi)(1+\eta). \end{aligned} \tag{4.3}$$

The interpolation function $\mathcal{N}_3$ in the example of Figure 4.2(a) is plotted in Figure 4.2(b). The FEM function $\Phi_{\overline{\boldsymbol{p}}}^{FEM}$ associated to the node $\overline{\boldsymbol{p}}$ is thus constructed with the influence of this node in all the cells of the mesh as illustrated in Figure 4.2(c). This function is C^∞ everywhere except at $\overline{\boldsymbol{p}}$ and on the edges of its adjacent cells where it is C^0.

4.2.3 Test functions and domain discretization

Following the Finite Element Method, a Bubnov-Galerkin scheme is adopted. The basis test functions is the same as the basis shape functions (i.e., $\boldsymbol{\Psi} \equiv \boldsymbol{\Phi}$). To approximate the integration, the domain Ω is subdivided into the dual grid of the grid of interpolation. As in the MLS discretization (Section 3.2.4, p.94), all the subdomains $\omega_{\overline{\boldsymbol{p}}}$ have the same volume ν (i.e., the volume of a cell of the grid) but they are centered on the nodes $\overline{\boldsymbol{p}} \in \boldsymbol{N}$. The integration term is approximated as constant in each subdomain $\omega_{\overline{\boldsymbol{p}}}$ and evaluated at the center node $\overline{\boldsymbol{p}}$ (i.e., quadrature with one point, Appendix E, p.169).

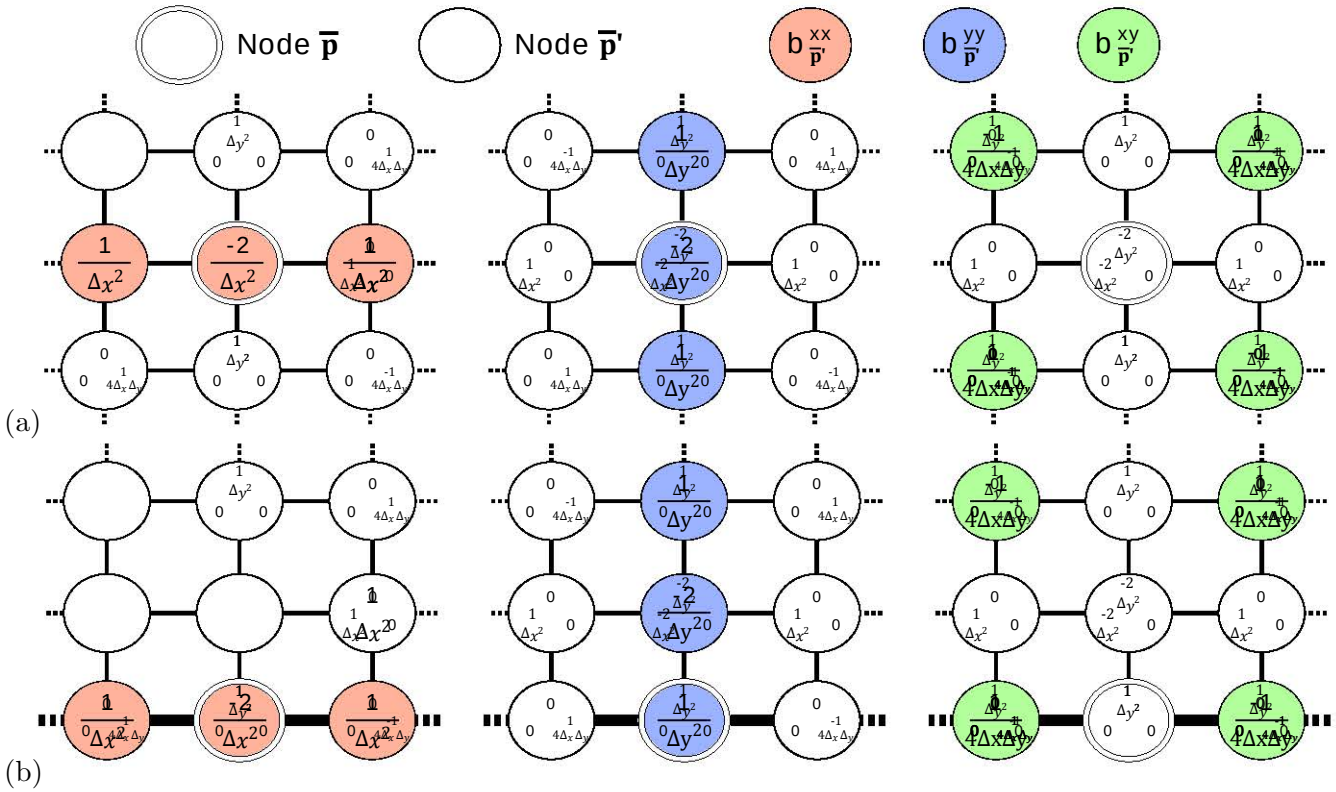


Figure 4.3: *Two examples of finite difference schemes of second derivatives away from any discontinuities. (a) On a grid node away from the grid's borders, and (b) On a grid node on one of the grid's borders.*

As presented in Section 2.2.4.4 (p.83), this approach leads to the normal equations of a least squares system with each constraint written as a row (system (2.60)), which is convenient for the implementation.

4.2.4 Finite difference approximation of the continuous energy

The chosen discretization assumes that the energy term is evaluated at the grid's nodes. However, the bending energy involves second derivatives, and the FEM shape functions are only C^0 at the mesh nodes. We propose to approximate this evaluation with finite differences.

The second derivatives at a node $\overline{\boldsymbol{p}} \in \boldsymbol{N}$ are written as

$$\partial^2_{xx} u(\overline{\boldsymbol{p}}) \approx \frac{u(\overline{\boldsymbol{p}} + \Delta x) - 2u(\overline{\boldsymbol{p}}) + u(\overline{\boldsymbol{p}} - \Delta x)}{\Delta x^2}, \tag{4.4}$$

$$\partial^2_{yy} u(\overline{\boldsymbol{p}}) \approx \frac{u(\overline{\boldsymbol{p}} + \Delta y) - 2u(\overline{\boldsymbol{p}}) + u(\overline{\boldsymbol{p}} - \Delta y)}{\Delta y^2}, \tag{4.5}$$

$$\partial^2_{xy} u(\overline{\boldsymbol{p}}) \approx \frac{u(\overline{\boldsymbol{p}} + \Delta x + \Delta y) - u(\overline{\boldsymbol{p}} + \Delta x - \Delta y) - u(\overline{\boldsymbol{p}} - \Delta x + \Delta y) + u(\overline{\boldsymbol{p}} - \Delta x - \Delta y)}{4\Delta x \Delta y}, \tag{4.6}$$

where Δx and Δy are the grid spacings in the x and y axes respectively. As the functions $\Phi^{FEM}_{\overline{\boldsymbol{p}}}$ have the Kronecker Delta Property (i.e., $\Phi^{FEM}_{\overline{\boldsymbol{p}}} = u_{\overline{\boldsymbol{p}}},\ \forall \overline{\boldsymbol{p}} \in \boldsymbol{N}$), Equations (4.4), (4.5) and (4.6) reduce to

a linear combination of the coefficients in $\boldsymbol{U}$ as

$$\partial^2_{xx} u(\overline{\boldsymbol{p}}) \approx \sum_{\overline{\boldsymbol{p}}' \in \boldsymbol{N}} b^{xx}_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})\ u_{\overline{\boldsymbol{p}}'} = \boldsymbol{b}^{xx}(\overline{\boldsymbol{p}})^T \cdot \boldsymbol{U} \tag{4.7}$$

$$\partial^2_{yy} u(\overline{\boldsymbol{p}}) \approx \sum_{\overline{\boldsymbol{p}}' \in \boldsymbol{N}} b^{yy}_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})\ u_{\overline{\boldsymbol{p}}'} = \boldsymbol{b}^{yy}(\overline{\boldsymbol{p}})^T \cdot \boldsymbol{U} \tag{4.8}$$

$$\partial^2_{xy} u(\overline{\boldsymbol{p}}) \approx \sum_{\overline{\boldsymbol{p}}' \in \boldsymbol{N}} b^{xy}_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})\ u_{\overline{\boldsymbol{p}}'} = \boldsymbol{b}^{xy}(\overline{\boldsymbol{p}})^T \cdot \boldsymbol{U} \tag{4.9}$$

with $b^{xx}_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})$, $b^{yy}_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})$ and $b^{xy}_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})$ the combination factors associated to the node $\overline{\boldsymbol{p}}' \in \boldsymbol{N}$ and its corresponding coefficient $u_{\overline{\boldsymbol{p}}'}$ when writing the second derivatives of u at $\overline{\boldsymbol{p}}$. The general scheme of finite differences is illustrated in Figure 4.3(a) where the nodes with not null combination factors are represented.

When the second derivatives are evaluated on a node on the border of the grid, the finite differences are approximated by the evaluation on the neighboring nodes when necessary, as illustrated in Figure 4.3(b).

It is unclear to the author whether using the finite difference method gives an approximation of the bending energy or a different problem. In the MLS discretization (Section 3.5.1, p.114), the proposed finite differences assume that the MLS functions have the KΔP. The second derivatives of the bending energy are therefore approximated without involving the high degree of derivability of the shape functions. Here, the same problem is posed: the finite differences smooth the coefficients themselves without involving the shape functions.

4.2.5 Handling the discontinuities with ghost nodes

4.2.5.1 Definition

As done in the presented prototype using the Moving Least Squares functions, we represent faults and unconformities with a discontinuous jump in the implicit function u (Section 3.2.5, p.95). To introduce the discontinuity in u, we propose to use the ghost fluid method (Section 2.1.4.2, p.72) but restricted to the grid cells crossed by the discontinuities as illustrated in Figure 4.4. Nodes are added on either side of the discontinuities on top of already existing grid corners.

Two different sets of nodes are differentiated: the set of grid nodes $\boldsymbol{N}$ and the set of ghost nodes $\boldsymbol{N}_g$ (Figure 4.4). The implicit function is augmented to

$$\begin{aligned} u(\boldsymbol{x}) &= \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}} \kappa_{\overline{\boldsymbol{p}}}(\boldsymbol{x}) \Phi_{\overline{\boldsymbol{p}}}(\boldsymbol{x})\ u_{\overline{\boldsymbol{p}}} \quad + \quad \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}_g} \kappa_{\overline{\boldsymbol{p}}}(\boldsymbol{x}) \Phi_{\overline{\boldsymbol{p}}}(\boldsymbol{x})\ u_{\overline{\boldsymbol{p}}} \quad , \\ &= \boldsymbol{\Phi}_\kappa(\boldsymbol{x})^T \cdot \mathcal{U} \quad , \end{aligned} \tag{4.10}$$

with $\mathcal{U}^T = [\boldsymbol{U}^T, \boldsymbol{U}_g^T]$, $\boldsymbol{\Phi}_\kappa = [\kappa_{\overline{\boldsymbol{p}}} \Phi_{\overline{\boldsymbol{p}}} \ \mid \ \forall \overline{\boldsymbol{p}} \in (\boldsymbol{N} \cup \boldsymbol{N}_g)]$ and $\kappa_{\overline{\boldsymbol{p}}}$ the jump function defined as

$$\kappa_{\overline{\boldsymbol{p}}}(\boldsymbol{x}) = \begin{cases} 1, & \boldsymbol{x} \in S_{\overline{\boldsymbol{p}}} \\ 0, & \boldsymbol{x} \notin S_{\overline{\boldsymbol{p}}} \end{cases} , \tag{4.11}$$

with $S_{\overline{\boldsymbol{p}}}$ the support for the jump function defined differently on a grid or a ghost node (Figure 4.4) as

$$\forall \overline{\boldsymbol{p}} \in \boldsymbol{N}, \quad \boldsymbol{x} \in S_{\overline{\boldsymbol{p}}} \ \ if \ \ (\boldsymbol{x} \in \Omega)\,, \ \left(\forall F,\ \overline{\boldsymbol{x}\overline{\boldsymbol{p}}} \cap F = \emptyset\right), \tag{4.12}$$

$$\forall \overline{\boldsymbol{p}} \in \boldsymbol{N}_g, \quad \boldsymbol{x} \in S_{\overline{\boldsymbol{p}}} \ \ if \ \ (\boldsymbol{x} \in \Omega)\,, \ \left(\exists F,\ \overline{\boldsymbol{x}\overline{\boldsymbol{p}}} \cap F \neq \emptyset\right), \tag{4.13}$$

$$\tag{4.14}$$

with F a fault and $\overline{\boldsymbol{x}\overline{\boldsymbol{p}}}$ the segment between the position $\boldsymbol{x}$ and the node $\overline{\boldsymbol{p}}$.

This formulation is strictly equivalent to the XFEM basis functions as described in Section 2.1.4.2 (p.72), but we consider it more intuitive on an implementation aspect. When evaluating u at a position $\boldsymbol{x}$, if $\boldsymbol{x}$ belongs to a cell crossed by one or several discontinuities, the cell corners used for interpolation are found following Figure 4.4. If the segment between $\boldsymbol{x}$ and a corner $\overline{\boldsymbol{p}}$ intersects any discontinuity F (i.e., $\exists F,\ \overline{\boldsymbol{x}\overline{\boldsymbol{p}}} \cap F \neq \emptyset$), then the corresponding ghost node $\overline{\boldsymbol{p}} \in \boldsymbol{N}_g$ is used instead of the grid node $\overline{\boldsymbol{p}} \in \boldsymbol{N}$ for the evaluation. With this technique, the implicit function is continuously evaluated on either side of the discontinuities, regardless of their geometry.

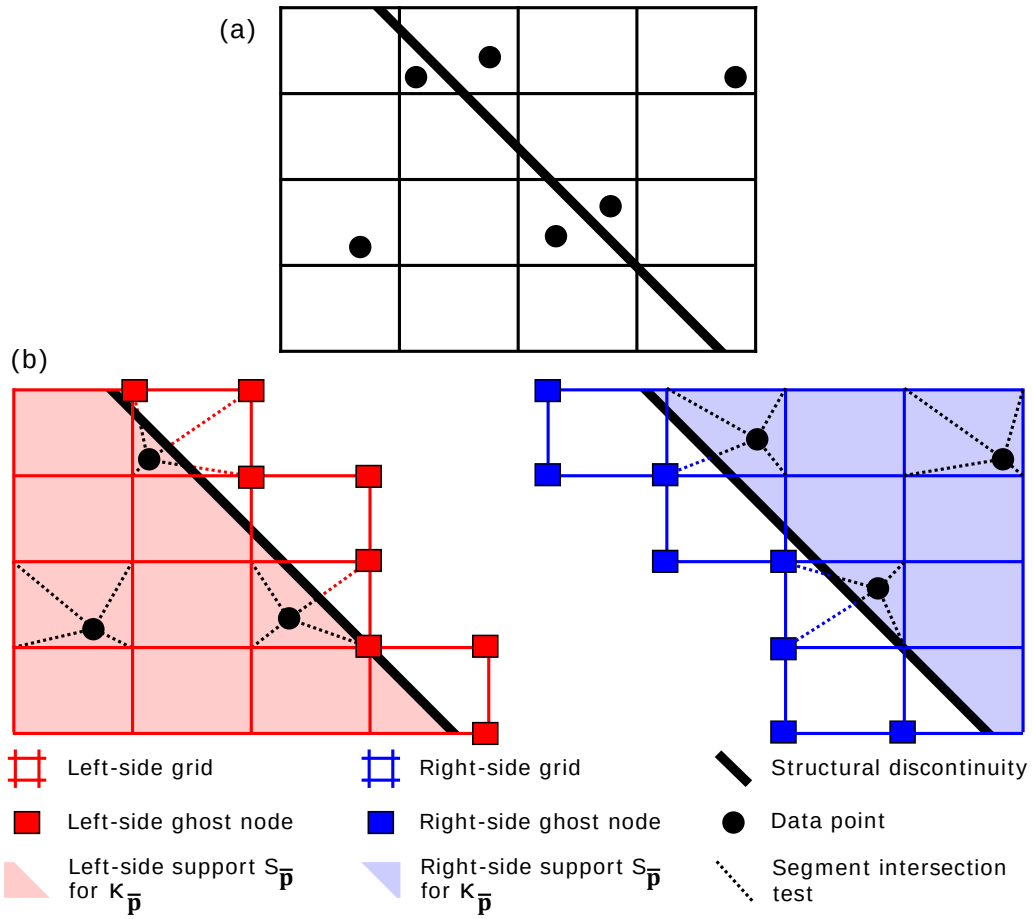


Figure 4.4: *Illustration of the used ghost-cell technique by separating the evaluation on the left and on the right side of a discontinuity. (a) The initial problem with the interpolation grid, data points, and an infinite fault. (b) The evaluation scheme with ghost and grid nodes, defining the data point neighbors with segment intersection tests.*

4.2.5.2 Structural discontinuities with complex geometries

Fault tips

Faults represent slip surfaces that can laterally vanish in space, where the displacement on each side of the fault becomes null. The implicit function is supposed to be discontinuous on the fault and to become continuous at the fault tips. The chosen strategy to achieve the transition from an interpolation with ghost nodes to an interpolation without ghost nodes is illustrated at a fault tip in 2D in Figure 4.5. Ghost nodes are added on each corner of the containing cell and segment-discontinuity intersection tests are performed to define the grid and ghost supports $S_{\overline{\boldsymbol{p}}}$ for the evaluation of u at a position $\boldsymbol{x}$.

Intersections of discontinuities

Several discontinuities (e.g., faults, unconformities) can branch one onto another in space. The presented method creates as many ghost nodes $\overline{\boldsymbol{p}} \in \boldsymbol{N}_g$ as needed to interpolate on each area delimited by the discontinuities and their geometry. The corresponding ghost supports $S_{\overline{\boldsymbol{p}}}$ are thus dependent on the areas created by the fault intersections and have a more complex definition than in Equation (4.13). In Figure 4.6(a), three areas are delimited by the fault intersection, so two ghost nodes are associated to each grid node. In Figure 4.6(b), four areas are delimited by the fault intersection, so three ghost nodes are associated to each grid node. This principle can be adapted to any number of areas.

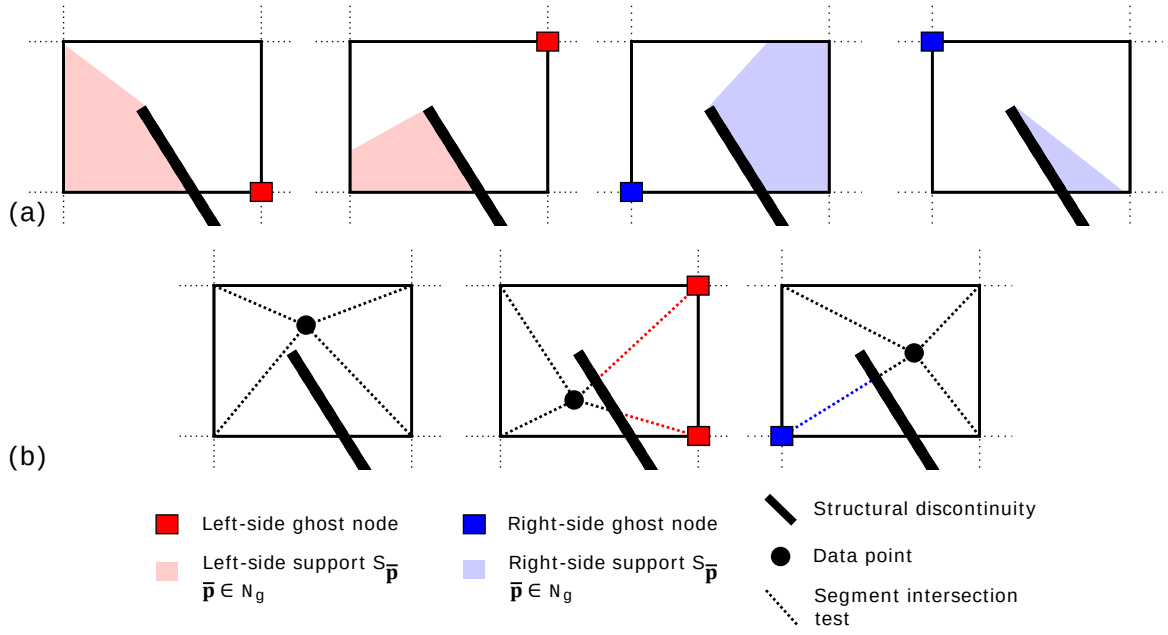


Figure 4.5: *Illustration of the ghost-cell technique applied on a fault tip. A ghost node is used only if the segment between that node and a given data point intersects the discontinuity. (a) The jump supports $S_{\overline{p}}$ associated to each ghost node, and (b) Three examples of data point evaluation schemes using ghost and grid nodes.*

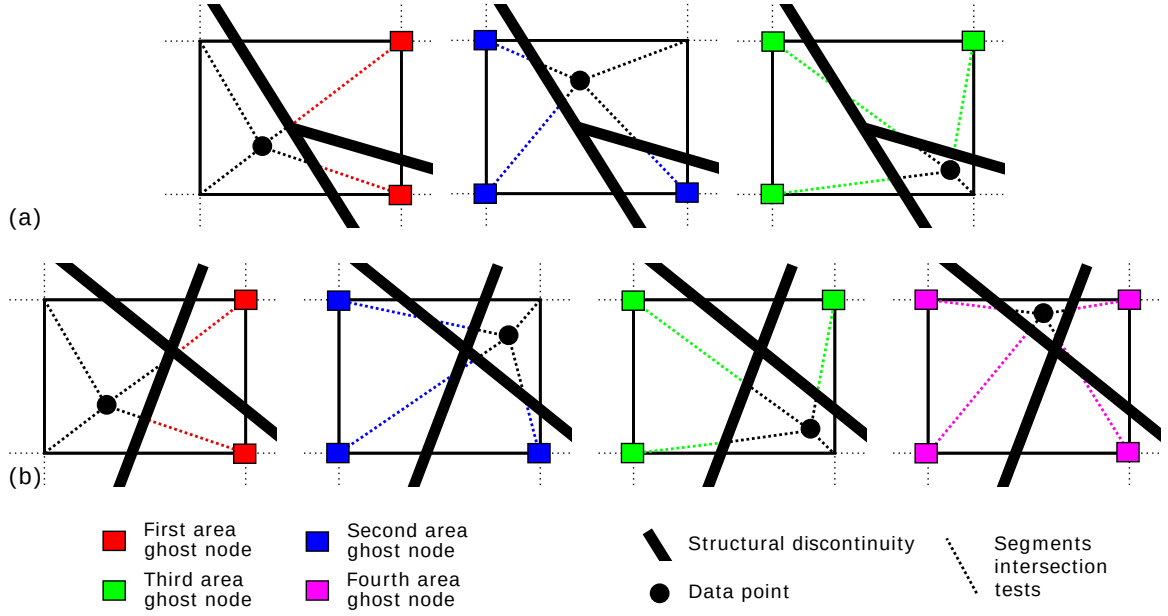


Figure 4.6: *Illustration of the used ghost-cell technique applied on several intersecting discontinuities. Ghost nodes are added for each area sealed by the discontinuities. (a) Three examples of data point evaluation schemes with three fault zones, and (b) Four examples of data point evaluation schemes with four fault zones.*

4.2.5.3 Finite difference method with ghost nodes

The conventional finite difference schemes presented in Figure 4.3 only apply to nodes far from the discontinuities. When close to one or several discontinuities, the finite differences are written with both grid and ghost nodes. As an example, Figure 4.7(a) illustrates the not null combination factors when approximating the second derivatives of u at a grid node $\overline{\boldsymbol{p}} \in \boldsymbol{N}$ close to a discontinuity. If close

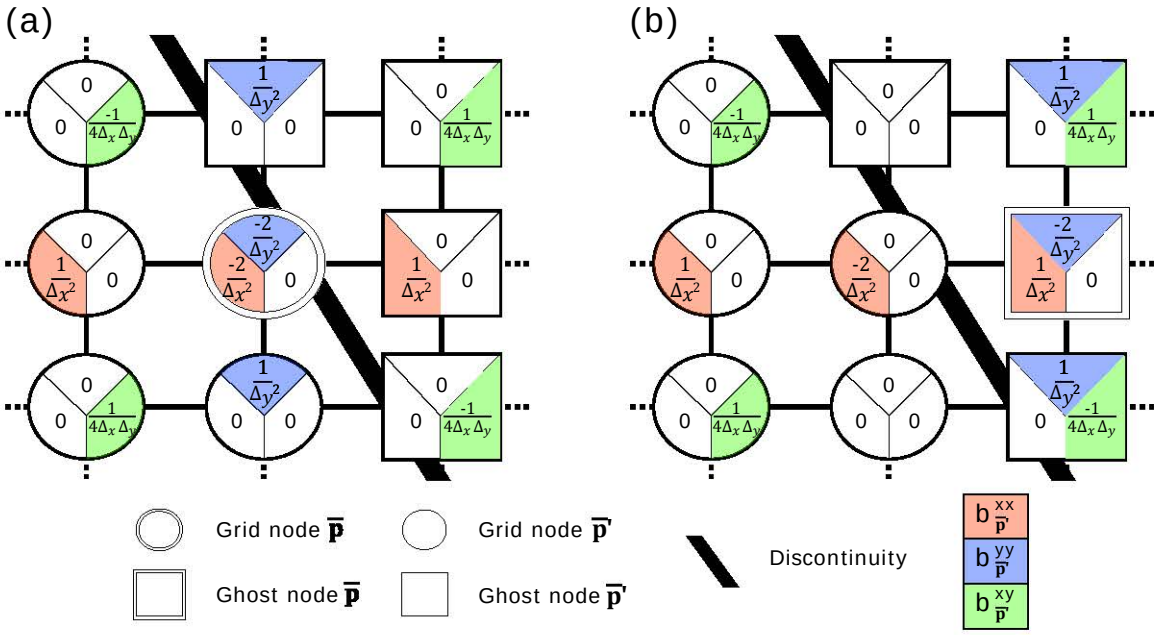


Figure 4.7: *Two examples of finite difference schemes of second derivatives close to a discontinuity. (a) On a grid node, and (b) On a ghost node.*

to discontinuities with complex geometries, the involved neighboring ghost nodes are chosen with the same strategy as for the evaluation (Section 4.2.5.2). This is also to be adapted in the case of a grid node close to both discontinuities and grid borders.

The approximation of the second derivatives of u evaluated at a grid node is thus augmented to

$$\partial^2_{xx}u(\overline{\boldsymbol{p}}) \approx \sum_{\overline{\boldsymbol{p}}'\in\boldsymbol{N}} \kappa_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})\; b^{xx}_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})\; u_{\overline{\boldsymbol{p}}'} + \sum_{\overline{\boldsymbol{p}}'\in\boldsymbol{N}_g} \kappa_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})\; b^{xx}_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})\; u_{\overline{\boldsymbol{p}}'} = \boldsymbol{b}^{xx}_{\kappa}(\overline{\boldsymbol{p}})^T \cdot \mathcal{U} \tag{4.15}$$

$$\partial^2_{yy}u(\overline{\boldsymbol{p}}) \approx \sum_{\overline{\boldsymbol{p}}'\in\boldsymbol{N}} \kappa_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})\; b^{yy}_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})\; u_{\overline{\boldsymbol{p}}'} + \sum_{\overline{\boldsymbol{p}}'\in\boldsymbol{N}_g} \kappa_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})\; b^{yy}_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})\; u_{\overline{\boldsymbol{p}}'} = \boldsymbol{b}^{yy}_{\kappa}(\overline{\boldsymbol{p}})^T \cdot \mathcal{U} \tag{4.16}$$

$$\partial^2_{xy}u(\overline{\boldsymbol{p}}) \approx \sum_{\overline{\boldsymbol{p}}'\in\boldsymbol{N}} \kappa_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})\; b^{xy}_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})\; u_{\overline{\boldsymbol{p}}'} + \sum_{\overline{\boldsymbol{p}}'\in\boldsymbol{N}_g} \kappa_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})\; b^{xy}_{\overline{\boldsymbol{p}}'}(\overline{\boldsymbol{p}})\; u_{\overline{\boldsymbol{p}}'} = \boldsymbol{b}^{xy}_{\kappa}(\overline{\boldsymbol{p}})^T \cdot \mathcal{U} \tag{4.17}$$

with $\boldsymbol{b}^{ij}_{\kappa} = [\kappa_{\overline{\boldsymbol{p}}} b^{ij}_{\overline{\boldsymbol{p}}} \;\mid\; \forall \overline{\boldsymbol{p}} \in (\boldsymbol{N} \cup \boldsymbol{N}_g)]$ the combination factors defined on the grid and ghost nodes respectively.

In the modeling problem, the energy term is defined on the entire domain Ω. According to the definition of u (Equation (4.10)), all grid and ghost nodes are involved in the energy minimization. Therefore, we propose to discretize the continuous term by evaluating the second derivatives not only on the grid nodes, but also on the ghost nodes. Figure 4.7(b) illustrates the not null combination factors when approximating the second derivatives of u at a ghost node $\overline{\boldsymbol{p}} \in \boldsymbol{N}_g$. Here again, the chosen neighbors for the approximation must be adapted in the case of discontinuities with complex geometries and the proximity to grid borders.

4.2.6 Example of a system of equations

In 2D, the least squares system corresponding to the minimization problem of Equation (2.55) (p.81) with the presented discretization can be written as

$$\begin{cases} \lambda_\epsilon \sqrt{\nu}\; \boldsymbol{b}^{xx}_{\kappa}(\overline{\boldsymbol{p}})^T \cdot \mathcal{U} = 0, & \forall \overline{\boldsymbol{p}} \in (\boldsymbol{N} \cup \boldsymbol{N}_g) \\ \lambda_\epsilon \sqrt{\nu}\; \boldsymbol{b}^{yy}_{\kappa}(\overline{\boldsymbol{p}})^T \cdot \mathcal{U} = 0, & \forall \overline{\boldsymbol{p}} \in (\boldsymbol{N} \cup \boldsymbol{N}_g) \\ \lambda_\epsilon \sqrt{\nu}\; \boldsymbol{b}^{xy}_{\kappa}(\overline{\boldsymbol{p}})^T \cdot \mathcal{U} = 0, & \forall \overline{\boldsymbol{p}} \in (\boldsymbol{N} \cup \boldsymbol{N}_g) \\ \lambda_{\boldsymbol{p}}\; \boldsymbol{\Phi}_{\kappa}(\boldsymbol{p})^T \cdot \mathcal{U} = \lambda_{\boldsymbol{p}} \alpha_{\boldsymbol{p}}, & \forall \boldsymbol{p} \in \boldsymbol{D} \end{cases} . \tag{4.18}$$

We solve this system with an LU decomposition solver for sparse matrices.

4.2.7 Extraction of the iso-surfaces

After solving system (4.18), the obtained coefficients $\mathcal{U}$ are used to evaluate the implicit function u in space. As for the MLS discretization (Section 3.2.7, p.95), the domain Ω is discretized on a grid of visualization points gathered in the set $\boldsymbol{N}_v$. The implicit function u is thus evaluated on each visualization point (with Equation (4.10)), and the iso-surfaces are extracted linearly on each grid element with marching triangles in 2D. As the same extraction technique is employed, this method holds the same limitations concerning the generation of sealed models. The resolution of visualization is chosen fine enough to have the stair-steps under the 2D fault lines.

4.3 Comparison with the moving least squares (MLS) discretization

The proposed method is tested on a few 2D examples and the results are compared with the MLS technique. Some examples are visually compared using a (300×300) grid to have high quality images, and some others are compared with the model and data distances D_{model} and D_{data} (Section 3.4.2, p.103) evaluated on a (100×100) grid.

The computing times of the MLS discretization are discussed in Section 3.3 (p.96). For the visualization results with the ghost discretization, it takes less than a second to create the implicit function (between $0.1s$ and $0.3s$) and around three seconds to print the function on the (300×300) grid. Concerning the model and data distances, the computing times involving both the creation of the implicit function and the distance evaluations on the (100×100) grid are given in the corresponding graphs. All the models are run on a laptop with Intel Core i7-4940 3GHz with 32 Gb of RAM, and running Windows 7 enterprise 64 bits.

4.3.1 Comparison tests in 2D

4.3.1.1 Visual comparisons of the interpolation

The method is tested on the cross section of Figure 4.1 with three different resolutions of the interpolation grid. These are respectively represented side by side with results obtained with the MLS discretization using the same nodal resolutions in Figure 4.8. For a coarse resolution of the grid (Figure 4.8(a)), the iso-surfaces obtained with the bi-linear interpolation contain angles at the edges of the grid cells. For a fine resolution (Figure 4.8(e)), the angles are not anymore observable. Such artifacts are not observed in the MLS interpolation which globally produces smoother results.

Visually, except for this difference of continuity in the results, both discretizations seem to give the same function u with approximately the same gradient everywhere in space. They also seem to converge towards the same solution when the resolution of the grid increases.

The ghost nodes efficiently handle the discontinuities, even with small angle intersections, as shown in Figure 4.9(a). It even has the advantage to ensure the interpolation in narrow areas as compared to the MLS discretization using the visibility criterion at the bottom of Figure 4.9(b). These advantages still have limits which are discussed in Section 4.3.2.2.

4.3.1.2 Distance comparisons of the interpolation

Visual comparisons are not enough to appreciate the difference of interpolation between the two proposed methods. We propose to perform a few comparison tests on the model used for sensibility tests on the MLS prototype (Figure 3.12, p.104). In Figure 4.10, the model and data distances D_{model} and D_{data} (Section 3.4.2) are compared between ghost and MLS discretizations with a varying number of interpolation nodes $\#\boldsymbol{N}$ (this number does not take into account the number of ghost nodes $\#\boldsymbol{N}_g$). The reference model used to evaluate D_{model} is the same as in Section 3.4.2, hence created with the MLS discretization.

In Figure 4.10(a), the models are computed using the reference data points given in Figure 3.12. In Figure 4.10(b), they are computed using the noisy data points presented in Figure 3.14. In these two figures, distance values of the same order are evaluated on ghost and MLS results created with the same number of nodes. When using the reference data, the results from both techniques converge close to the reference model when the number of nodes increases. This is expected as these two techniques solve the same modeling problem given by system (2.55) (p.81). A result with a null D_{model} is obtained

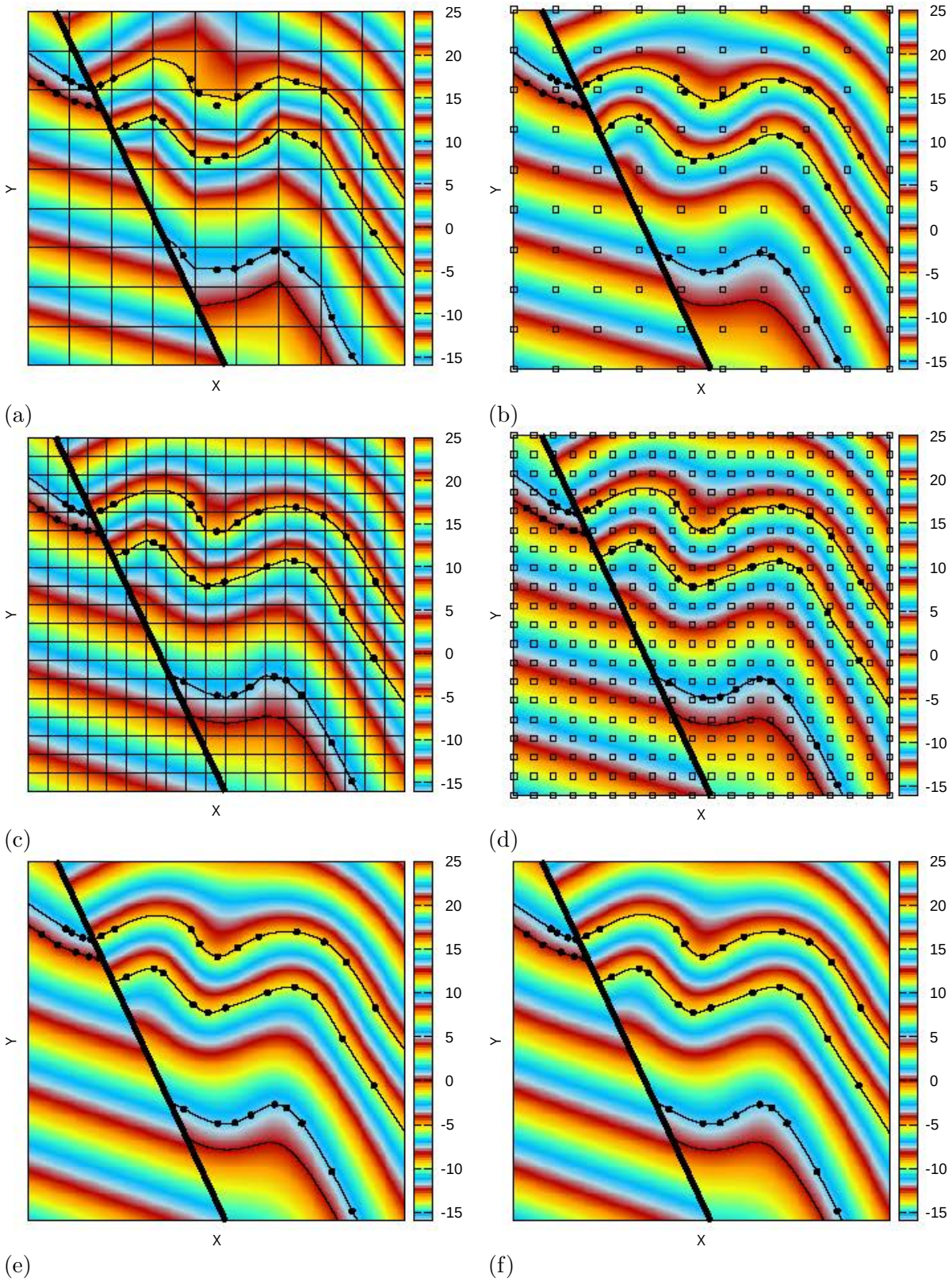


Figure 4.8: *Visual comparison models computed with the proposed discretizations on the numerical data presented in Figure 4.1 and with three different nodal resolutions. (a), (c) and (e) The FEM interpolation with ghost nodes, and (b), (d) and (f) The MLS interpolation with the visibility criterion. The nodal resolution is of* (10×10) *in (a) and (b),* (20×20) *in (c) and (d), and* (100×100) *in (e) and (f) where the grid and nodes are not represented for visibility reasons.*
Iso-values from bottom to top: $[0, 1, 7, 11.5]$.
$\lambda_\epsilon = 1$, $\lambda_{\boldsymbol{p}} = 1$.

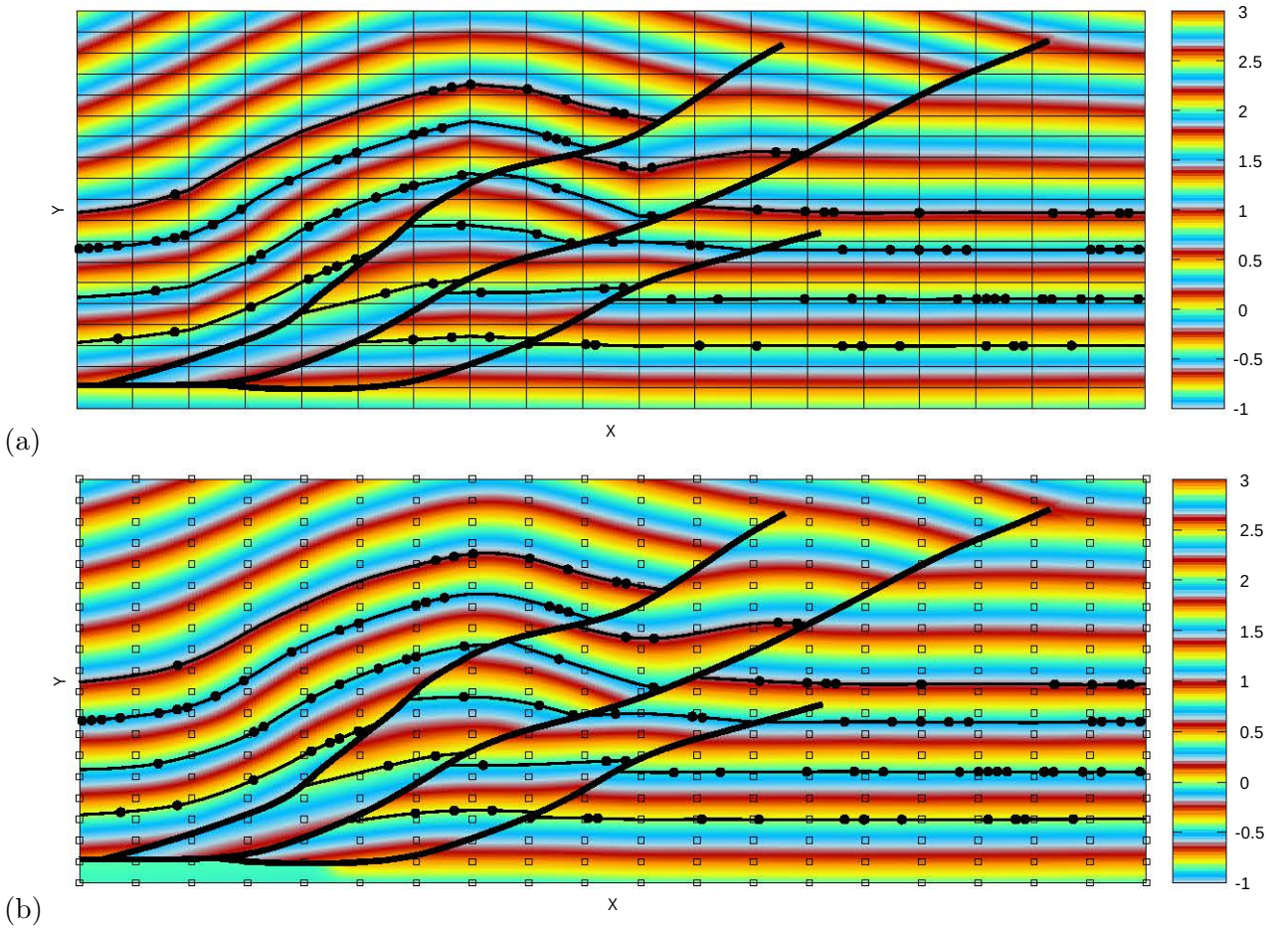


Figure 4.9: *Visual comparison models computed with the proposed discretizations on the numerical data used in Figure 3.6. (a) The FEM interpolation with ghost nodes, and (b) The MLS interpolation with the visibility criterion. The nodal resolution is fixed to* (20×20). $\lambda_\epsilon = 10$, $\lambda_{\boldsymbol{p}} = 1$.

with the MLS discretization (i.e., with (50×50) : $\#\boldsymbol{N} = 2500$ nodes) because the reference model was created with the same parameters.

Although close to each other, these evaluated distances are not exactly identical. Concerning the model distances, D^{MLS}_{model} is almost always smaller than D^{ghost}_{model}, even with a fine nodal resolution. This is due to the used reference model which was created with the MLS discretization: when using a reference model created with the ghost discretization, D^{ghost}_{model} tends to be slightly smaller than D^{MLS}_{model}. These remain two different approximations of the proposed modeling problem, which is why these two modeling techniques do not converge to the exact same solution. Concerning the data distances, D^{MLS}_{model} is also smaller than D^{ghost}_{model} in most results, even with a fine nodal resolution. This may be due to a slightly greater impact of the data weight $\lambda_{\boldsymbol{p}}$ on the least squares system in the MLS approximation than in the ghost approximation.

4.3.2 Benefits and limits of the Cartesian grid with ghost nodes

4.3.2.1 Computational efficiency

The noticeable difference between the two methods is the time taken to compute a model. In Figures 4.10(a) and (b), both methods show a linear dependency on the number of nodes, but the ghost discretization is much faster. This is due to the fact that the general system (4.18) is much sparser. In particular, the finite difference schemes only involve three to four coefficients per equation in comparison to nine coefficients (i.e., with $\rho = 1.99$) in the second derivative evaluations with the MLS

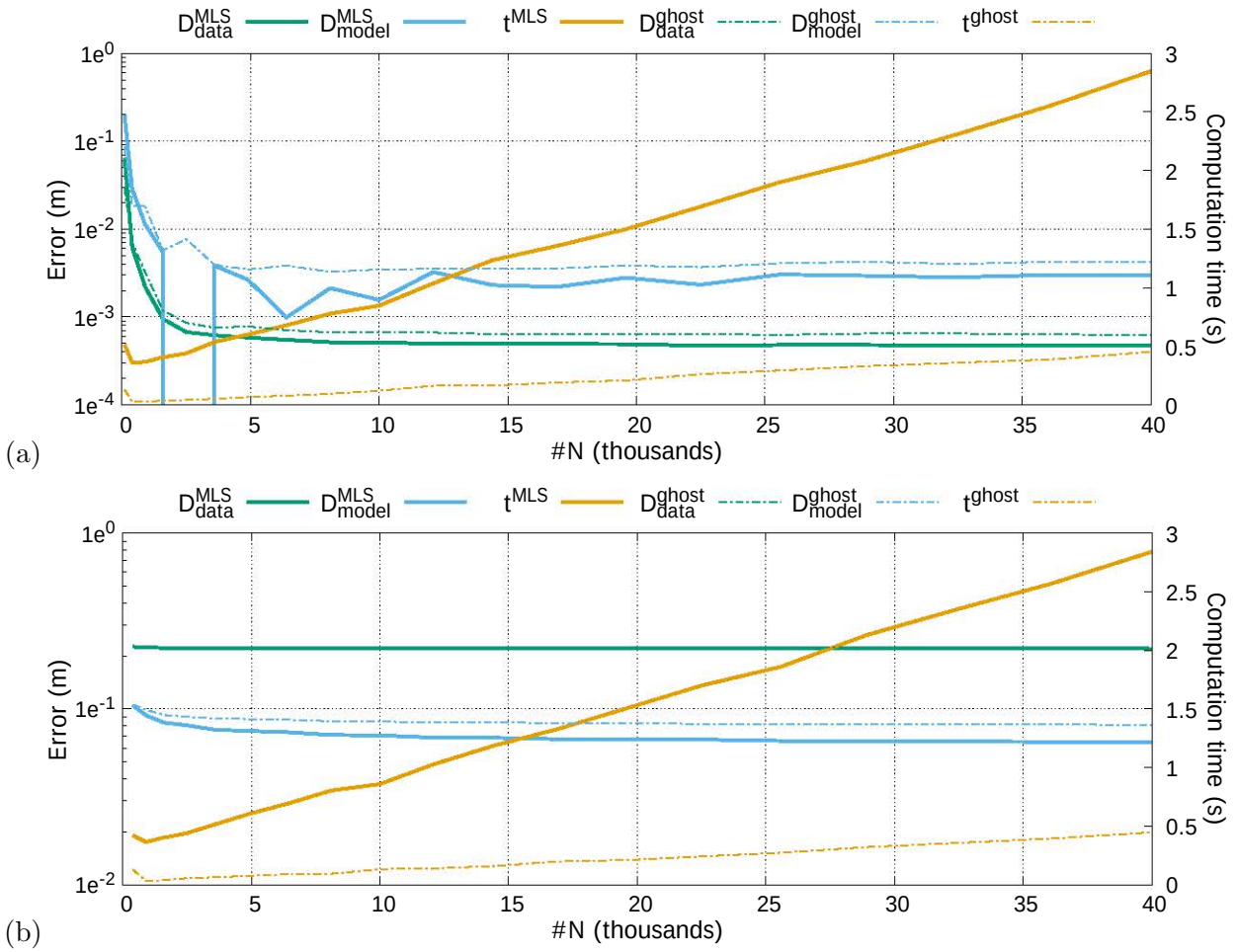


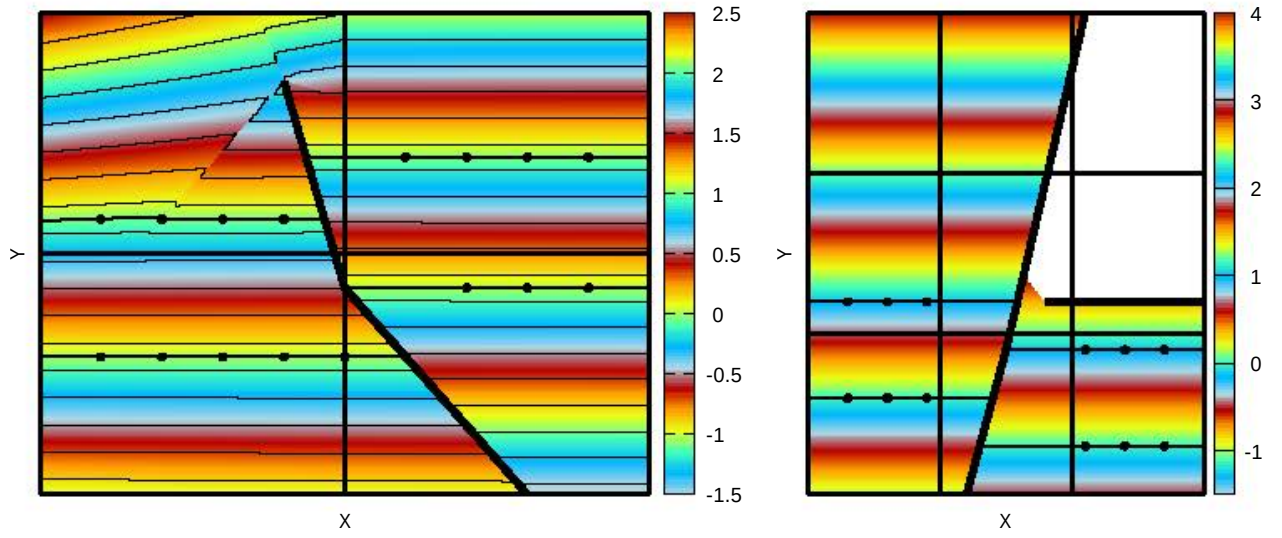
Figure 4.10: *Comparison tests of efficiency and convergence between ghost and MLS discretizations. (a) Performed on the model of reference* $\lambda_\epsilon = 1$, $\lambda_{\boldsymbol{p}} = 1$. *(b) Performed on the model of reference with noise in the data*$(1m)$. $\lambda_\epsilon = 30$, $\lambda_{\boldsymbol{p}} = 1$.

functions. Even if the finite differences were used with the MLS technique, the evaluation of data points involves four coefficients in the ghost interpolation (i.e., the containing cell corners) whereas it involves nine coefficients on average with the MLS interpolation. Although the ghost technique adds unknowns and smoothing equations to the problem, those modifications are local, hence have a negligible impact on the dimensions of the system. Also, the ghost interpolation involves less coefficients in the evaluation on each visualization point as compared to the MLS interpolation (i.e., four instead of nine), so printing the implicit function on the visualization grid is also performed faster with the ghost technique.

4.3.2.2 Limits of the ghost nodes

The strategy employed to decide whether a ghost node is used or not is close to the visibility criterion. It therefore has the same limits as discussed in Section 3.3.3 (p.101): it lacks stability at fault tips and renders two fault blocks independent one from another.

The discontinuous jumps in the implicit function at the fault tip of Figure 4.11(a) are observed with a visualization grid much finer (i.e., 300×300) than the interpolation grid (i.e., 3×3). These artifacts are equivalent to the ones obtained with the visibility criterion in Figure 3.11(a). The use of ghost nodes at fault tips could be adapted to mimic other meshless techniques such as the diffraction and the transparency criteria (as used in Section 3.3.3). The crack tip function proposed in (Moës *et al.*, 1999) could also be tested. In Figure 4.11(b), two faults are close in the same cell but do not intersect. The connectivity between the two regions on the right side is not captured, and as the two fault blocks are independent, the upper right area is left isolated without data, and hence has no



(a) (b)

Figure 4.11: *Limits of the ghost nodes. (a) Unexpected stratigraphic jumps at a fault tip computed with a coarse resolution (i.e., using a* (3×3) *interpolation grid), (b) Isolated section when dealing with non-intersecting faults with a coarse resolution (i.e., using a* (4×4) *interpolation grid).*
$\lambda_\epsilon = 1$, $\lambda_{\boldsymbol{p}} = 1$.

solution. This is due to the way the ghost nodes are handled, using segment-discontinuity intersection tests. This shows that a resolution adapted to the scale of the geological structures being studied is expected with the proposed method. Adapting the ghost nodes technique to mimic the transparency criterion could also avoid such situations.

4.3.2.3 Lack of flexibility at a given resolution

For a coarse nodal resolution, the MLS interpolation creates models visually more continuous than the ghost interpolation (Section 4.3.1.1). This is related to the number of neighbors used for the evaluation of the implicit function: using more neighbors in the MLS functions creates a slower but highly continuous interpolation. It would therefore be interesting to compare the two techniques with the notion of efficiency as defined in Liu & Gu (2005) (i.e., computational cost *vs.* accuracy, Section 2.1.6, p.77). In the applications of Figures 4.10(a) and (b), the higher continuity given by the MLS interpolation is not enough to produce results closer to the data points and the reference model for coarse resolutions. This may be caused by the fault network which is poorly handled by the visibility criterion with a small number of nodes. To perform the study on efficiency, the ghost node technique should also be implemented for the MLS discretization.

The interpolation on a Cartesian grid can also be restrictive. Increasing the continuity of the interpolation in a localized area requires to refine the entire domain. This is not the case in the MLS discretization as nodes can be randomly added in local areas while ensuring a stable interpolation.

4.4 Perspectives and optimizations

4.4.1 Implementation in 3D

The ghost nodes are not yet implemented in 3D. The main reason for this is the complexity of the chosen algorithm to handle their creation and the related equations for arbitrary fault network connectivity and geometry. The current workflow does not sufficiently use the cells connectivity, focusing on each node separately, and creating complex relations between ghost and grid nodes in specific cases (e.g., parallel faults or other non-intersecting faults within a same cell).

A Cartesian grid based structural modeling using finite differences and FEM shape functions was

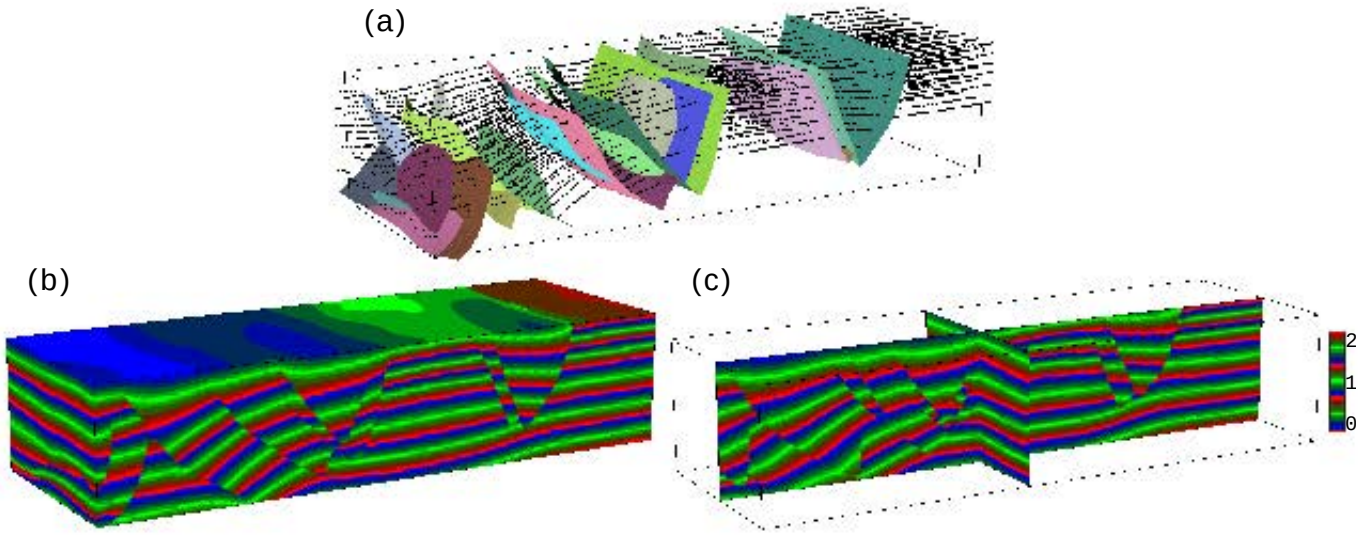


Figure 4.12: *Model computed with the finite differences smooth interpolation (*Irakarama et al.*, 2018a). (a) The data points and faults as triangulated surfaces, and (b) and (c) The solid and section views of the obtained implicit function.*
(Data set provided by IFPEN)

implemented in a 3D prototype in the collaboration work (Section 4.1). A result is given in Figure 4.12 on a 3D model including several intersecting faults. The numerical data count $15,302$ data points and 27 triangulated surfaces. Although a high resolution of the grid is necessary to handle data points close to the discontinuities, the regular structure of the interpolation allowed a parallelization of the code, yielding fast results. The implicit function was computed on a grid of 4 millions of corner nodes in 35 seconds. The visualization is performed directly on the computation grid and the iso-values were not extracted, hence avoiding the evaluation step for visualization. The model of Figure 4.12 was computed on a workstation with 3.5 GHz CPU and Quadro M4000 GPU.

4.4.2 Tree structures

Replacing the Cartesian grid by tree structures could avoid a homogeneous fine resolution of the interpolation grid everywhere in the domain Ω. Finer resolutions could be used to capture local details, such as high curvatures and complex fault geometries, and coarser resolutions could limit the number of unknowns and equations written in areas with little complexity.

Several tree structures can be considered. The best-known are the kd-tree and the quadtree (*resp.* octree) in 2D (*resp.* 3D). There are two difficulties with such supports: (i) they do not form a mesh on which the conventional FEM shape functions can be created (i.e., all the FEM shape functions would not be C^0 everywhere in Ω), and (ii) they require further techniques and approximations to compute the second derivatives (i.e., conventional finite differences cannot be written properly).

These issues have already been studied in the literature, such as constructing a continuous interpolation on quadtree meshes with the Natural Element Method (Tabarraei & Sukumar (2007), Section 2.1.3.2, p.65), or as using substitution schemes to write the finite difference operators on quadtree and octree meshes (Chen *et al.*, 2007). There is also the possibility to study other tree structures such as the binary triangle tree (or triangle bintree, Duchaineau *et al.* (1997)) on which conventional FEM shape functions can be constructed. In addition, such meshes have already been used to handle sets of scattered data with an adapted approximation of the bending energy as a smoothing penalization (Hjelle & Dæhlen, 2005).

Résumé du Chapitre 5

Dans ce cinquième chapitre, nous décrivons plusieurs possibilités d'amélioration du problème continu proposé pour la modélisation structurale.

Pour cela, nous proposons trois modifications du terme de régularisation : (i) remplacer l'énergie de courbure par d'autres énergies physiques, (ii) modifier localement les paramètres de l'énergie de courbure, et (iii) enrichir cette énergie avec d'autres fonctionnelles continues. Chaque opportunité est illustrée en discrétisant les équations proposées avec les moindres carrés glissants.

Certaines de ces modifications sont proposées pour traiter de problèmes communs aux techniques de modélisation structurale par méthodes implicite. En particulier, nous évitons la création d'incohérences stratigraphiques dans des problèmes de variation d'épaisseur en variant, dans l'espace, la pénalisation de l'énergie de courbure dans le système de minimisation. Également, nous proposons de préserver la géométrie d'une charnière de pli dans des zones sous-échantillonnées en imposant l'axe du pli comme direction préférentielle de rigidité. Ce critère d'anisotropie est ensuite généralisé en une énergie 2D décrite localement par une ellipse d'anisotropie. L'avantage commun à ces approches est qu'elles sont décrites de manière continue sur le domaine d'étude, et sont donc définies indépendamment d'une discrétisation donnée.

Chapter 5

Improving the continuous problem

5.1 Motivation: the limits of the bending energy in implicit structural modeling

In the previous chapters, the bending energy has been presented as robust enough to deal with sparse, irregular and noisy data (Chapters 3, p.91 and 4, p.119). This ability is key for complex applications. However, as it is the case for other smoothing regularizations, it performs badly on anisotropic, periodic and thickness variation features where data are missing (Section 1.3.3, p.50). The same limits are met with the bending energy.

For instance, Figure 5.1 shows two modeling limits computed with the MLS discretization (such limits are also met with the ghost discretization). Figure 5.1(a) deals with the thickness variation issue already tested with DSI and PFM (Figure 1.26, p.50). The same *bubble effect* is observed in the stratigraphic field. In Figure 5.1(c), the implicit function bends abnormally close to the fault, which could be interpreted as a border effect. This is in fact related to the length of the fault: the fault throw is assumed to be null at a fault tip, but the tip is here close to data points imposing a non-null throw. The implicit function smoothly accommodates in between, hence the observed trend.

A first approach to deal with these issues is mentioned in Section 1.4 (p.50): smoothing issues may be managed by improving the quality of the data (Hillier *et al.*, 2014; Laurent, 2016; Laurent *et al.*, 2016; Grose *et al.*, 2017; de la Varga *et al.*, 2019). This is efficient but it generally requires an expertise of the studied geological structures, some manual interactions, and subjective interpretations. As an example, additional normal gradient data are used in Figure 5.1(b) to address the thickness variation. These data were randomly extracted from a densely interpreted version of this model. In Figure 5.1(d), the fault is manually extended to smoothly accommodate the variation of the throw.

In this section, we present a second approach: the data set is unchanged and we focus on modifying the modeling problem instead, trying to reduce manual interactions. Actually, some of the proposed solutions do involve manual interactions (e.g., the ones imposing anisotropy), but they have the potential to be automated. Possible modifications of the standard framework of this thesis (Section 2.2, p.79) are presented to handle specific geological settings.

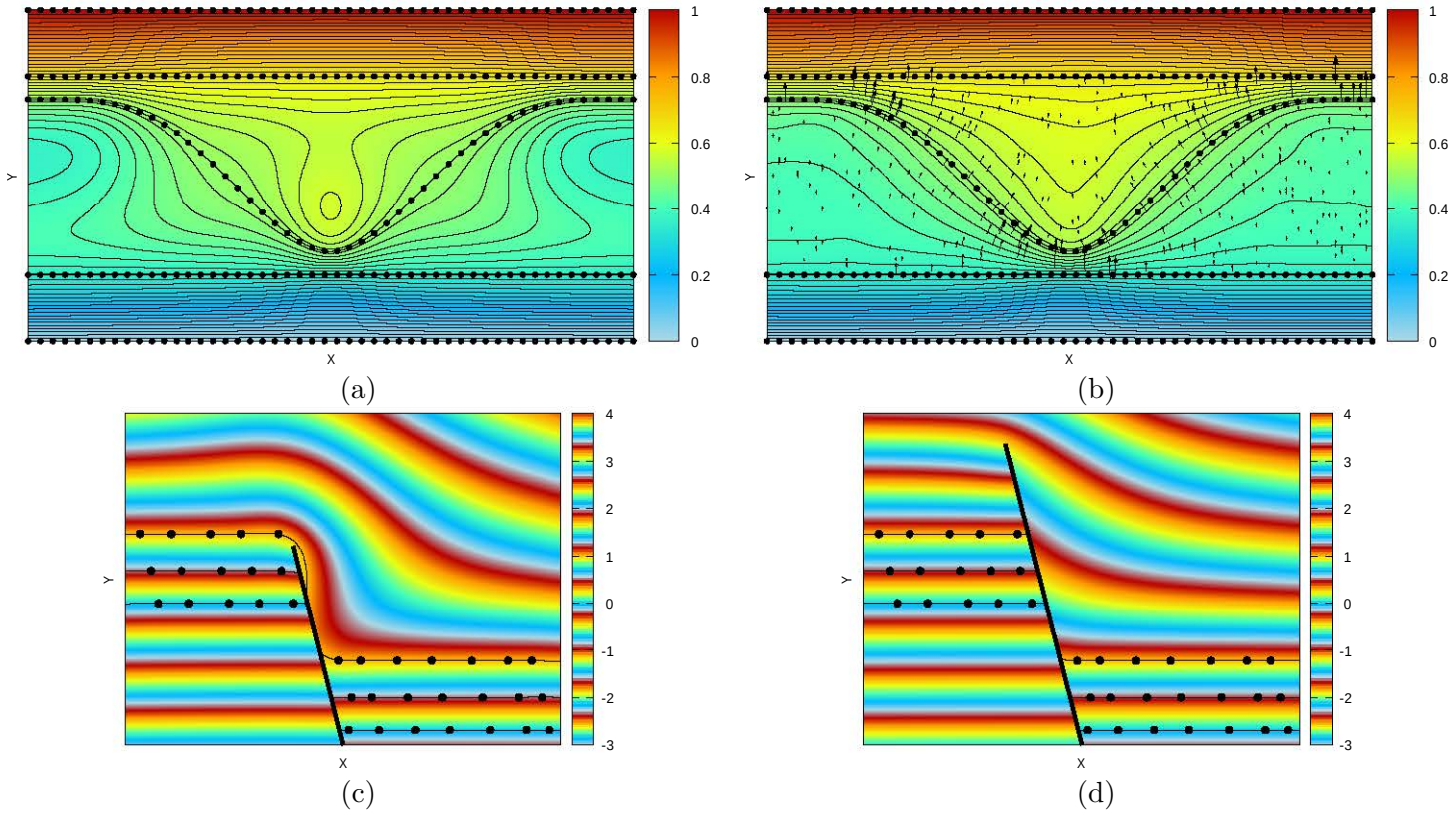


Figure 5.1: *Two modeling limits when employing a smooth energy and two examples of correction with artificial data. (a) and (b) An under-sampled thickness variation issue, (c) and (d) A fault vanishing abruptly in space.*
(a) and (b) Iso-values from bottom to top: $[0, 0.35, 0.5, 0.65, 1]$, $\lambda_\epsilon = 0.2$, $\lambda_{\boldsymbol{p}} = 10$,
(c) and (d) Iso-values from bottom to top: $[0, 0.5, 1]$, $\lambda_\epsilon = 1$, $\lambda_{\boldsymbol{p}} = 1$.

5.2 Replacing the bending energy by other well known smoothing energies

The proposed framework offers the possibility to test other energies than the bending energy in the general minimization problem (Equation 5.1, p.136) repeated here:

$$\min_u \left(J(u)\right) = \min_u \left(J_E(u) + J_{data}(u)\right), \tag{5.1}$$

with J_{data} the set of functionals associated to the data constraints and J_E the set of functionals of continuous energies. Here, well known energies are tested on a simple data set to discuss their applicability to implicit structural modeling.

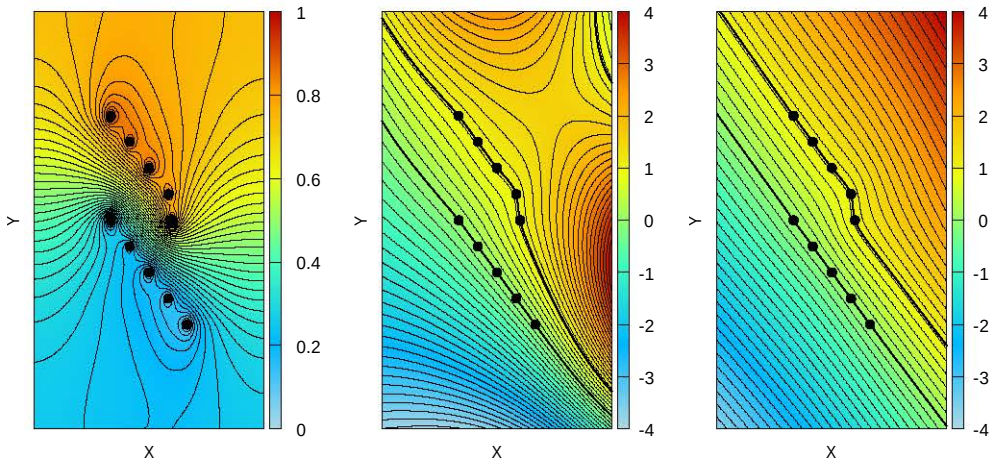
In Figure 5.2(a), the functional J_E is replaced by the functional $J_{Dirichlet}$ of the Dirichlet energy (Courant, 1950)

$$J_{Dirichlet}(u) = \frac{1}{2}\int_\Omega \lambda_\epsilon^2 ||\nabla u||^2 d\Omega. \tag{5.2}$$

The obtained implicit function collapses on the data points to fit the imposed iso-values and it tends to be constant where data are missing and perpendicular to the borders. This generates high curvatures in the extracted iso-values, which is not realistic geologically (Section 1.1.5, p.27). Therefore, the Dirichlet energy is not adapted to the studied structural modeling problem.

In Figure 5.2(b), the functional J_E is replaced by the functional $J_{Laplacian}$ of the Laplacian energy

$$J_{Laplacian}(u) = \frac{1}{2}\int_\Omega \lambda_\epsilon^2 (\partial_{ii}^2 u)^2 d\Omega, \tag{5.3}$$



(a) (b) (c)

Figure 5.2: *Test of several penalization energies on a synthetic data set with two parallel horizons except for one shifted data point. (a) Dirichlet energy, (b) Laplacian energy, (c) Bending energy.*
Iso-values from bottom to top: $[0, 1]$, $\lambda_\epsilon = 1$, $\lambda_{\boldsymbol{p}} = 1$.

with a sum assumed on i in each coordinate of space. The obtained implicit function is close to the one created with the bending energy (Figure 5.2(c)), but the shifted data point makes it flip at the top. The Laplacian energy is not adapted to the studied structural modeling problem.

Other ideas could involve mechanical energies, other energies minimizing the curvature of a surface or a volume, or a mix of these energies. The best continuous energy fitting geological structures is yet to be found but we think that the proposed framework can be used to test new ideas.

5.3 Modifying the bending energy

Another approach to improve the modeling problem is to tune the parameters of the bending energy to fit a given data set. In this section, we suggest to use the physical meaning of this energy to vary spatially its weight on the system of equations to handle thickness variations. We focus on 2D applications as the used concepts are straightforward in this dimension, but it can be adapted to any other dimensions.

5.3.1 Physical meaning of the bending energy

The bending energy is also called thin plate energy (Wahba, 1990) as it is used to control the global curvature of a thin plate being bended in space. This is directly applicable to implicit structural modeling in 2D with the concept of underlying topographic 3D surface of a model.

In 2D, a position $\boldsymbol{p}(x, y) \in \Omega$ is associated to an implicit value $u(\boldsymbol{p})$. If projected on the z axis of an orthonormal coordinate system $\{x, y, z\}$ such as $z(\boldsymbol{p}) = u(\boldsymbol{p})$, it defines a surface in 3D. This topographic surface can be seen as a thin plate being bended to fit data points spatially.

Figure 5.3 illustrates the topographic 3D surfaces obtained for three different smoothing weights λ_ϵ on a noisy data set (decimated and perturbed from Figure 3.4(a), p.96). With a great λ_ϵ, the mean plan passing through the data points is computed (Figures 5.3(a) and (d)). Then, the smaller λ_ϵ, the more the surface can deviate from the mean plan to fit the noisy data points (Figures 5.3(b), (c), (e) and (f)). It is only because this deviation is allowed with rigidity that the bending energy is convenient for structural modeling: the global mean plan is not recovered abruptly where data are missing as it would increase the curvature energy value. If the noise's range is not too wide as compared to the structures, it is possible to find a degree of global rigidity that approximates geological structures while filtering noise pollution (Figure 5.3(b) and (e)).

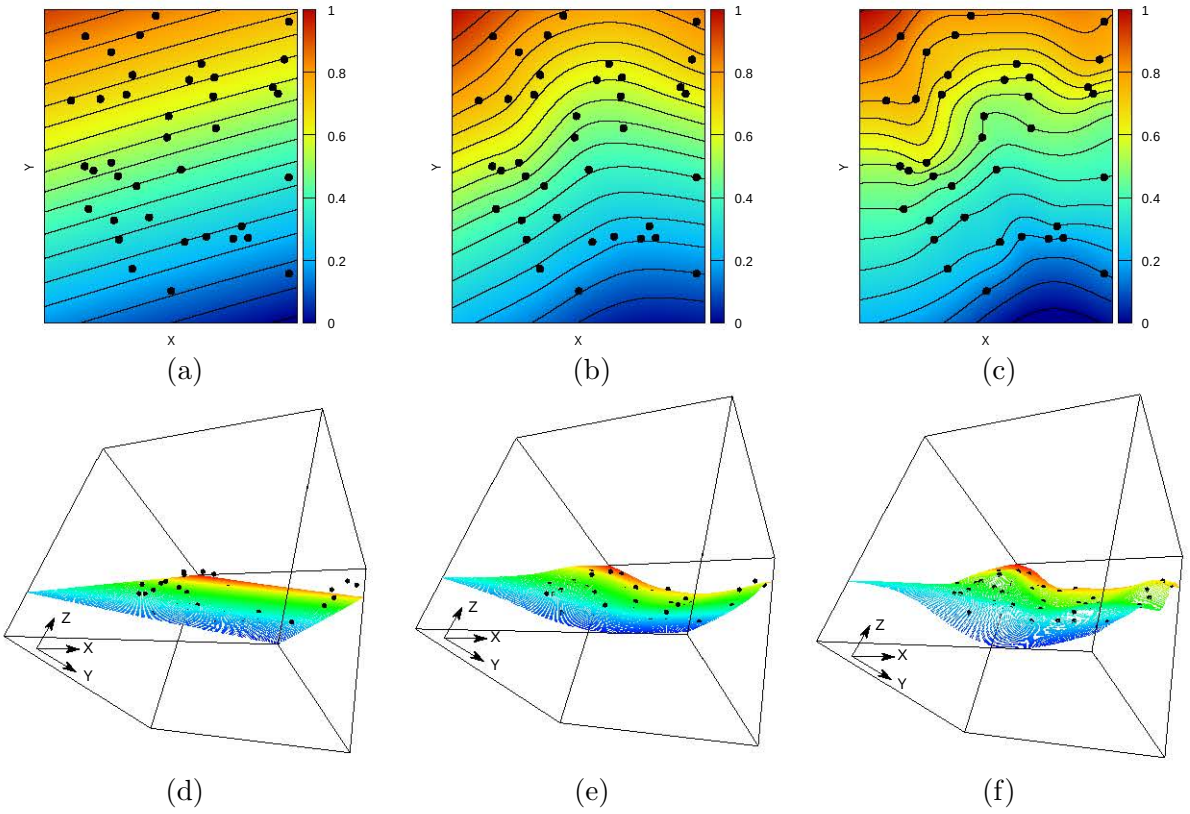


Figure 5.3: *Implicit functions and underlying topographic 3D surfaces computed on the same set of perturbed data points but with different energy weights* λ_ϵ.
Iso-values from bottom to top: $[0, 0.07, 0.14, 0.20, 0.26, 0.32, 0.38, 0.43, 0.48, 0.53, 0.58, 0.64, 0.70, 0.74, 0.80, 0.87, 0.92, 1]$.
(a) and (d) $\lambda_\epsilon = 2$, *(b) and (e)* $\lambda_\epsilon = 0.5$, *(c) and (f)* $\lambda_\epsilon = 0.05$.
$\lambda_{\boldsymbol{p}} = 1$.

5.3.2 Thickness variation context

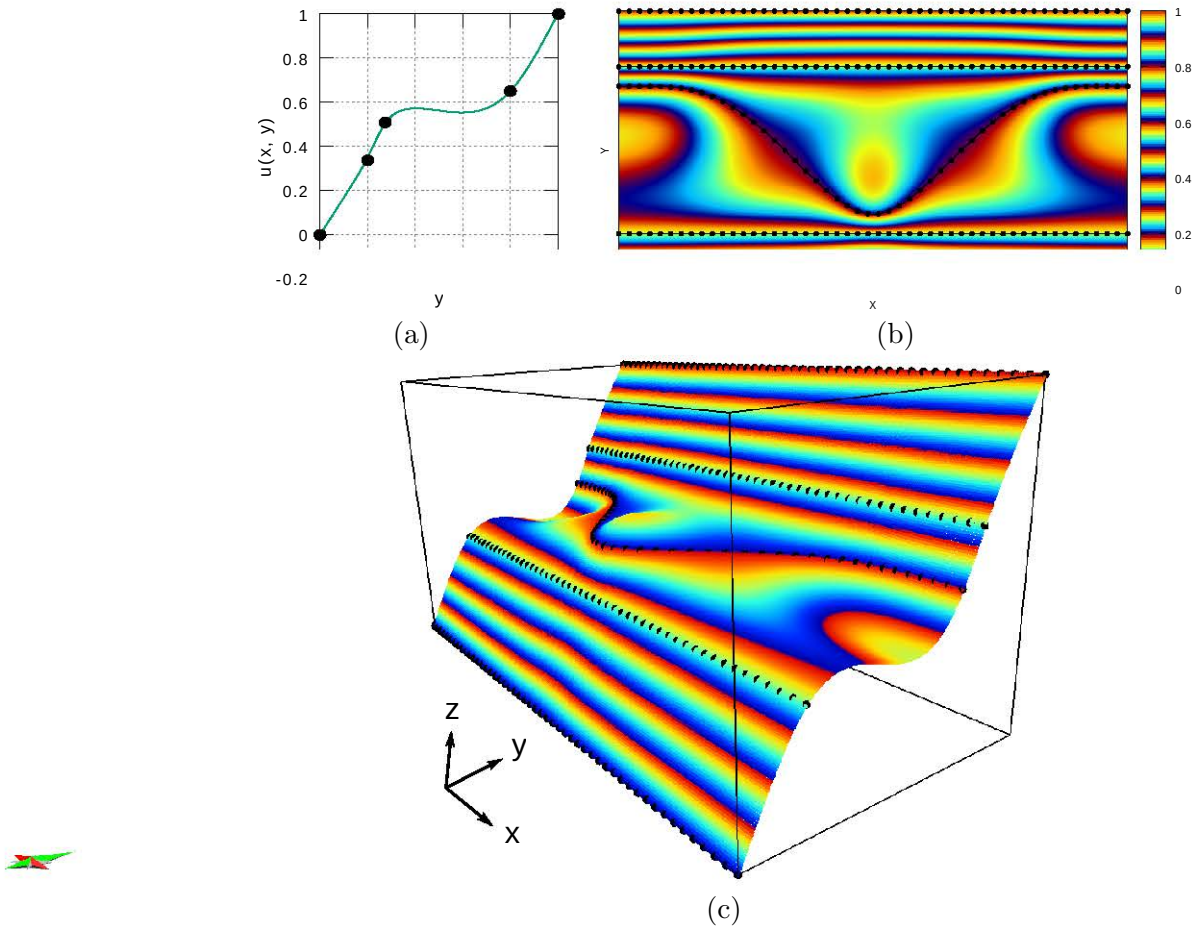
The concept of topographic 3D surface gives another way of understanding the limits of smoothing with thickness variation issues. The model of Figure 5.1(a) is re-plotted in Figure 5.4 in 1D, 2D and 3D with a banded color scale to better illustrate the issue. On this model, no noise is affecting the data points. An expected solution would therefore pass exactly through these data, which is why a small λ_ϵ value is chosen as compared to the $\lambda_{\boldsymbol{p}}$ value. The deviation from the mean plan imposed by the data points in this configuration implies the three following equivalent issues: (i) a non-monotonicity of the z function within layers on the cross section in 1D (Figure 5.4(a)), (ii) closed iso-surfaces in the implicit function in 2D (Figure 5.4(b)), and (iii) local bumps and holes in the underlying topographic 3D surface (Figure 5.4(c)). The local extrema are imposed by the abrupt change in the gradient's norm to fit all the data points. With the rigidity of the bending energy, the surface is smoothly curved to transit from one gradient to the other, and is necessarily characterized by this non-monotonicity in the function.

This example is not isolated. All thickness variations impose a change in the gradient's norm, which is not the case with isopaque folds for instance. Depending on the intensity of this variation in space, stratigraphic inconsistencies as presented here may or may not be created.

5.3.3 Correction with the Weighted Curvature Minimization criterion

5.3.3.1 Concept

The Weighted Curvature Minimization (WCM, Renaudeau *et al.* (2018)) locally modulates the rigidity of the underlying topographic 3D surface to fix the thickness variation issues.



(a) (b) (c)

Figure 5.4: *Stratigraphic inconsistency of a 2D model with thickness variation observed in three different dimensions. (a) Cross section along the y axis realized in the middle of the model (x is fixed), (b) The 2D stratigraphic function, (c) The underlying topographic 3D surface.*
Iso-values from bottom to top: $[0, 0.35, 0.5, 0.65, 1]$, $\lambda_\epsilon = 0.2$, $\lambda_{\boldsymbol{p}} = 10$.

Avoiding local extrema in Figure 5.4(a) is equivalent to state that the sought solution is a linear by parts (or monotonous) function, each part being an interval between two horizons (i.e., a layer). In other words, we want to minimize the curvature of the underlying topographic 3D surface within the layers, and maximize it close to horizon data. The WCM criterion therefore consists in weighting the smoothing energy differently in space. Let r be the distance from a position $\boldsymbol{x}$ to the closest data point $\boldsymbol{p}$ as

$$\forall \boldsymbol{x} \in \Omega, \quad r = min\left(||\boldsymbol{p} - \boldsymbol{x}||,\ \forall \boldsymbol{p} \in \boldsymbol{D}\right). \tag{5.4}$$

The standard minimization problem (Equation (2.55), p.81) is then augmented to

$$\min_u \left(\frac{1}{2} \int_\Omega \lambda_\epsilon(r)^2 \left(\partial_{ij}^2 u\right)^2 d\Omega + \frac{1}{2} \sum_{\boldsymbol{p} \in \boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \left(u(\boldsymbol{p}) - \alpha_{\boldsymbol{p}}\right)^2 \right), \tag{5.5}$$

where λ_ϵ is dependent on r and varies between a minimum value λ_ϵ^{min} reached at data point positions, and a maximum value λ_ϵ^{max} reached away from any data points. The thin plate is therefore more flexible close to the data points than away from them.

Figure 5.5 illustrates possible functions of λ_ϵ depending on r. As the dilatation parameter, the range ρ can be given as a radial distance for a spherical support (i.e., as presented here), or a vector $\boldsymbol{\rho}(\rho_x, \rho_y)$ for a cubic support (i.e., defining $\lambda_\epsilon(r)$ as in Equation (2.27), p.66). Nonetheless, this range can be different from the dilatation parameter.

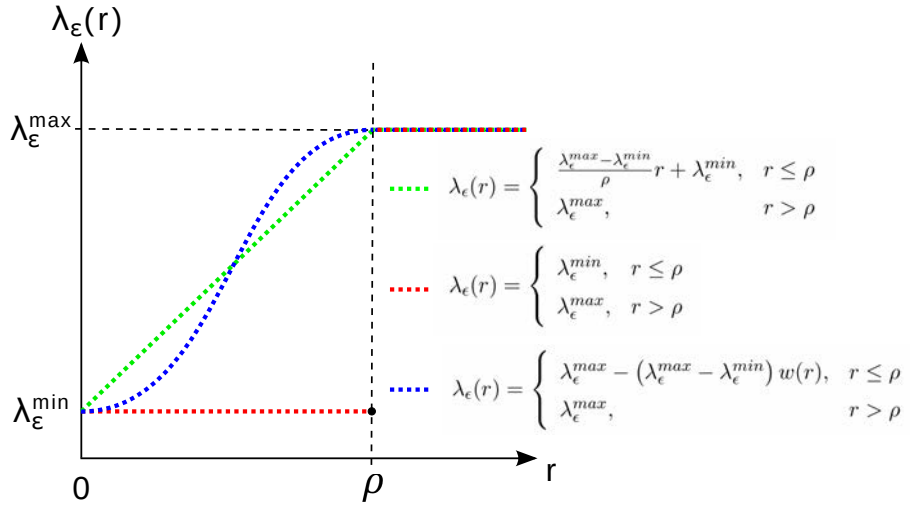


Figure 5.5: *Three examples of variation for $\lambda_\epsilon(r)$ within a range ρ and between minimum and maximum values λ_ϵ^{min} and λ_ϵ^{max}. The function w can be any weight function such as the forth order spline (Equation (3.10)).*

5.3.3.2 Discretization

The minimization problem with the WCM criterion (Equation (5.5)) can be discretized as done for the standard minimization problem (Section 2.2.4, p.82). In the two presented prototypes (Chapters 3 and 4), the energy equations are written on each discretization node. The distance r is then centered on each node $\overline{\boldsymbol{p}}$ as

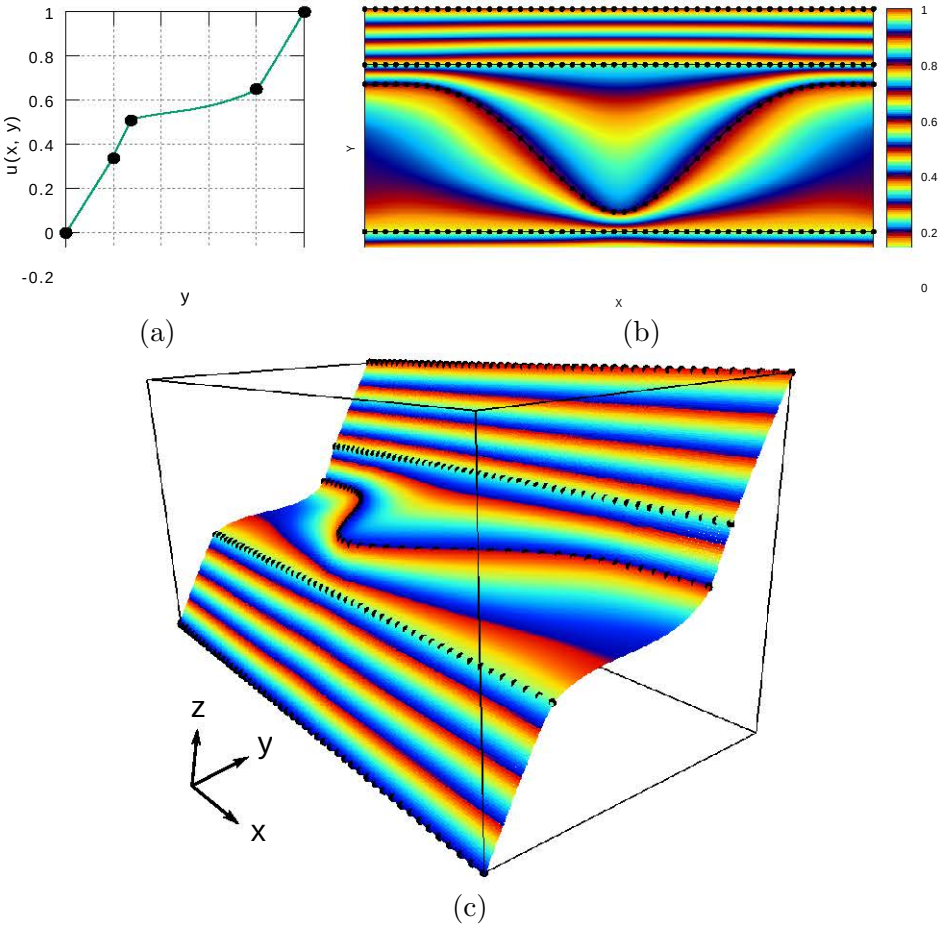
$$\forall \overline{\boldsymbol{p}} \in \boldsymbol{N}, \quad r = min\left(||\boldsymbol{p} - \overline{\boldsymbol{p}}||,\ \forall \boldsymbol{p} \in \boldsymbol{D}\right), \tag{5.6}$$

and the varying weight $\lambda_\epsilon(r)$ replaces the constant weight λ_ϵ in systems (3.2.6) (p.95) and (4.2.6) (p.126).

5.3.3.3 Applications

The WCM criterion is applied on the thickness variation issue presented in Figures 5.1(a) and 5.4 with the MLS discretization. The Heaviside function of Figure 5.5 (i.e., the red one) is used to control the weight $\lambda_\epsilon(r)$. The range $\boldsymbol{\rho}$ is chosen equal to the standard dilatation parameter of the MLS functions (Section 3.2.3.1, p.93) so that at least nine nodes around a data point are associated with a small weight λ_ϵ^{min}. The number of discretization nodes is chosen fine enough to have nodes associated with λ_ϵ^{max} within the areas of thickness variation (i.e., (50×50)). This results in a 1D cross section (Figure 5.4(a)) close to a piecewise linear function and the disappearing of the closed iso-surfaces in the 2D implicit function or the bumps and holes in the 3D surface.

This concept can possibly handle any intensity of thickness variation in the layers. It also preserves the weighting system and the smoothing capability: although the 3D surface is more flexible close to the data points, it can still be rigid enough to filter the noise. Finally, this criterion is unrelated to the way discontinuities are handled, so it can also apply to profiles with both thickness variation and discontinuities. Figure 5.7 shows a few examples of synthetic 2D models handled with the WCM criterion.



(a) (b) (c)

Figure 5.6: *2D model with thickness variation handled with the WCM criterion. (a) Cross section along the y axis realized in the middle of the model (x is fixed), (b) The 2D stratigraphic function, (c) The underlying topographic 3D surface.*
Iso-values from bottom to top: $[0, 0.35, 0.5, 0.65, 1]$, $\lambda_\epsilon^{max} = 1$, $\lambda_\epsilon^{min} = 0.2$, $\lambda_{\boldsymbol{p}} = 10$.

5.3.4 Limits of the Weighted Curvature Minimization criterion

The WCM criterion introduces several additional parameters to the modeling problem: one extra weight value, the function of variation of the smoothing weight, and the range of influence of this function away from the data points. It also requires the nodal sampling to be fine enough to have nodes associated to different weight values within each layer with thickness variations. These conditions make it difficult to tune all the parameters together, hence necessitating both expertise of the algorithm and multiple computations of a same model before obtaining a satisfactory result. In addition, there is no theoretical proof that a combination of these parameters can always be found to produce a result without stratigraphic inconsistencies.

As the WCM criterion is data driven, it also tends to produce high curvature transitions between sampled areas and areas missing data points. In Figure 5.8, angles are observable in the extracted iso-surfaces both at the last data points close to the borders and close to the center where data points are missing. By weighting the bending energy differently in space, the WCM criterion allows the underlying topographic 3D surface to deviate from the mean plan in a piecewise linear manner within the layers. In the absence of data points, the mean plan is recovered. As the 3D surface is more flexible around data points, the mean plan is abruptly recovered from sampled areas to non-sampled areas, hence the observed trend. If the data points close to non-sampled areas are coplanar with the mean plan, these curvatures are not observed.

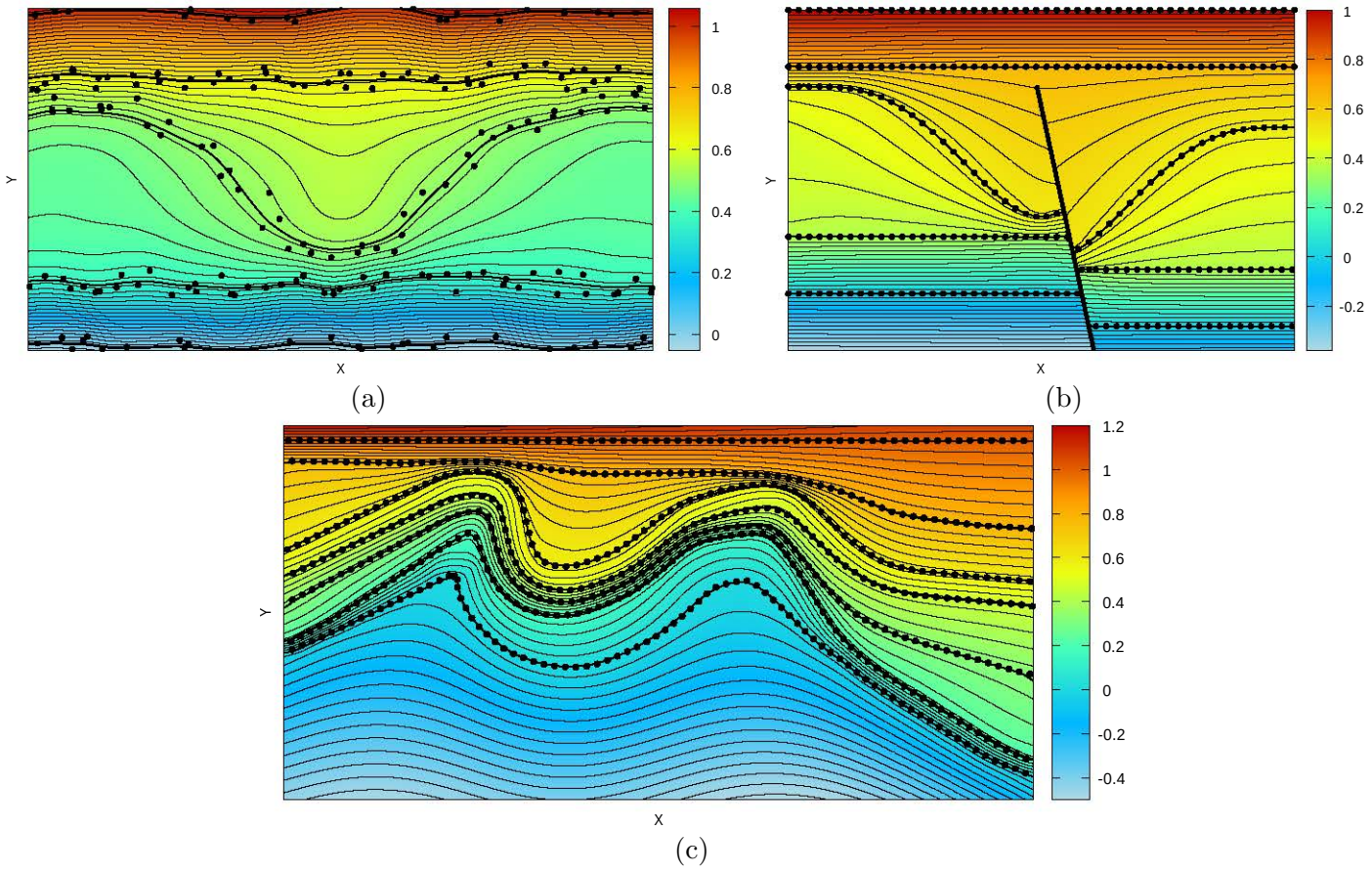


Figure 5.7: *Three synthetic examples of thickness variation issues dealt with the WCM criterion. (a) and (b) The model from Figure 5.6 with (a) added noise in the data points and (b) an added normal fault, (c) A model with multiple cases of thickness variation in the layers. (a) and (b) Iso-values from bottom to top:* $[0, 0.35, 0.5, 0.65, 1]$, *(a)* $\lambda_{\epsilon}^{max} = 6$, $\lambda_{\epsilon}^{min} = 1$, $\lambda_{\boldsymbol{p}} = 10$, *(b)* $\lambda_{\epsilon}^{max} = 1$, $\lambda_{\epsilon}^{min} = 0.2$, $\lambda_{\boldsymbol{p}} = 10$, *(c) Iso-values from bottom to top:* $[0, 0.18, 0.32, 0.46, 0.58, 0.77, 1]$, $\lambda_{\epsilon}^{max} = 1$, $\lambda_{\epsilon}^{min} = 0.1$, $\lambda_{\boldsymbol{p}} = 1$.

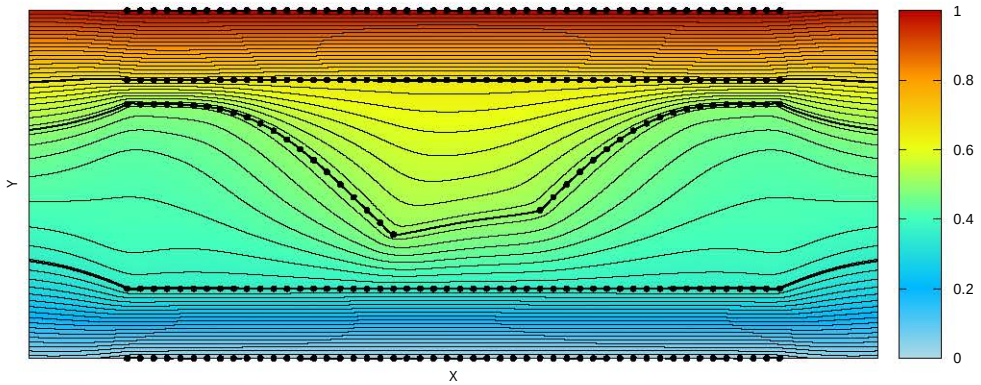


Figure 5.8: *Limit of the WCM criterion: high curvatures are produced at the edge of areas missing data points.*
Iso-values from bottom to top: $[0, 0.35, 0.5, 0.65, 1]$, $\lambda_{\epsilon}^{max} = 1$, $\lambda_{\epsilon}^{min} = 0.2$, $\lambda_{\boldsymbol{p}} = 10$.

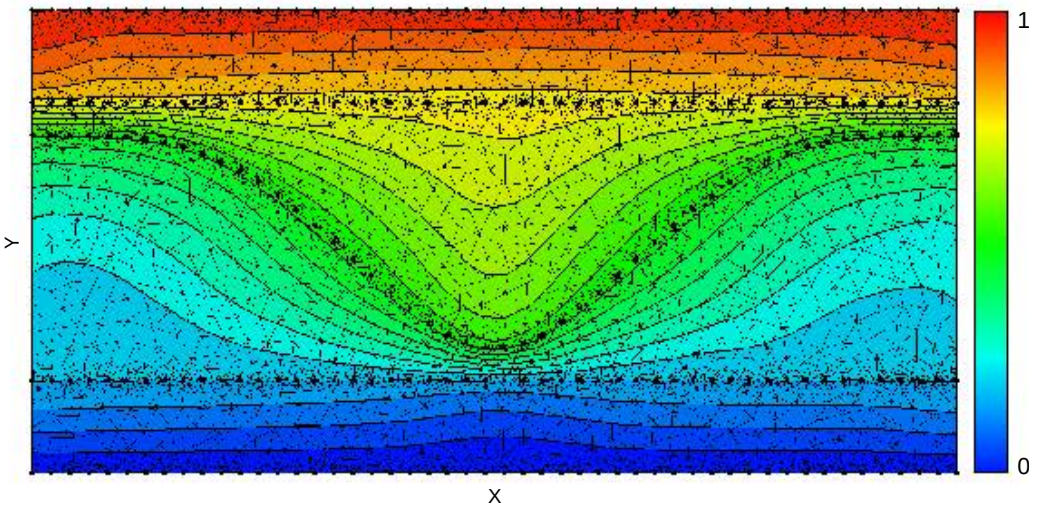


Figure 5.9: *Undersampled thickness variation issue handled with Volume Based Modeling using a mesh with adaptive resolution.*

5.3.5 Relation with Discrete Smooth Interpolation on meshes with adaptive resolution

Similar results to the ones computed with the WCM criterion can be obtained, to some extent, with the Volume Based Modeling software we used in Section 1.3.1 (p.34) to compute DSI results. This is possible when using an adaptive resolution of the mesh which is finer on the data points, and coarser away from them. If the transition is abrupt enough to have coarser mesh elements within the layers with thickness variation, the stratigraphic inconsistencies shown in Figure 1.26(a) may be avoided as shown in Figure 5.9.

In this version of DSI, there is a direct relation between the weight of the smoothing factor and the size of an element, but this relation is not balanced properly in the global system of equations. Therefore, although the number of finer elements and smoothing equations increase close to the data points, their associated weights are overly reduced. It results in an effect on the system of equations equivalent to the WCM criterion's. The main difference is that in Volume Based Modeling, it is a secondary effect of the discretization leading to an unbalanced system of equations which cannot be easily controlled. In comparison, the WCM criterion is defined in the continuous problem which ensures a better control of the interpolation and allows to handle thickness variations even with a homogeneous discretization.

5.4 Enriching the continuous problem with an anisotropic direction

It is possible to improve the modeling problem by adding specific functionals to the standard minimization problem. This can help to adapt the proposed approach to relate, for instance, the local anisotropy of a data set or an expected geological structure.

5.4.1 Concerned issues and existing solutions

In Martin & Boisvert (2017), anisotropy of ore volumes is captured by partitioning the domain of study and modifying the Radial Basis Functions (RBF) locally. This is performed iteratively and automatically by modifying step by step the shape of the partitions in regards to the anisotropy computed in each partition separately. Concerning the interpolation, a rotation matrix relates the anisotropy in each partition when evaluating the RBF (i.e., the points are artificially rotated to evaluate their mutual distances). As RBF can be linked to the minimization problem (Section 2.2.2, p.80), the way they impose anisotropy on their interpolation can be seen as a specific enrichment of the smoothing energy. Our aim is then to impose a direction of anisotropy directly on the minimization problem for a structural modeling application.

Capturing the full geometry of an under-sampled fold often requires to incorporate global or local anisotropy. In Hillier *et al.* (2014) (Fig. 2), the proposed isotropic RBF interpolation does not preserve the geometry of a studied curved fold but smoothes it instead where data are missing. The fold is then reproduced by adding tangent data along the fold's hinge line. This technique is efficient and we do not pretend to bring a better solution, only an alternative one.

5.4.2 Imposing a direction of anisotropy on the minimization problem

We start by considering a straight fold modeled with three interpreted cross sections distant one from another. In Figure 5.10, only one of the three horizons is extracted from the stratigraphic function, generated with the MLS discretization, to better observe the features. As in Hillier *et al.* (2014), the smoothing performed by the bending energy does not preserve the hinge of the fold in Figure 5.10(a), creating these saddle shaped transitions between the cross sections.

To solve this issue, we propose to increase the rigidity of smoothing in the axial direction of the fold. Let $\boldsymbol{v}$ be a unit vector along this direction. The second order development of the implicit function's value evaluated at the position $(\boldsymbol{x}+\boldsymbol{v})$ is

$$\forall(\boldsymbol{x},\boldsymbol{x}+\boldsymbol{v})\in\Omega,\quad u(\boldsymbol{x}+\boldsymbol{v})=u(\boldsymbol{x})+\nabla u(\boldsymbol{x})\cdot\boldsymbol{v}+\frac{1}{2}\boldsymbol{v}^T\cdot\nabla^2 u(\boldsymbol{x})\cdot\boldsymbol{v}, \tag{5.7}$$

with $\nabla^2=\boldsymbol{H}$ the Hessian operator. A preferential rigidity of smoothing along the direction of the vector $\boldsymbol{v}$ can be enforced by minimizing the second order residual

$$\forall(\boldsymbol{x},\boldsymbol{x}+\boldsymbol{v})\in\Omega,\quad \min(u(\boldsymbol{x}+\boldsymbol{v})-(u(\boldsymbol{x})+\nabla u(\boldsymbol{x})\cdot\boldsymbol{v}))\Leftrightarrow min\left(\frac{1}{2}\boldsymbol{v}^T\cdot\nabla^2 u(\boldsymbol{x})\cdot\boldsymbol{v}\right). \tag{5.8}$$

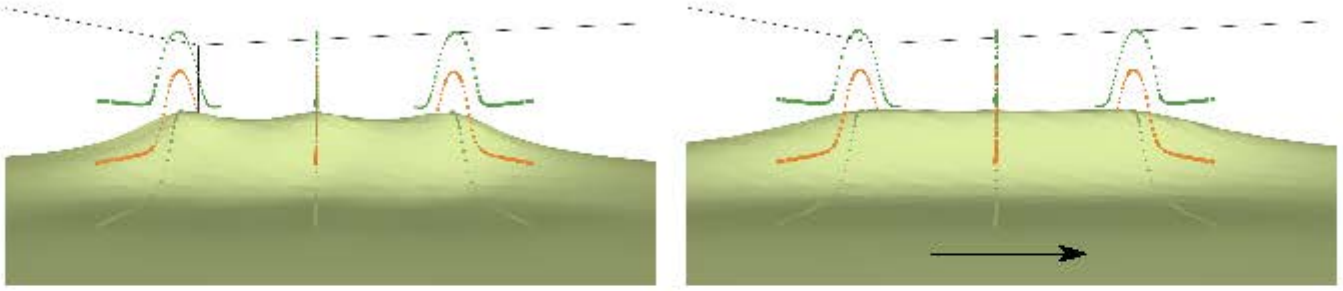
In 2D applications, this is equivalent to minimizing the distance between the topographic 3D surface and its tangent line at any position $\boldsymbol{x}$ along $\boldsymbol{v}$.

The standard minimization problem is augmented to

$$\min_u\left(\frac{1}{2}\int_\Omega\lambda_\epsilon^2\left(\partial_{ij}^2 u\right)^2 d\Omega+\frac{1}{2}\int_\Omega\lambda_v^2\left(\boldsymbol{v}^T\cdot\boldsymbol{H}(u)\cdot\boldsymbol{v}\right)^2 d\Omega+\frac{1}{2}\sum_{\boldsymbol{p}\in\boldsymbol{D}}\lambda_{\boldsymbol{p}}^2\left(u(\boldsymbol{p})-\alpha_{\boldsymbol{p}}\right)^2\right), \tag{5.9}$$

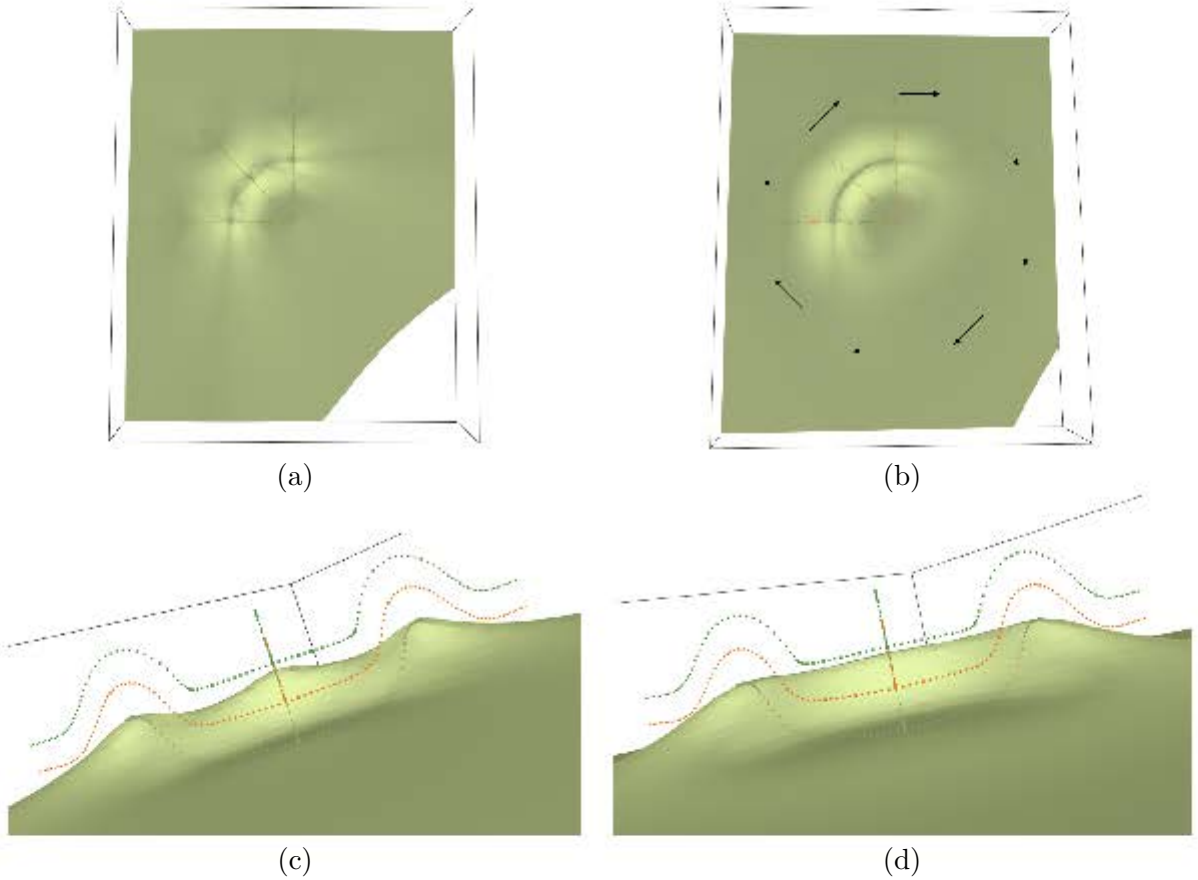
with λ_v the analogous weight of λ_ϵ for the rigidity in the direction of anisotropy. Figure 5.10(b) shows a result obtained with the MLS discretization of Equation (5.9) using a constant vector $\boldsymbol{v}$ along the fold's axial direction. The rigidity term is thus constantly added everywhere in the model, which is equivalent to impose a global direction of anisotropy.

It is also possible to vary $\boldsymbol{v}$ in space to relate local directions of anisotropy. In Figure 5.11 the hinge line of the fold is curved, so the vector $\boldsymbol{v}$ is chosen to follow the expected fold's axial direction to reduce the saddle shaped transitions where data are missing. Note that it is also possible to control the magnitude of anisotropy by varying λ_v in space as done with λ_ϵ in Section 5.3.3.



(a) (b)

Figure 5.10: *3D model of a fold sampled with three distant cross sections. All the data points were used to compute the implicit function, but only one horizon was extracted for visualization. (a) Result of the standard minimization problem, (b) Result of the standard minimization problem augmented with the arrow's direction of anisotropy.*
$\lambda_\epsilon = 1$, $\lambda_v = 10$, $\lambda_{\boldsymbol{p}} = 10$.



(a) (b)

(c) (d)

Figure 5.11: *3D model of a circular fold sampled with three distant cross sections. All the data points were used to compute the implicit function, but only one horizon was extracted for visualization. (a) and (c) Result of the standard minimization problem, (b) and (d) Result of the standard minimization problem augmented with a variable direction of anisotropy (in circle as indicated by the arrows).*
$\lambda_\epsilon = 1$, $\lambda_v = 10$, $\lambda_{\boldsymbol{p}} = 10$.

5.5 Replacing the bending energy by an anisotropic energy

5.5.1 Equivalence between isotropic smoothing and bending energy

The first step is to verify that the bending energy is isotropic. In 2D, the isotropic version of the rigidity enrichment proposed in Section 5.4.2 can be obtained by integrating, in all the directions of space, the unit vector $\boldsymbol{v}_{iso}$

$$\forall \theta \in [0, 2\pi], \quad \boldsymbol{v}_{iso}(\theta) = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}. \tag{5.10}$$

It defines the functional J_{iso} as

$$J_{iso}(u) = \frac{1}{2}\int_{\Omega} \lambda_{iso}^2 \int_0^{\pi} \left(\boldsymbol{v}_{iso}(\theta)^T \cdot \boldsymbol{H}(u) \cdot \boldsymbol{v}_{iso}(\theta)\right)^2 d\theta d\Omega. \tag{5.11}$$

After integration, detailed in Appendix I, this gives

$$J_{iso}(u) = \frac{1}{2}\int_{\Omega} \lambda_{iso}^2 \frac{\pi}{4} \left(\left(\partial_{xx}^2 u\right)^2 + \left(\partial_{yy}^2 u\right)^2 + 2\left(\partial_{xy}^2 u\right)^2 + \frac{1}{2}(\partial_{xx}^2 u + \partial_{yy}^2 u)^2 \right) d\Omega. \tag{5.12}$$

Therefore, the bending energy is not exactly the isotropic version of the directional rigidity enforcement. However, by the positiveness of the integral, when the functional J_{iso} is minimized, the Laplacian term (i.e., $(\partial_{xx}^2 u + \partial_{yy}^2 u)^2$) should already be minimized by the two first terms (i.e., $(\partial_{xx}^2 u)^2$ and $(\partial_{yy}^2 u)^2$). In the proposed discretizations, the bending energy represents then an approximation of J_{iso} in the least squares sense.

5.5.2 Anisotropy as an ellipse

The bending energy can be approximated as an isotropic energy. The enrichment proposed in Section 5.4.2 is reduced to only one direction of anisotropy. In 2D, the system gathering these two functionals (i.e., Equation (5.9)) is thus imposing a smoothing with a first direction of anisotropy along $\boldsymbol{v}$ with the magnitude λ_v (i.e., with $||\boldsymbol{v}|| = 1$), and a second direction perpendicular to $\boldsymbol{v}$ with a magnitude λ_ϵ. The issue is that the transition between these two magnitudes is not continuous but jumps abruptly to λ_ϵ for any direction linearly independent from $\boldsymbol{v}$. The goal is thus to suggest an energy which defines an elliptical anisotropy.

An anisotropic energy can be constructed by modifying the demonstration of isotropy of the bending energy. The integration is performed with a vector $\boldsymbol{v}_{ani}$ defining an ellipse as

$$\forall(\theta, \alpha) \in [0, 2\pi], \forall a \in [0, 1], \quad \boldsymbol{v}_{ani}(\theta) = \begin{bmatrix} \cos\alpha \\ \sin\alpha \end{bmatrix} \cos\theta + \begin{bmatrix} -a\sin\alpha \\ a\cos\alpha \end{bmatrix} \sin\theta = \boldsymbol{v}_1 \cos\theta + \boldsymbol{v}_2 \sin\theta, \tag{5.13}$$

with α the angle between the direction of maximal magnitude and the axis x, and a the factor of reduced magnitude in the perpendicular direction. The ellipse's maximal magnitude is normalized as it is controlled by the anisotropic energy's weight λ_{ani}.

We then define the functional J_{ani} as

$$J_{ani}(u) = \frac{1}{2}\int_{\Omega} \lambda_{ani}^2 \int_0^{\pi} \left(\boldsymbol{v}_{ani}(\theta)^T \cdot \boldsymbol{H}(u) \cdot \boldsymbol{v}_{ani}(\theta)\right)^2 d\theta d\Omega. \tag{5.14}$$

After integration, detailed in Appendix J, the anisotropic energy's functional J_{ani} corresponds to

$$\begin{aligned} J_{ani}(u) = \frac{1}{2}\int_{\Omega} \lambda_{ani}^2 \frac{\pi}{4} \Big(& \left(\cos^2\alpha\ \partial_{xx}^2 u + \sin^2\alpha\ \partial_{yy}^2 u + \sin 2\alpha\ \partial_{xy}^2 u\right)^2 \\ & + a^4 \left(\sin^2\alpha\ \partial_{xx}^2 u + \cos^2\alpha\ \partial_{yy}^2 u + \sin 2\alpha\ \partial_{xy}^2 u\right)^2 \\ & + 2a^2 \left(\cos\alpha\ \sin\alpha(\partial_{yy}^2 u - \partial_{xx}^2 u) + \cos 2\alpha\ \partial_{xy}^2 u\right)^2 \\ & + \frac{1}{2}\left((\cos^2\alpha + a^2\sin^2\alpha)\ \partial_{xx}^2 u + (\sin^2\alpha + a^2\cos^2\alpha)\ \partial_{yy}^2 u + \sin 2\alpha(1 - a^2)\ \partial_{xy}^2 u\right)^2 d\Omega. \end{aligned} \tag{5.15}$$

As the bending energy, this functional is defined with squared terms of second derivatives of u, which is suitable for the two proposed least squares discretizations. It can therefore replace J_ϵ directly in the minimization problem. In this case, the minimization problem is comparable (but not equivalent) to Equation (5.9) when $\lambda_v \equiv \lambda_{ani}$, $\boldsymbol{v} \equiv [\cos\alpha \sin\alpha]$ and $\lambda_\epsilon \equiv a\ \lambda_{ani}$. This energy could also be written in 3D by integrating, in all the directions of space, the term $\left(\boldsymbol{v}_{ani}^T \cdot \boldsymbol{H}(u) \cdot \boldsymbol{v}_{ani}\right)^2$ where $\boldsymbol{v}_{ani}$ defines an ellipsoid. By modifying the ellipse of anisotropy in space (or ellipsoid in 3D), the interpolation and the modeled anisotropy can then be controlled locally.

5.5.3 Anisotropy replacing the weighted curvature minimization

The anisotropic energy (Equation (5.15)) can replace the weighted curvature minimization (WCM) criterion (Section 5.3.3). The main attribute of the WCM criterion is to control spatially the isotropic rigidity of the underlying topographic 3D surface of a 2D implicit model. To obtain a piecewise linear interpolation in the 1D cross section of Figure 5.6(a), it is not necessary to soften the 3D surface in all the directions of the 2D space, only perpendicularly to the expected iso-surfaces. This can be handled with the anisotropic functional J_{ani} using local ellipses aligned on the expected iso-surfaces and with a reduced second magnitude close to the data points.

In Figures 5.12(a) and (b), synthetic models of the angle α and the coefficient a are presented for the thickness variation issues shown in Figures 5.1(a), 5.4 and 5.6. The angle α is equal, at a given position, to the angle between the expected iso-surface and the x axis. The coefficient a is either small in the areas of expected high curvatures, or large in the remaining of the domain. Figure 5.12(c) shows the stratigraphic function obtained with the anisotropic energy using these local parameters. As a comparison, the model from Figure 5.6(b) obtained with the WCM criterion is shown in Figure 5.12(d) in the corresponding color scale. The used model is an unrealistic case of thickness variation to test the developed techniques on extreme cases. Therefore, we cannot compare these techniques in term of quality of interpolation with this model; they should be tested on more realistic models in the future.

This approach also enables to reduce the WCM criterion's effects at the edges of missing data areas. Figure 5.13 shows a model computed with the anisotropic energy using extended versions of the parameter models from Figures 5.12(a) and (b). These were extended on the borders with null values of α and horizontally prolongated a values. The observed angles on Figure 5.8, computed with the WCM criterion, are avoided on the borders and the fine layers are partly reproduced in the center of the model.

Further studies on the relations between the parameters of anisotropy and the quality of the results in different geological settings still need to be performed. Also, the next step would be to propose an algorithm to compute those local parameters automatically, which is a challenging problem already studied for instance in Martin & Boisvert (2017).

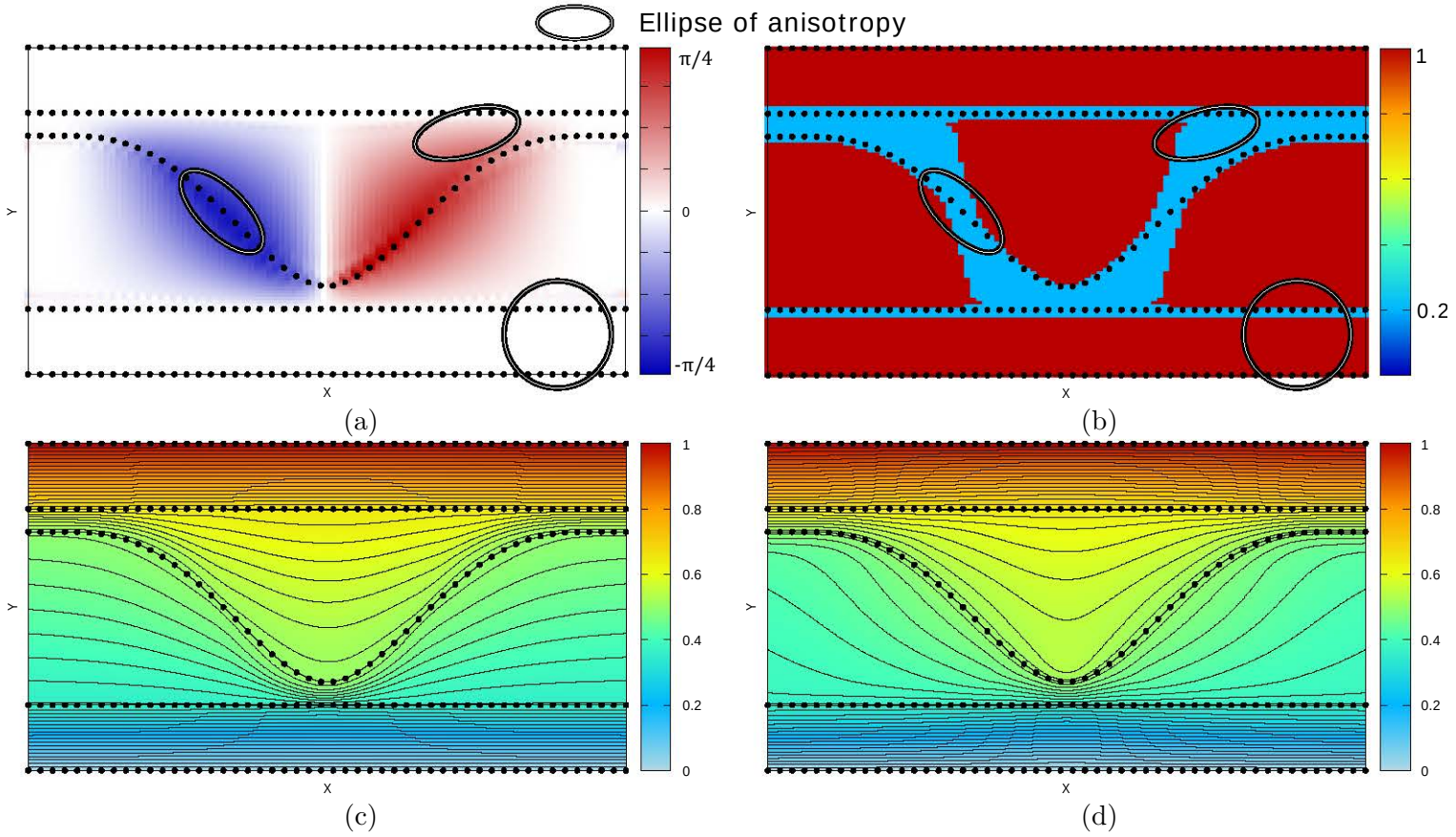


Figure 5.12: *2D model with thickness variation handled with the presented anisotropic energy. (a) Angle α giving the direction of first magnitude of anisotropy, (b) Coefficient a giving the second magnitude of anisotropy reached perpendicularly to the first magnitude. Some ellipses of anisotropy are represented as examples, (c) The stratigraphic function computed with the anisotropic energy using the parameters from (a) and (b), and (d) The corresponding model computed with the WCM criterion.*
Iso-values from bottom to top: $[0, 0.35, 0.5, 0.65, 1]$, $\lambda_{ani} = 1$, $\lambda_{\boldsymbol{p}} = 10$.

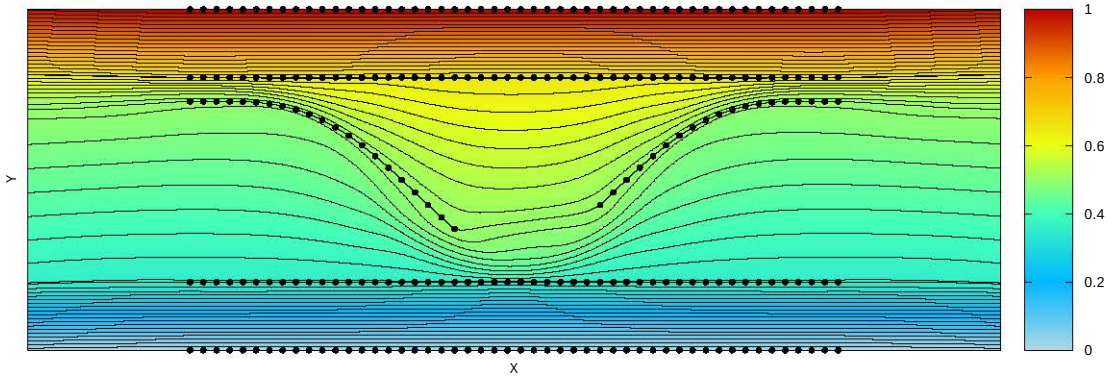


Figure 5.13: *Reducing the limits of the WCM criterion with the presented anisotropic energy: smoothing differently depending on the direction of space allows to perpetuate thickness variations away from the data points.*
Iso-values from bottom to top: $[0, 0.35, 0.5, 0.65, 1]$, $\lambda_{ani} = 1$, $\lambda_{\boldsymbol{p}} = 10$.

Résumé du Chapitre 6

Dans ce sixième chapitre, nous utilisons les moindres carrés glissants et les équations du problème continu proposé pour d'autres cas de modélisation géologique.

La première application concerne le développement d'un plugin pour Petrel spécialisé dans la modélisation de structures géologiques 3D, avec la particularité que seules des données interprétées à partir d'imageries de puits sont utilisées. L'ancienne version de ce plugin utilisait des méthodes de projections et des méthodes de modélisation explicite pour extrapoler les observations effectuées le long du trajet du puit. La chaine de modélisation regroupait un certain nombre d'étapes nécessitant l'intervention systématique de l'utilisateur. De plus, l'approche explicite pouvait créer des surfaces d'horizons qui s'intersectaient dans l'espace. Grâce à l'approche implicite, le plugin réalisé au cours de cette thèse ne comporte aucune étape intermédiaire et garanti la création de surfaces d'horizons non sécantes. L'application est présentée en détail avec les différents types de contraintes spécifiques à l'imagerie de puits.

La seconde application concerne la modélisation 3D d'enveloppes de volumes de sel. La principale différence avec une application de modélisation structurale est qu'une surface unique est modélisée. Ainsi, tous les points de donnée décrivent la même surface ou, en approche implicite, la même iso-valeur. Des points associés à des iso-valeurs différentes sont donc artificiellement ajoutés dans le domaine d'étude. Nous présentons plusieurs modèles synthétiques et deux cas d'application.

La dernière application introduit l'utilisation des moindres carrés glissants pour la restauration mécanique des structures géologiques. Les équations de la mécanique des milieux continus sont présentées et discrétisées en conséquence. L'approche est illustrée sur un cas synthétique de déplissement en 2D.

Chapter 6

Other modeling applications of the proposed workflow

In this section, we discuss the adaptability of this thesis's standard workflow to specific geomodeling applications: structural modeling restricted to borehole data, salt envelope modeling, and geomechanical restoration.

6.1 Structural modeling of borehole image interpretations

As an engineering project during the thesis, a prototype plugin to Petrel has been implemented in collaboration with Zhenhua Li and Philippe Marza (Schlumberger). This prototype enables the construction of structural models in the near-well space from interpreted borehole data such as logs, borehole images and deep-reading electromagnetic measurements.

6.1.1 Specificities of the application

This project is based on an already existing plugin to Petrel (non-commercialized) dealing with geological interpretation of borehole data. This plugin gathers interactive tools to visualize and interpret log data (gamma ray, photoelectric factor, density, neutron, resistivity), borehole images (density, resistivity, acoustic) and deep-reading electromagnetic measurements.

From the log data, formation tops are manually picked (i.e., hard data points). From the borehole images, formation dip vectors are either manually or automatically picked (i.e., gradient and tangent data). Other structural features such as fault evidences, or fold axes are also characterized with their position and dip orientation. From the deep-reading electromagnetic measurements, 2D vertical resistivity profiles are produced along the well trajectory (sub-horizontal well). These profiles show the formation layering and stratigraphic horizon evidences which can be manually picked (i.e., soft data points).

Data interpreted from the logs and borehole images are located along the well trajectory, on a polyline in the 3D space. Data interpreted from the deep-reading measurements are located on a vertical 2D section containing the well and may be up to thirty meters above or below the wellbore. Several wellbores can be combined in a single interpretation and the results are visualized in the same 3D domain.

From all these interpretations, the workflow proposed by the existing plugin starts with a computation of isopach maps for each stratigraphic layer and an upscaling of the borehole dip data. The isopach maps are then used with the upscaled dip vectors and the stratigraphic information to project dips, well markers and horizon evidences in the surroundings of the wellbore. The obtained vectors and data points are finally used to model stratigraphic surfaces, one by one, with B-splines (i.e., parametric surfaces, Section 1.1.6, p.28).

The advantage of this workflow resides in its modeling flexibility as different thickness maps and projections can be generated by tuning the plugin's parameters. The issue is that the generated models do not always honor all the validity conditions presented in Section 1.1.5 (p.27), for instance the horizons can intersect. It hence requires a lot of expertise and manual interactions to produce a valid model, and a modification in the interpreted numerical data implies to resume this modeling

workflow from scratch. Therefore, the global study is usually performed in two steps: interpreting all the borehole data first, and then computing the structural model.

6.1.2 Implicit modeling plugin

The new plugin reduces the workload required in the second part of the study. It comes as an extension to the existing plugin and uses this thesis's framework (Section 2.2, p.79), with the Moving Least Squares discretization (Chapter 3, p.91), to model the stratigraphic surfaces.

The proposed modeling method computes the surfaces in only one step, avoiding the sequence of operations: creation of isopach maps, upscaling of borehole dips, projections of data points to horizons, and surface creation. It handles all the numerical constraints concerned by this application: point type constraints (Sections 2.2.3.1 and 2.2.5.1), vector type constraints (Sections 2.2.5.2 and 2.2.5.3), and hard data constraints (Section 2.2.5.5) (p.80 to 85). Moreover, it can handle any settings of these constraints if they fulfill the minimum requirements presented in Section 2.2.5.7. The produced models are honoring the validity conditions in a straightforward manner using the concept of implicit modeling (Section 1.2.2, p.31). Finally, creating a model takes a few seconds to a few minutes and the algorithm requires a reduced number of parameters (i.e., resolution of the nodal discretization and smoothing and data weights, the remaining being set by default following Section 3.2, p.91).

With this approach, the study does not need to be separated into two steps (interpretation and then modeling). As long-term objective, we would like to interpret the data and construct the structural model as the well is drilled. A few images of models computed and visualized with the plugin to Petrel are shown in Figure 6.1. This plugin was accepted by Schlumberger representatives and will be tested by quality analysts in the coming year for consulting.

Other implicit methods could have been proposed for this application. For instance, the Potential Field Method (Section 1.3.2, p.40) is adapted to this application when only one wellbore is concerned. In this case, the domain of study follows the well trajectory with a reduced extension perpendicularly. This enables the use of compactly supported interpolants as undefined stratigraphic volumes can be avoided even with relatively small supports. It also simplifies the way discontinuities are handled (Section 1.3.2.7, p.46): with a reduced visibility away from the wellbore, faults are considered infinite, which is convenient to define the jump functions and their supports. However, in the case where several wellbores are considered, the conditions are less advantageous: the domain must contain all the well trajectories with arbitrary directions in 3D and the faults can vanish laterally in space to explain the appearance and disappearance of stratigraphic sequences between wellbores.

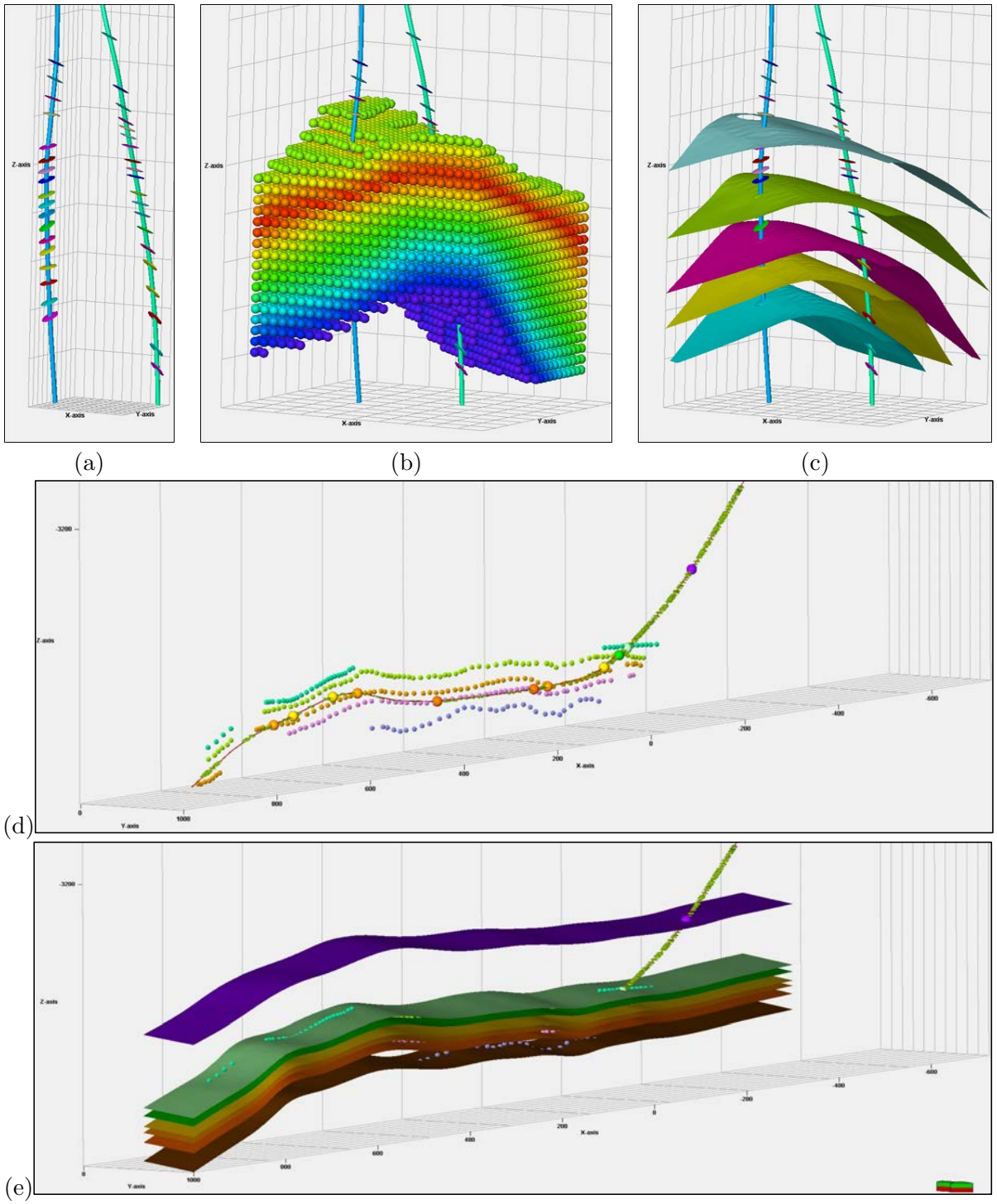


Figure 6.1: *Two 3D structural models created with borehole image interpretations and this thesis's framework in a plugin to Petrel. (a) Two vertical wellbores with oriented well markers giving the presence of horizons and their dip, (b) The computed stratigraphic function cut by a maximum and a minimum value, (c) A few extracted iso-surfaces honoring the numerical data from (a), (d) A sub-horizontal wellbore with well markers and orientation data along the well trajectory, and with points away from the well trajectory that were picked on deep-reading electromagnetic measurements, and (e) The extracted iso-surfaces honoring the numerical data from (d).*
(Data sets provided by Philippe Marza, Schlumberger)

6.2 Salt envelope surface modeling

6.2.1 Specificities of the application

A salt geobody is generally modeled by its boundary line in 2D or its boundary surface in 3D. Each salt envelope is modeled separately but may include disconnected bodies. The surface may therefore contain several pieces, which can independently be opened (and infinite) or closed. The numerical data come generally from the automatic or manual interpretation of seismic profiles, but other origins are also possible (Section 1.1.4, p.26). Therefore, these data have supposedly the same nature as in structural modeling (Section 1.1.4.5, p.26) but in each model, all the data are expected to belong to the same salt surface.

Salt envelope modeling is close to unique surface construction modeling (Section 1.5, p.54), the main difference is that only one closed surface is expected in the latter. Consequently, we believe that other existing methods in this field may be better adapted than the one presented here.

The proposed framework for implicit structural modeling is employed for the generation of salt surfaces. An implicit function is created in a domain Ω surrounding the numerical data, and only the iso-surface associated to the salt envelope is extracted (e.g., iso-value fixed to 0 by default).

We consider the case where only data points are involved, but the proposed framework can handle other types of constraints (Section 2.2.5, p.83). In this case, as all the data points are associated to the same iso-value, artificial data must be added to impose a non-zero gradient norm in the implicit function. In unique surface construction, oriented normal vectors are generally computed locally from the available point cloud and used to add artificial data, such as normal gradients (Macêdo *et al.*, 2009; Gois *et al.*, 2013), or points at the head of the vectors (Turk & O'brien, 1999; Carr *et al.*, 2001; Dinh *et al.*, 2001). However, algorithms computing these oriented vectors are complex on arbitrary surfaces (Hoppe *et al.*, 1992; Ledez, 2003) and the created artificial data have an impact on the solution. For simplicity, we propose to add data points manually away from the modeled salt surface (as in Frank *et al.* (2007)).

In the following, we use the MLS prototype presented in Chapter 3 (p.91) to construct the salt envelope surfaces with the above discussed numerical data.

6.2.2 Synthetic models

Synthetic models with regularly and accurately distributed data points are shown in Figure 6.2. They were computed with a few artificial data points: one in the center of the sphere in Figure 6.2(a), one at the bottom in Figures 6.2(b), (c), and (e), and one in each corner of the bunny's cube, one in its body, one in its head, and one in each of its ears in Figure 6.2(d). Such data are sufficient as these salt interfaces are densely sampled. The Stanford bunny (Figure 6.2(d)) needs more points as the geometry of the ears, elongated and thinner than the main body, is difficult to seize while producing a singled closed surface. However, it is still possible to compute a disconnected closed volume from the main opened surface if necessary (Figure 6.2(e)).

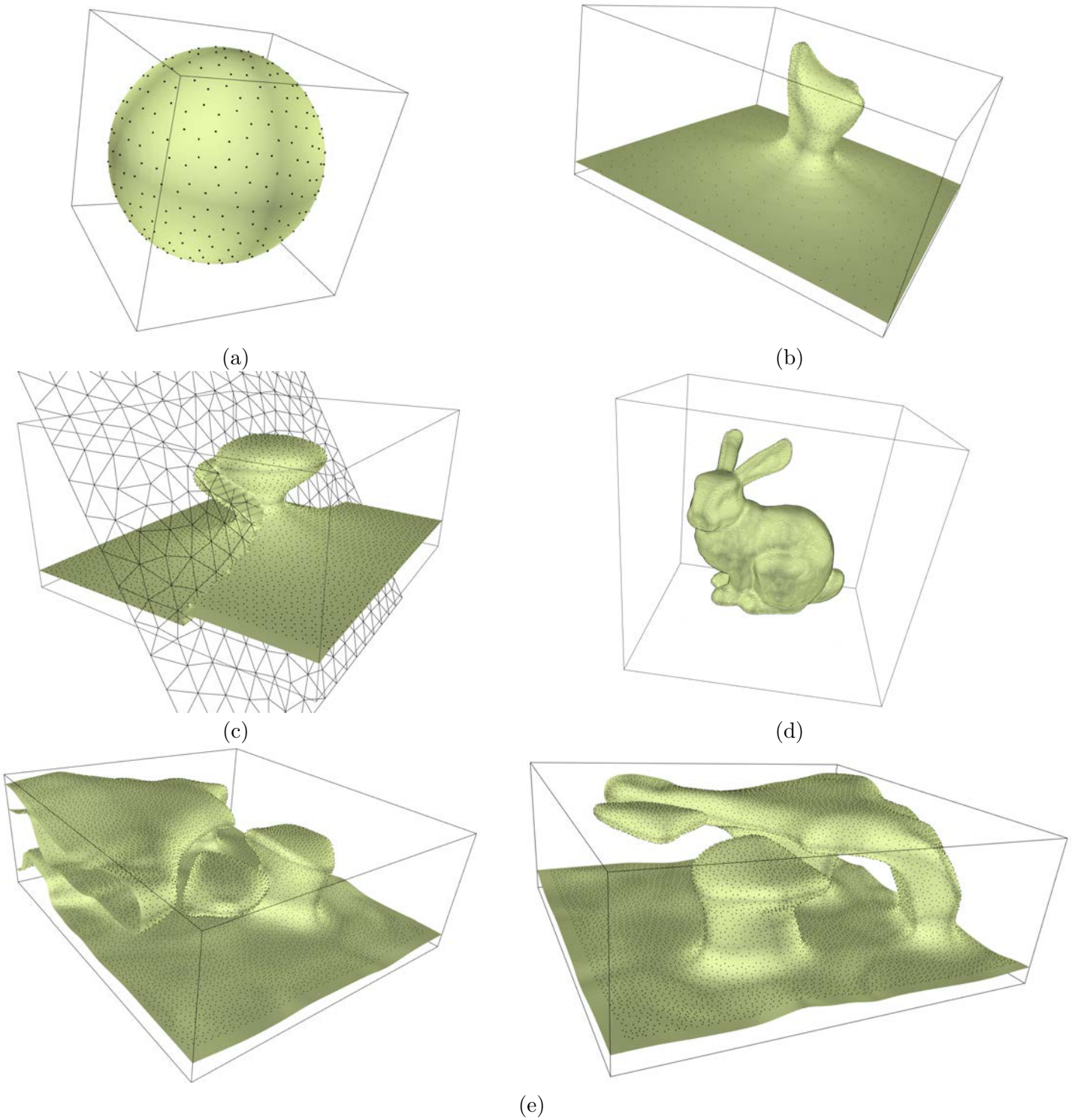


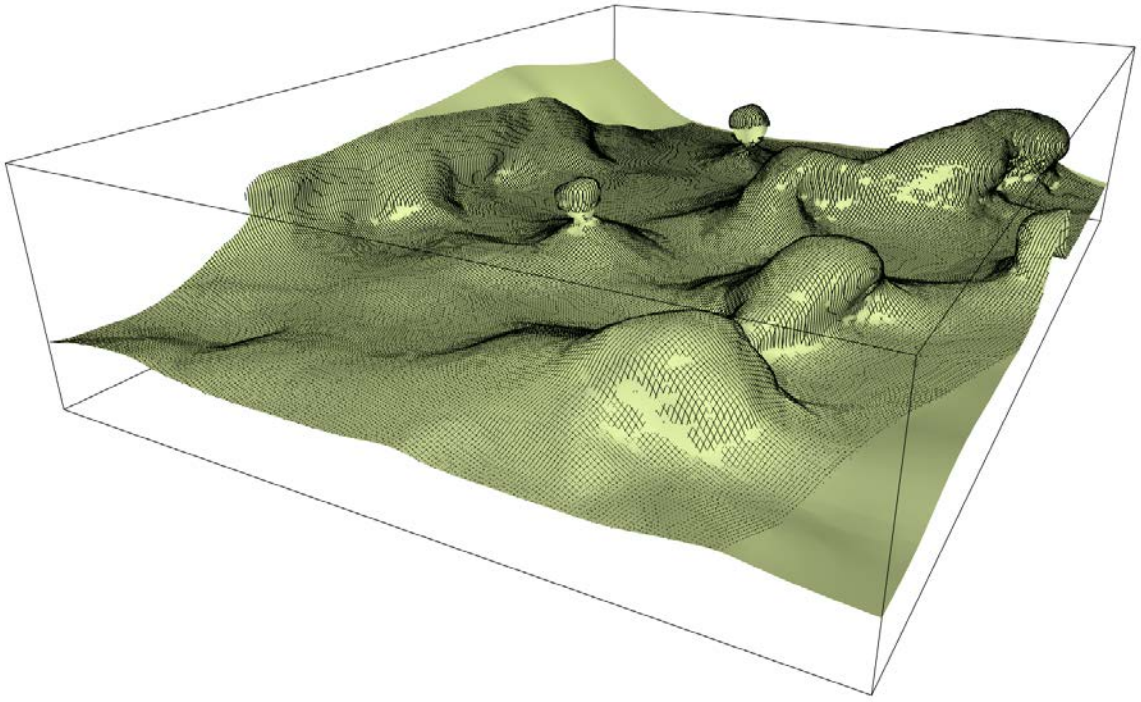
Figure 6.2: *3D synthetic models showing the potential of this thesis's framework to fit complex structures for salt envelope surface modeling. (a) A sphere, (b) A single diapir, (c) An unrealistic faulted diapir, (d) The Stanford Bunny, and (e) A complex salt structure with a diapir, an overturned canopy, and a disconnected body.*
(Data sets (a), (b), (c) and (e) provided by Laurent Maerten (Schlumberger), and (d) by Stanford University Computer Graphics Laboratory)

6.2.3 Application cases

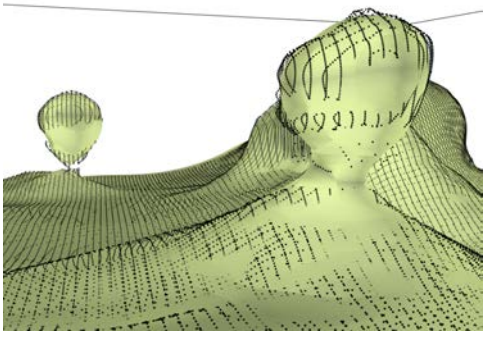
6.2.3.1 Surface reconstruction in the Santos Basin

For the study of radial faults generated near salt diapirs in the Santos Basin, offshore Brazil (Coleman *et al.*, 2018), a salt envelope surface was created based on seismic profiles, manual interpretations and using explicit surface modeling. A dense point set was then extracted from the polygonal surface. This point set was shared with us to test our modeling method and reproduce a salt surface implicitly.

The model shown in Figure 6.3(a) was created with one artificial data point at the center top of the domain. The particularity of the data set is that it is dense everywhere except at the edges of the diapirs shown in Figure 6.3(b). The data points may not be sufficient enough to decide whether each of these diapirs are isolated from the main salt surface or not. It even seems like those points were extracted from distinct sets of surfaces. The algorithm gives a possible solution, connecting only one of the two diapirs.



(a)



(b)

Figure 6.3: *Reconstruction of a 3D salt surface of the Santos Basin modeled with this thesis's framework and using a poinset extracted from an initial meshed surface. (a) Global view of the surface, and (b) Zoom on the two thin diapirs.*
(Data set provided by Alexander Coleman, Christopher A-L. Jackson (Imperial College) and Petroleum Geo-Services)

6.2.3.2 Analogue model of the Nordkapp Basin

In Marin *et al.* (2018), an analogue sandbox model of the Nordkapp Basin, Barents Sea, was created for the estimation of physical properties around salt geobodies. By modeling the salt volume and the stratigraphic layers around it, temperature and pressure perturbations were evaluated around the salt bodies using PetroMod (Schlumberger, 2019), a basin and petroleum system modeling software. The analogue model was densely interpreted (with auto-tracking tools and manual picking) on ninety four slices generated with the Image-to-SEGY method, which converts RGB values into seismic amplitudes. The salt volume was then isolated with boolean operations based on the seismic amplitude values.

The interpretation point set of the salt envelope was shared with us to build the salt surface implicitly. The model shown in Figure 6.4 was created with ten artificial data points around the salt canopy to seize its complex geometry. Figure 6.4(d) shows a zoom on the model where the noise in the data points is clearly observable. The proposed method is flexible enough to model salt structures with complex geometries while enforcing a sufficient rigidity to filter a significant noise in the data.

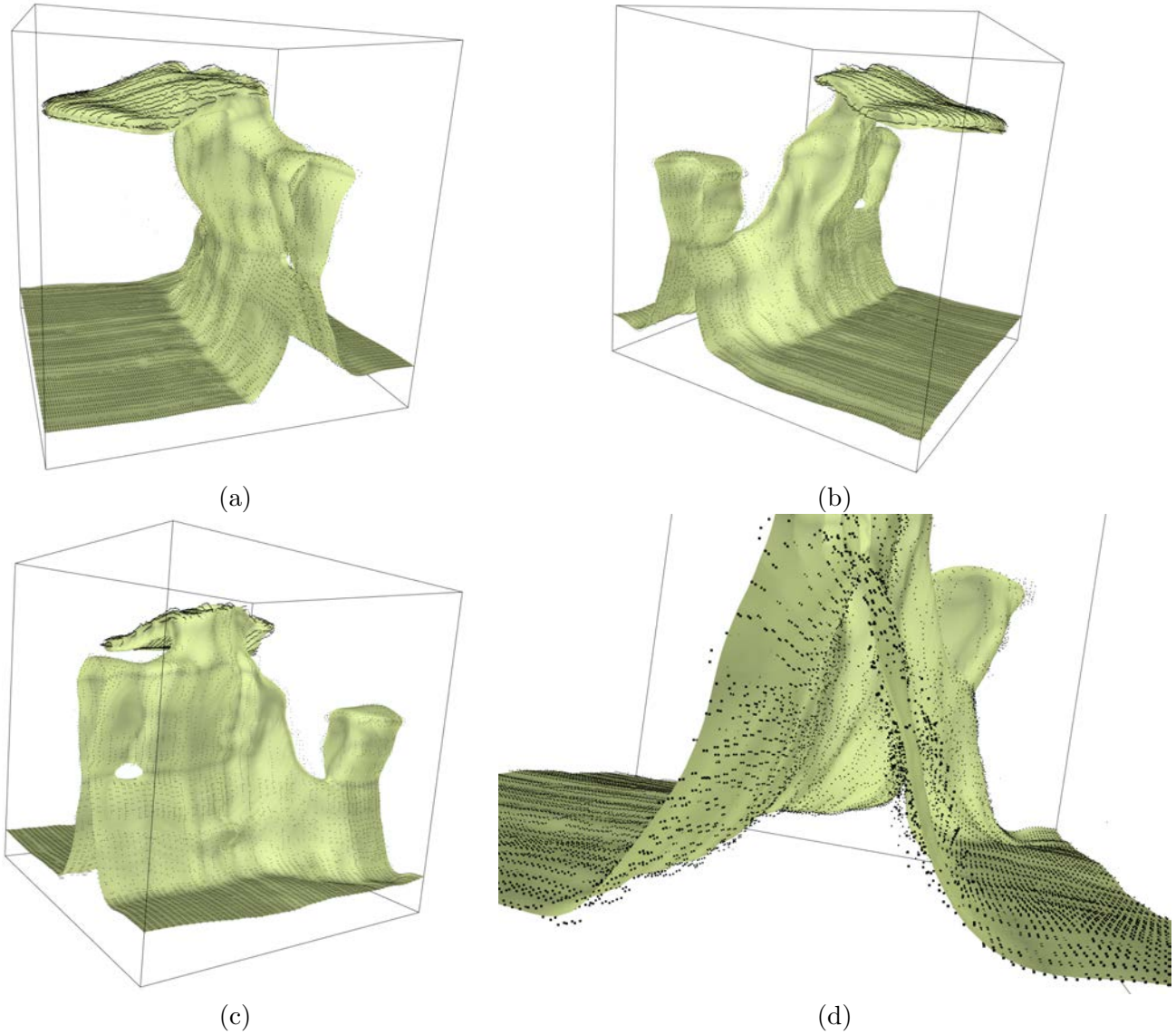


Figure 6.4: *3D salt surface constructed with this thesis's framework on a point set issued from the interpretation of parallel slides on an analogue sandbox model of the Nordkapp basin. (a), (b) and (c) Three different points of view of the constructed surface, and (d) Zoom showing the noise in the data point set.*
(Data set provided by Oriol Ferrer, University of Barcelona)

6.3 Mechanical structural restoration

The implicit structural modeling problem is posed as a weak problem in this thesis (Section 2.2.2, p.80). The discretizations employed to solve this problem may therefore be adapted to solve other weak problems. In this section, the use of Moving Least Squares shape functions is investigated for the mechanical restoration of structural models.

This work was realized by Lucille Chauveau during her internship in Schlumberger in 2018. She was mentored by Julien Renaudeau and Frantz Maerten, and Emmanuel Malvesin helped with technical details.

6.3.1 Overview of the application

Structural restoration consists in unfaulting and unfolding a model to recover the paleo-geometry of the studied structures. It can represent, for instance, a way to invalidate structural interpretations (Dahlstrom, 1969) or assess model reservoir deformations (Maerten & Maerten, 2006). Geomechanical restoration uses continuum mechanics and material constitutive laws to evaluate the displacement $\boldsymbol{u}$ of deformation during the unfaulting and unfolding. A review of numerical methods for structural mechanical restoration can be found in Chauvin (2017): Section 1.

6.3.1.1 Strong form

The mass conservation and the conservation of linear momentum are combined to define a typical partial differential equation (PDE) in solid mechanics (Liu & Gu, 2005) as

$$\boldsymbol{L}^T \cdot \boldsymbol{\sigma} + \boldsymbol{b} = \rho \frac{\partial^2 \boldsymbol{u}}{\partial t^2} + c \frac{\partial \boldsymbol{u}}{\partial t}, \tag{6.1}$$

with $\boldsymbol{L}$ the differential matrix, $\boldsymbol{\sigma}$ the stress tensor, ρ the volumtric mass density, c the damping coefficient, $\boldsymbol{b}$ the external body forces (e.g., the gravity force) and $\boldsymbol{u}$ the displacement vector.

This application is in 2D, so the vector $\boldsymbol{u}$ is defined in the x and y axes as

$$\forall \boldsymbol{x} \in \Omega, \quad \boldsymbol{u} = \begin{bmatrix} u_x(\boldsymbol{x}) \\ u_y(\boldsymbol{x}) \end{bmatrix}, \tag{6.2}$$

with Ω the domain of study gathering the layers to be restored. We consider the body forces negligible as compared to the stress (i.e., $\boldsymbol{b} \approx 0$), and the system to be at the equilibrium state at each restoration step (i.e., $\partial^2_{tt}\boldsymbol{u} = 0$ and $\partial_t \boldsymbol{u} = 0$). Locally, the stress tensor (small deformation hypothesis) is further detailed as

$$\forall \boldsymbol{x} \in \Omega, \quad \boldsymbol{\sigma} = \boldsymbol{D} \cdot \boldsymbol{\epsilon} = \frac{1}{2} \boldsymbol{D} \cdot \boldsymbol{L} \cdot \boldsymbol{u}, \tag{6.3}$$

with $\boldsymbol{\epsilon}$ the strain tensor, $\boldsymbol{D}$ the stiffness tensor defined with the Young modulus ν and the Poisson coefficient E for a 2D isotropic material as

$$\boldsymbol{D} = \frac{E}{1-\nu} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}, \tag{6.4}$$

and $\boldsymbol{L}$ defined as

$$\boldsymbol{L} = \begin{bmatrix} \partial_x & 0 \\ 0 & \partial_y \\ \partial_y & \partial_x \end{bmatrix}. \tag{6.5}$$

The system is also subject to Neumann and Dirichlet boundary conditions which may take different forms depending on the specificities of the application (e.g., free borders, flattening borders, fault contacts). The partial differential equation under general Neumann and Dirichlet constraints then corresponds to

$$\begin{cases} \boldsymbol{L}^T \cdot \boldsymbol{\sigma} = 0, & \forall \boldsymbol{x} \in \Omega \\ < \boldsymbol{\sigma}, \boldsymbol{n} >= \boldsymbol{t}, & \forall \boldsymbol{x} \in \Gamma_N \\ \boldsymbol{u} = \overline{\boldsymbol{u}}, & \forall \boldsymbol{x} \in \Gamma_D \end{cases}, \tag{6.6}$$

with $\boldsymbol{n}$ the vector normal to the boundary Γ_N, $\boldsymbol{t}$ a known traction vector, and $\overline{\boldsymbol{u}}$ a known displacement. The traction will be considered as null on all the Neumann boundaries Γ_N (i.e., free borders) in the application.

6.3.1.2 Weak form

The weak form of the strain energy corresponding to the PDE given by Equation (6.6) is written as

$$J_\sigma(\boldsymbol{u}) = \frac{1}{2}\int_\Omega (\boldsymbol{L}\cdot\boldsymbol{v})^T\cdot\boldsymbol{D}\cdot(\boldsymbol{L}\cdot\boldsymbol{u})\,d\Omega. \tag{6.7}$$

with $\boldsymbol{v}$ the test function. This is usually subjected to external forces (body forces, traction) as

$$W(\boldsymbol{u}) = \int_\Omega \boldsymbol{v}^T\cdot\boldsymbol{b}\;d\Omega + \int_{\Gamma_N}\boldsymbol{v}^T\cdot\boldsymbol{t}\;d\Gamma_N, \tag{6.8}$$

but it is approximated as null in this application. The Dirichlet conditions are then enforced by penalization, enriching the strain energy J_σ in the weak formulation:

$$J_\sigma(\boldsymbol{u}) + \frac{1}{2}\int_{\Gamma_D}\lambda\boldsymbol{v}^T\cdot(\boldsymbol{u}-\overline{\boldsymbol{u}})d\Gamma_D = 0. \tag{6.9}$$

A Bubnov-Galerkin scheme is employed, so the test and shape basis functions are the same and denoted as $\boldsymbol{\Phi}$. The Moving Least Squares functions (Section 2.1.3.4, p.67) are used to build $\boldsymbol{\Phi}$. The implicit function is then written as

$$\boldsymbol{u} = \begin{bmatrix}\boldsymbol{\Phi}(\boldsymbol{x}) & 0\\ 0 & \boldsymbol{\Phi}(\boldsymbol{x})\end{bmatrix}^T\cdot\begin{bmatrix}\boldsymbol{U}_x\\ \boldsymbol{U}_y\end{bmatrix} = \boldsymbol{B}^T\cdot\boldsymbol{U}, \tag{6.10}$$

with a nodal discretization in the domain Ω to support the MLS functions. Equation (6.9) reduces to

$$\int_\Omega \left(\;(\boldsymbol{B}\cdot\boldsymbol{L}^T)\cdot\boldsymbol{D}\cdot(\boldsymbol{B}\cdot\boldsymbol{L}^T)^T\;\right)\cdot\boldsymbol{U}d\Omega + \int_{\Gamma_D}\lambda\left(\boldsymbol{B}\cdot\boldsymbol{B}^T\right)\cdot\boldsymbol{U}\;d\Gamma_D = \int_{\Gamma_D}\lambda\boldsymbol{B}\cdot\overline{\boldsymbol{u}}\;d\Gamma_D. \tag{6.11}$$

6.3.2 Developed prototype

To properly solve Equation (6.11), the first step is to define the domain of study Ω and the structures to restore. We construct the structural model of a given data set (Figure 6.5(a)) with the presented MLS prototype (Chapter 3, p.91, giving Figure 6.5(b)). The domain Ω is then considered to be contained between the top and bottom horizons iso-surfaces (Figure 6.5(c)).

The Dirichlet boundary conditions are further detailed in Figure 6.5(c). We use a flattening constraint on the top of the domain as

$$\forall\boldsymbol{x}(x,y)\in\Gamma_D,\quad u_y(\boldsymbol{x}) = y_{ref} - y, \tag{6.12}$$

which means that the points can still have a displacement in the x axis and are not only projected vertically as illustrated in the figure. We also propose to fix one point to avoid an infinity of solutions of the problem by translation:

$$\boldsymbol{x}_0\in\Omega,\quad \boldsymbol{u}(\boldsymbol{x}_0) = 0. \tag{6.13}$$

These conditions are imposed by Lagrange multipliers as described in Section 2.2.5.5 (p.85). For simplicity, we propose to discretize these conditions on the data points used to construct the top horizon (Figure 6.5(c)). In this case, the developments of Appendix G (p.171) can be followed step by step to enforce the Dirichlet conditions.

Finally, the interpolation nodes are discretized regularly in Ω. The nodal resolution used for stratigraphic modeling is preserved but only the nodes contained in Ω are kept and used to construct

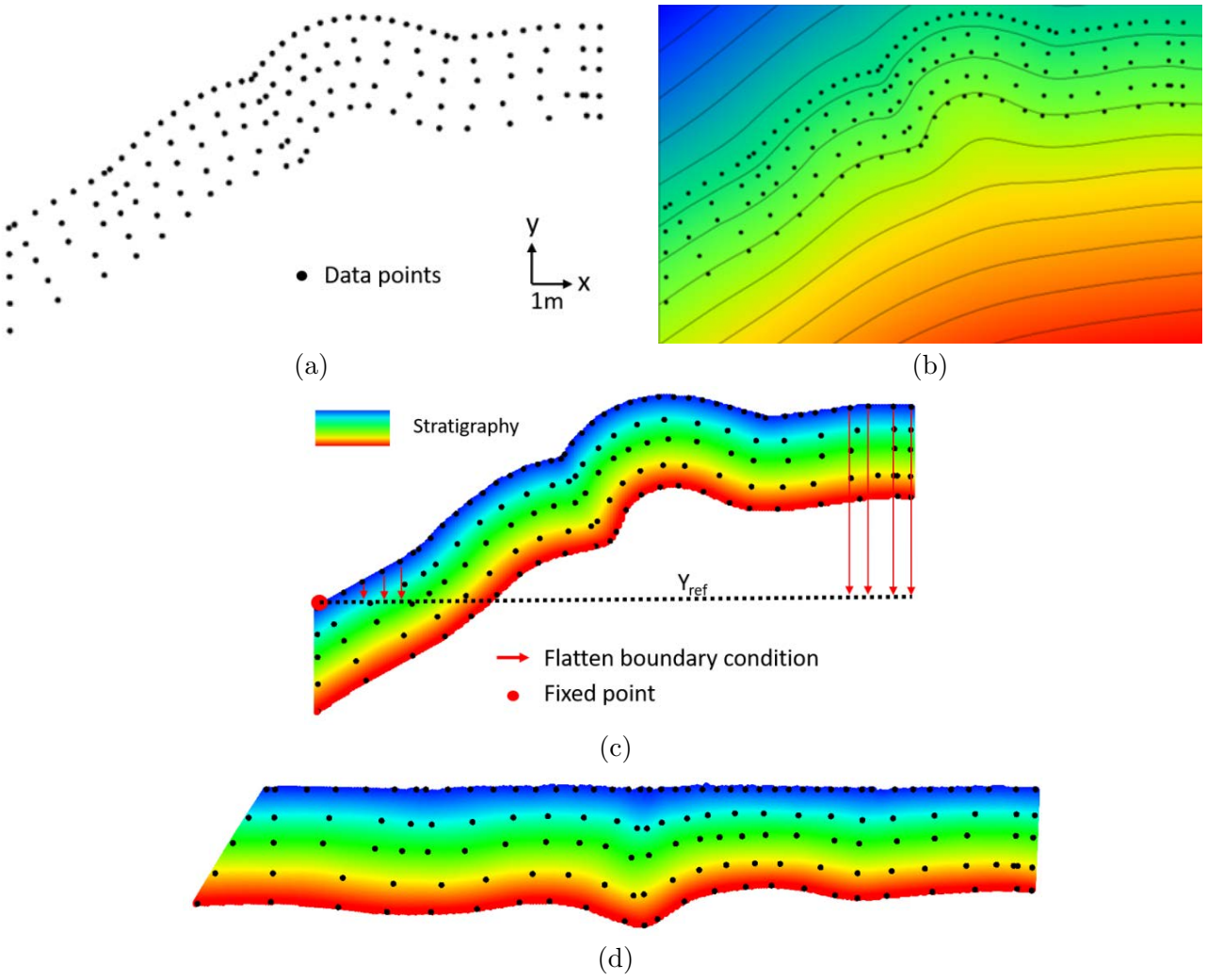


Figure 6.5: *Geomechanical restoration workflow using Moving Least Squares shape functions for a folded domain in 2D. (a) The initial data points, (b) The stratigraphic model constructed with the structural modeling prototype using MLS functions, (c) The domain of study for the first restoration step and the applied boundary conditions, and (d) The first restored state.*

the shape functions. The domain is approximated by a grid following the structures (i.e., a grid with stair-steps) and the strain energy is considered constant in each of the grid's cells and evaluated at their center point (i.e., quadrature with one point performed on each discretization node, as in Appendix E, p.169).

After solving Equation (6.11), the displacement $\boldsymbol{u}$ is applied to each data and visualization point to model the restored structures (Figure 6.5(d)). A restored state was computed on the same model with Dynel2D, a mechanics-based restoration software using the Finite Element Method (Maerten (2010): Section 10). The two results seem visually close but no comparison study was performed, so we cannot yet conclude on this matter. Only one restoration step is performed here. Proceeding with the next step would require to erase the top layer and the top horizon data points, and start over with the workflow.

The unfaulting was also investigated with an additional point-to-point constraint to remove the throw at the top of the fault, and additional contact constraints between the two fault blocks. The point-to-point constraint was enforced as a Dirichlet constraint as done for the flattening. The contact constraints were implemented with an iterative algorithm minimizing the distance between the nodes closest to the fault. Unfortunately, this approach did not lead to satisfactory results (yet) and need to be further investigated in the future.

General conclusions

In this thesis, we focus on developing geomodeling tools to assist the creation of numerical models representing the geological structures. A special intent is to suggest mathematically robust solutions for implicit structural modeling. The contributions of the thesis are here summarized and discussed for future developments.

A continuous framework for implicit structural modeling

In Chapter 2 (p.59), we suggest posing the problem of implicit structural modeling with continuous equations. This approach enables a formulation of the problem which is not dependent on the discretization. It enables new ideas concerning the modification of the modeling problem and its discretization, and it establishes links between the existing and the newly created methods.

A generic version of the problem is posed as a minimization of a sum of functionals. We suggest to use discrete functionals for data constraints and continuous functionals for regularization. A standard version of this framework is given as a spatial regression of data points penalized by the bending energy. A complete workflow to discretize the given equations is also presented with a review of adapted numerical methods.

As discussed in Appendix B, it is still unclear to the author whether the proposed minimization problem is equivalent to solving a Partial Differential Equation under constraints. Further studies are needed on this matter to help linking, for instance, the proposed framework to other methods solving the biharmonic equation. In particular, it would be interesting to better understand the differences between the bending energy and the Laplacian energy which produce different results although they solve the same equation (Chapter 5, p.135). For this, a first approach would be to impose explicitly the boundary conditions given in Briggs (1974) and the ones discussed in the appendix.

Two discretizations of the continuous framework are implemented. Other methods can be created using other shape and test functions as presented in the numerical review. For instance, the Partition of Unity Method seems attractive for its flexibility: as any type of intrinsic basis functions can be used, it may be possible to adapt the interpolation functions per partitions to relate specific local features. For instance, this principle is used in Martin & Boisvert (2017), but other interpolants than Radial Basis Functions could be mixed in the method. However, a partitioning of the domain adapted to implicit structural modeling still needs to be defined with this approach.

An implicit structural modeling method using Moving Least Squares functions

A first example of discretization of the continuous framework is presented in Chapter 3 (p.91) using the locally defined Moving Least Squares (MLS) functions (McLain, 1976; Lancaster & Salkauskas, 1981). These functions have already been used in geomodeling to construct fault surfaces (BaDughaish, 2013), ore volumes (Manchuk & Deutsch, 2019), and structural models (Maerten, 2018). The particularity of our approach is that the functions are centered on nodes regularly sampled in the domain of study. The visibility criterion (Belytschko *et al.*, 1994b) is then used to handle faults and unconformities.

The regular sampling allows us to define default parameters for the MLS functions to ensure the stability of the interpolation in most geological settings. In complex fault networks, a local resampling based on a minimum number of neighboring nodes to construct the MLS functions is proposed. With

this, any geological setting is theoretically handled by the proposed method, but the approach still holds some limits.

The first issue is that the visibility criterion produces discrepancies in the results close to the fault tips when relatively big supports of interpolation are used. As we scale the length of the supports on the nodal sampling, even though the interpolation is stabilized locally with the generation of additional nodes, the results may involve non-negligible artifacts at the fault tips when using a coarse nodal sampling. To reduce these artifacts, we propose to use the transparency method (Organ *et al.*, 1996) instead of the visibility criterion. A modified version of this method is also suggested to control the dependency of the modeled structures between fault blocks. Unfortunately, the related parameters must be tuned empirically, so further sensitivity analyses are required to suggest normalized values.

The second issue is that the local generation of nodes is random, so even if a sufficient number of neighbors to construct the MLS functions is obtained, this does not ensure their stability. Other local sampling strategies should be investigated, such as using repulsion factors or density maps to generate nodes evenly around a given position.

Other opportunities of optimization of the code are discussed. These mainly involve: (1) the approximation of the MLS functions by normalized weight functions depending on the distribution of the surrounding nodes, (2) a better management of intersection tests performed between nodes and fault surface triangles, (3) the parallelization of the code, and (4) the use of different sampling resolutions (Müller *et al.*, 2004; Slak & KOSEC, 2019) with an adaptive dilatation parameter. Complex fault geometries could also be better handled using the concept of ghost nodes.

An implicit structural modeling method using the concept of ghost nodes

A second example of discretization of the continuous framework is presented in Chapter 4 (p.119). It uses shape functions constructed on a Cartesian grid to perform the interpolation. The continuous equations, which involve second order derivatives, are approximated with finite differences. Also, the concept of ghost nodes (or Extended Finite Element Method, Moës & Belytschko (2002)) is introduced to handle faults and unconformities. The schemes of finite differences are then adapted and written on the ghost nodes as well.

The ghost nodes are presented as an efficient way to introduce a discontinuous jump in a mesh based interpolation without using a conformal mesh to the fault surfaces. It is used on a Cartesian grid for simplicity, but it may be employed on any other mesh, and even on nodal based meshless functions. Like the visibility criterion, it produces artifacts in the interpolation at fault tips and ensures the independency between fault blocks. Further techniques to deal with these issues should be investigated, in particular in the existing literature on Extended Finite Element Method and associated (Babuska & Melenk, 1995; Melenk & Babuška, 1996; Belytschko *et al.*, 2001; Moës & Belytschko, 2002).

This Cartesian grid method is proposed as an alternative to the MLS prototype to solve the same modeling problem. A comparison study between these two approaches is performed on a 2D synthetic model. The two methods converge to approximately the same solution when refining the two discretizations. Therefore, the main difference between the two proposed methods resides in the approximation of the modeling problem made by their respective discretization. This emphasizes that the proposed modeling problem is independent from the discretization.

In this comparison study, the present method is also proven to be, at a given resolution, computationally more efficient than the MLS discretization, but to produce less continuous and accurate results. Therefore, a comparison study on the computational efficiency of these methods could be led at a fixed accuracy instead of a fixed resolution, as proposed in Liu & Gu (2005).

Improvements on the Cartesian grid method with ghost nodes may involve the use of tree structures to have an adaptive resolution of the mesh. Several tree structures may be considered: e.g., kd-trees, quadtrees (and octrees), and binary triangle trees (and binary tetrahedron trees, Duchaineau *et al.* (1997)). In each of those cases, specific strategies must be developed in order to ensure the creation of bases functions at least C^0 on the domain of study (Tabarraei & Sukumar, 2007), and a stable approximation of the continuous equations (Chen *et al.*, 2007).

A criterion to handle thickness variation issues: the weighted curvature minimization

A modification of the standard framework is suggested in Chapter 5 (p.135) to handle thickness variation issues. It uses the physical meaning of the bending energy to modify the intensity of smoothing in space. The smoothing energy is enforced with a weak penalization weight close to the numerical data, and a great penalization weight away from the data. This results in a stratigraphic function which accommodates smoothly the variations of thickness within the layers. The technique is tested on synthetic models which have no meaning geologically, but they demonstrate the capability of the technique to handle extreme cases.

Although we managed to find a setting of parameters that avoid stratigraphic inconsistencies in all the tested models of thickness variation issue, this approach does not ensure that such a setting can always be found for any other models. Therefore, other constraints to strictly forbid the creation of such inconsistencies should be investigated. For instance, the closed iso-surfaces avoided with the WCM criterion are associated to one or several positions where the gradient of the implicit function is null. Defining inequality constraints on the gradient's norm may represent a way to enforce a monotonous implicit function, and therefore avoid the observed inconsistencies.

A method to impose a local anisotropy on the interpolation

Other modifications of the standard framework are suggested in Chapter 5 to relate an observed anisotropy of the structures directly into the regularization. This is either done by enriching the bending energy with an anisotropic functional, or by replacing directly the bending energy with an anisotropic energy. In the MLS and the Cartesian grid prototypes, the regularization is discretized on the entire domain of study with local constraints. Therefore, an anisotropic functional can efficiently impose a local anisotropy on the interpolation.

The enrichment technique is tested on 3D synthetic models of folds where data are missing in high curvature areas. In these tests, the fold axes are enforced as directions of increased rigidity, maintaining the hinge of the folds between cross sections. The anisotropic energy is tested on the 2D model used for the weighted curvature minimization. This test shows that the anisotropic energy can also be employed to handle thickness variation issues in the layers.

This last contribution represents the most immature part of the thesis. Therefore, several necessary developments are missing. For instance, we need to evaluate the relations between the parameters of the proposed functionals and the quality of the results in different applied settings to assess the pertinence of the anisotropic terms. Based on this sensitivity analysis, some algorithms to compute those parameters locally and automatically should be investigated, as in Martin & Boisvert (2017). Also, the anisotropic energy should be written in 3D by adapting the equations presented in Appendix J.

Global limits of the presented results and perspectives

Application to real case studies

The framework and the numerical methods developed in this thesis are tested on synthetic cases. The Moving Least Squares approach is also tested on three real case studies: one using borehole interpretations (Figures 6.1(d) and (e), p.153), and two modeling salt envelope surfaces (Figures 6.3, p.156 and 6.4, p.157). The presented 3D models address most of the difficulties encountered in real case studies (i.e., noisy data points, Figure 6.4, unconformity surface, Figures 3.7(a),(c) and (e), p.99 fault connectivities with small angles, Figures 3.7(b),(d),(f) and 4.12 (p.132), and thin layers, Figures 6.1(d) and (e)). However, it would be interesting to test the developed techniques on studies which involve these issues all together.

In the same idea, the developed techniques to handle thickness variations and impose anisotropy on the modeled structures are not tested on applied cases. Using real data would, for instance, allow us to evaluate the pertinence of the obtained results. In opposition to the sensitivity analysis performed

on the MLS prototype, the difficulty here would be to define a metric to evaluate the quality of the results: what is the reference model in real case studies?

The faults are not part of the discretization

In the MLS and the Cartesian grid prototypes, faults and other discontinuities are not part of the discretization of the modeling problem. This represents both an advantage and an inconvenient to model geological structures. The advantage resides in the simplicity with which the discontinuities are introduced into the implicit function. It mainly involves intersection tests between segments in 2D and between segments and triangles in 3D. The inconvenient, as in the Potential Field Method, is that the created models are visualized on Cartesian grids for simplicity and that the used methods to extract the iso-surfaces do not create sealed models. Although truncation and projection techniques may help at creating valid models, it seems necessary to investigate other techniques of extraction and visualization (Gomes *et al.*, 2009; De Araújo *et al.*, 2015) and/or techniques of model repair and model simplification (Anquez *et al.*, 2017, 2019).

We also present a discretization of the mechanical restoration problem using the Moving Least Squares functions in Chapter 6 (p.151). The developed prototype handles the unfolding of 2D models using flattening constraints and many further developments remain. In particular, the unfaulting is necessary but challenging: the contact constraints between the two sides of a fault are traditionally written on the duplicated fault elements (Chauvin, 2017). As the fault elements are not part of the discretization, the contact constraints should be written differently, or the fault description should be adapted to the discretization.

Appendices

A Moving least squares demonstration

This section gives the construction details of the moving least squares (MLS) functions (McLain, 1976; Lancaster & Salkauskas, 1981) defined in Equation (3.3) (p.92). We use the demonstration from Nguyen *et al.* (2008) and clarify the approximation of the coefficients $\boldsymbol{a}(\boldsymbol{p})$ around a fixed position $\boldsymbol{p}$.

The implicit function u is redefined as a polynomial function of degree $order(\mathcal{P})$ with spatially varying coefficients $\boldsymbol{a}(\boldsymbol{x})$ as

$$\forall \boldsymbol{x} \in \Omega, \quad u(\boldsymbol{x}) = \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}} \Phi_{\overline{\boldsymbol{p}}}(\boldsymbol{x})\ u_{\overline{\boldsymbol{p}}} = \sum_{m \in \mathcal{P}} m(\boldsymbol{x})\ a_m(\boldsymbol{x}) = \mathcal{P}^T(\boldsymbol{x}) \cdot \boldsymbol{a}(\boldsymbol{x}), \tag{A.1}$$

with $\boldsymbol{N}$ the set of nodes, $\Phi_{\overline{\boldsymbol{p}}}$ the MLS function associated to the node $\overline{\boldsymbol{p}}$, $u_{\overline{\boldsymbol{p}}}$ its corresponding unknown, m a monomial in the polynomial basis $\mathcal{P}$, and a_m its associated coefficient. As all the coefficients a_m vary continuously in space, they must be determined for any position $\boldsymbol{x}$.

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

$$\forall (\boldsymbol{x}, \boldsymbol{p}) \in \Omega, \quad u(\boldsymbol{x}) \approx \mathcal{P}^T(\boldsymbol{x}) \cdot \boldsymbol{a}(\boldsymbol{p}). \tag{A.2}$$

Equation (A.2) can be written on an interpolation node $\overline{\boldsymbol{p}}$ as

$$\forall \overline{\boldsymbol{p}} \in \boldsymbol{N},\ \forall \boldsymbol{p} \in \Omega, \quad u(\overline{\boldsymbol{p}}) \approx \mathcal{P}^T(\overline{\boldsymbol{p}}) \cdot \boldsymbol{a}(\boldsymbol{p}), \tag{A.3}$$

where the implicit function $u(\overline{\boldsymbol{p}})$ is supposed to be as close as possible to the nodal value $u_{\overline{\boldsymbol{p}}}$ as

$$u(\overline{\boldsymbol{p}}) \approx u_{\overline{\boldsymbol{p}}}. \tag{A.4}$$

Equations (A.3) and (A.4) are combined for each node $\overline{\boldsymbol{p}} \in \boldsymbol{N}$ separately, defining the following system of $\#\boldsymbol{N}$ equations and $\#\mathcal{P}$ unknown,

$$\left\{ \begin{array}{ccccc} \mathcal{P}(\overline{\boldsymbol{p}}_1)^T & \cdot & \boldsymbol{a}(\boldsymbol{p}) & \approx & u_1 \\ \mathcal{P}(\overline{\boldsymbol{p}}_2)^T & \cdot & \boldsymbol{a}(\boldsymbol{p}) & \approx & u_2 \\ \vdots & & & & \\ \mathcal{P}(\overline{\boldsymbol{p}}_{\#\boldsymbol{N}})^T & \cdot & \boldsymbol{a}(\boldsymbol{p}) & \approx & u_{\#\boldsymbol{N}} \end{array} \right. . \tag{A.5}$$

The pertinence of each approximation in system (A.5) is related to the distance between each $\overline{\boldsymbol{p}}$ and $\boldsymbol{p}$ respectively: as the coefficients $a_m(\boldsymbol{x})$ vary continuously in space, the further $\overline{\boldsymbol{p}}$ is to $\boldsymbol{p}$, the less reliable is the approximation. Each approximation is thus weighted by a continuous function $w_{\overline{\boldsymbol{p}}}(\boldsymbol{p})$, centered on each position $\overline{\boldsymbol{p}} \in \boldsymbol{N}$ and becoming nil when $\overline{\boldsymbol{p}}$ is far from $\boldsymbol{p}$. Determining the coefficients $a_m(\boldsymbol{p})$ is thus equivalent to minimizing, in the least squares sense, the functional $J_{MLS}(\boldsymbol{a}(\boldsymbol{p}))$,

$$J_{MLS}(\boldsymbol{a}(\boldsymbol{p})) = \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}(\boldsymbol{p})} w_{\overline{\boldsymbol{p}}}(\boldsymbol{p}) \left(\mathcal{P}^T(\overline{\boldsymbol{p}}) \cdot \boldsymbol{a}(\boldsymbol{p}) - u_{\overline{\boldsymbol{p}}} \right)^2, \tag{A.6}$$

with $\boldsymbol{N}(\boldsymbol{p})$ the set of nodes close enough to $\boldsymbol{p}$ to have a non-zero weight function $w_{\overline{\boldsymbol{p}}}(\boldsymbol{p})$ (i.e., $\#\boldsymbol{N}(\boldsymbol{p}) <= \#\boldsymbol{N}$). If the number of neighboring nodes $\#\boldsymbol{N}(\boldsymbol{p})$ is smaller than the number of unknowns $\#\mathcal{P}$, the

minimization of Equation (A.6) has no solution. Minimizing J_{MLS} preserves Equation (A.4) as an approximation, which is why MLS functions do not have the Kronecker delta property (Equation (2.18), p.64).

An extremum of this functional is obtained by setting the derivative of J_{MLS} with respect to $\boldsymbol{a}(\boldsymbol{p})$ equal to zero, giving

$$\begin{cases} \sum\limits_{\overline{\boldsymbol{p}}\in\boldsymbol{N}(\boldsymbol{p})} w_{\overline{\boldsymbol{p}}}(\boldsymbol{p})2m_1(\overline{\boldsymbol{p}}) & \left(\mathcal{P}^T(\overline{\boldsymbol{p}})\cdot\boldsymbol{a}(\boldsymbol{p})-u_{\overline{\boldsymbol{p}}}\right) = 0 \\ \sum\limits_{\overline{\boldsymbol{p}}\in\boldsymbol{N}(\boldsymbol{p})} w_{\overline{\boldsymbol{p}}}(\boldsymbol{p})2m_2(\overline{\boldsymbol{p}}) & \left(\mathcal{P}^T(\overline{\boldsymbol{p}})\cdot\boldsymbol{a}(\boldsymbol{p})-u_{\overline{\boldsymbol{p}}}\right) = 0 \\ \vdots & \\ \sum\limits_{\overline{\boldsymbol{p}}\in\boldsymbol{N}(\boldsymbol{p})} w_{\overline{\boldsymbol{p}}}(\boldsymbol{p})2m_{\#\mathcal{P}}(\overline{\boldsymbol{p}}) & \left(\mathcal{P}^T(\overline{\boldsymbol{p}})\cdot\boldsymbol{a}(\boldsymbol{p})-u_{\overline{\boldsymbol{p}}}\right) = 0 \end{cases}, \tag{A.7}$$

which can be written, in matrix form, as

$$\sum_{\overline{\boldsymbol{p}}\in\boldsymbol{N}(\boldsymbol{p})} w_{\overline{\boldsymbol{p}}}(\boldsymbol{p})\mathcal{P}(\overline{\boldsymbol{p}})\cdot\mathcal{P}^T(\overline{\boldsymbol{p}})\cdot\boldsymbol{a}(\boldsymbol{p}) = \sum_{\overline{\boldsymbol{p}}\in\boldsymbol{N}(\boldsymbol{p})} w_{\overline{\boldsymbol{p}}}(\boldsymbol{p})\mathcal{P}(\overline{\boldsymbol{p}})u_{\overline{\boldsymbol{p}}}, \tag{A.8}$$

or, more compactly as

$$\boldsymbol{A}(\boldsymbol{p})\cdot\boldsymbol{a}(\boldsymbol{p}) = \boldsymbol{B}(\boldsymbol{p})\cdot\boldsymbol{U}, \tag{A.9}$$

with

$$\boldsymbol{A}(\boldsymbol{p}) = \sum_{\overline{\boldsymbol{p}}\in\boldsymbol{N}(\boldsymbol{p})} w_{\overline{\boldsymbol{p}}}(\boldsymbol{p})\mathcal{P}(\overline{\boldsymbol{p}})\cdot\mathcal{P}^T(\overline{\boldsymbol{p}}) \tag{A.10}$$

and with

$$\boldsymbol{B}(\boldsymbol{p}) = \begin{bmatrix} w_1(\boldsymbol{p})\mathcal{P}(\overline{\boldsymbol{p}}_1) \\ \cdots \\ w_{\#\boldsymbol{N}(\boldsymbol{p})}(\boldsymbol{p})\mathcal{P}(\overline{\boldsymbol{p}}_{\#\boldsymbol{N}(\boldsymbol{p})}) \end{bmatrix}. \tag{A.11}$$

Solving the least squares system defined by Equation (A.9) recovers the coefficients $a_m(\boldsymbol{p})$ for a given position $\boldsymbol{p}$. As $\boldsymbol{p}$ can be fixed anywhere in the domain Ω, the coefficients $\boldsymbol{a}(\boldsymbol{x})$ can be determined at any position $\boldsymbol{x}$ in Ω by

$$\boldsymbol{A}(\boldsymbol{x})\cdot\boldsymbol{a}(\boldsymbol{x}) = \boldsymbol{B}(\boldsymbol{x})\cdot\boldsymbol{U}. \tag{A.12}$$

Incorporating Equation (A.12) in Equation (A.1) leads to

$$u(\boldsymbol{x}) = \mathcal{P}^T(\boldsymbol{x})\cdot[\boldsymbol{A}(\boldsymbol{x})]^{-1}\cdot\boldsymbol{B}(\boldsymbol{x})\cdot\boldsymbol{U}, \tag{A.13}$$

with

$$u(\boldsymbol{x}) = \boldsymbol{\Phi}^T(\boldsymbol{x})\cdot\boldsymbol{U}. \tag{A.14}$$

Hence the Equation (3.3) (p.92),

$$\Phi_{\overline{\boldsymbol{p}}}(\boldsymbol{x}) = w_{\overline{\boldsymbol{p}}}(\boldsymbol{x})\mathcal{P}^T(\boldsymbol{x})\cdot[\boldsymbol{A}(\boldsymbol{x})]^{-1}\cdot\mathcal{P}(\overline{\boldsymbol{p}}). \tag{A.15}$$

B Discussion on the biharmonic equation and the bending energy in structural modeling

The relation between the bending energy and the biharmonic equation can be found with the weak formulation. The demonstration starts with the weak form of the bending energy given in Appendix C as

$$\int_\Omega \left(\partial^2_{ij}u\ \partial^2_{ij}v\right) d\Omega, \tag{B.1}$$

with v the test function.

By performing two integrations by parts on Equation (B.1), a weak formulation of the biharmonic equation is obtained:

- 1^{st} integration by parts

$$\int_\Omega \left(\partial^2_{ij}u\ \partial^2_{ij}v\right) d\Omega = \int_{\partial\Omega} \left(\partial_j v\ \partial^2_{ij}u\ n_i\right) d\partial\Omega - \int_\Omega \left(\partial_j v\ \partial^3_{iij}u\right) d\Omega, \tag{B.2}$$

- 2^{nd} integration by parts

$$\int_\Omega \left(\partial^2_{ij}u\ \partial^2_{ij}v\right) d\Omega = \int_{\partial\Omega} \left(\partial_j v\ \partial^2_{ij}u\ n_i\right) d\partial\Omega \quad - \int_{\partial\Omega} \left(v\ \partial^3_{iij}u\ n_j\right) d\partial\Omega + \int_\Omega \left(v\ \partial^4_{jiij}u\right) d\Omega, \tag{B.3}$$

$$\int_\Omega \left(v\ \partial^4_{iijj}u\right) d\Omega = -\int_{\partial\Omega} \left(\partial_j v\ \partial^2_{ij}u\ n_i\right) d\partial\Omega \quad + \int_{\partial\Omega} \left(v\ \partial^3_{iij}u\ n_j\right) d\partial\Omega + \int_\Omega \left(\partial^2_{ij}u\ \partial^2_{ij}v\right) d\Omega, \tag{B.4}$$

with $\boldsymbol{n}$ the normal vector of the boundary surfaces $\partial\Omega$. As faults and stratigraphic unconformities are treated as borders in the proposed framework (Section 2.2.6, p.87), they also belong to the boundary surfaces $\partial\Omega$.

The bending energy is indeed related to the biharmonic equation ($\partial^4_{iijj} \equiv \Delta^2$ is the biharmonic operator). The problem is that in structural modeling, there is *a priori* no boundary conditions (BCs). Therefore, we cannot conclude on an equivalence with a problem posed as a PDE under BC as the two boundary integrals are not supposed to be minimized. We did not try to explicitly minimize those terms in practice as they do not involve standard BCs for the FEM literature, but this should be tested in the future.

In Briggs (1974), the biharmonic equation is proposed for the contour mapping problem. The weak formulation is developed into the Laplacian energy, performing two different integrations by parts than presented here. This leads to BCs written as

$$\forall \boldsymbol{x} \in \partial\Omega,\ \partial^3_{iij}u(\boldsymbol{x})\ n_j = 0, \tag{B.5}$$

and

$$\forall \boldsymbol{x} \in \partial\Omega,\ \partial^2_{ii}u(\boldsymbol{x})\ n_j = 0. \tag{B.6}$$

If such BCs were to be considered legitimate, the minimization of the bending energy could be considered equivalent to solving the biharmonic equation under BCs written as Equation (B.5) and

$$\forall \boldsymbol{x} \in \partial\Omega,\ \partial^2_{ij}u(\boldsymbol{x})\ n_i = 0, \tag{B.7}$$

instead of Equation (B.6). As such BCs are not standard in the FEM litterature, we feel that a proper mathematical background is lacking here. Also, this boundary problem is inconvenient to suggest other PDEs, so we prefer to consider that the equivalence is not proven, and we hence present our framework with the energy approach.

C Euler-Lagrange equations on the minimization problem

Euler-Lagrange equations

The Euler-Lagrange equations enforce the first variation of the functional J to be nil (i.e., $\forall \delta u,\ \delta J(u) = 0$, with δJ and δu the variations of J and u). Let v be an arbitrary test function that we suppose sufficiently regular in Ω. Let g be the function

$$\forall \boldsymbol{x} \in \Omega, \quad g(\boldsymbol{x}) = u(\boldsymbol{x}) + t\ v(\boldsymbol{x}), \tag{C.1}$$

where $t \in \mathbb{R}$ is a constant.

Verifying that u minimizes J in all directions of derivation is given by

$$\partial_t J(g)|_{t=0} = \partial_t J_\epsilon(g)|_{t=0} + \partial_t J_{DPV}(g)|_{t=0} = 0. \tag{C.2}$$

Development of the data functional

$$\begin{aligned} J_{DPV}(g) &= \frac{1}{2}\sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ (g(\boldsymbol{p})-\alpha_{\boldsymbol{p}})^2 = \frac{1}{2}\sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ (u(\boldsymbol{p})+t\ v(\boldsymbol{p})-\alpha_{\boldsymbol{p}})^2 \\ &= \frac{1}{2}\sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ \left(u(\boldsymbol{p})^2+(t\ v(\boldsymbol{p}))^2+\alpha_{\boldsymbol{p}}^2+2t\ u(\boldsymbol{p})v(\boldsymbol{p})-2u(\boldsymbol{p})\ \alpha_{\boldsymbol{p}}-2t\ v(\boldsymbol{p})\ \alpha_{\boldsymbol{p}}\right) \end{aligned} \tag{C.3}$$

$$\partial_t J_{DPV}(g) = \frac{1}{2}\sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ \left(2t\ v(\boldsymbol{p})^2+2u(\boldsymbol{p})v(\boldsymbol{p})-2v(\boldsymbol{p})\ \alpha_{\boldsymbol{p}}\right) \tag{C.4}$$

$$\begin{aligned} \partial_t J_{DPV}(g)|_{t=0} &= \frac{1}{2}\sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ (2u(\boldsymbol{p})v(\boldsymbol{p})-2v(\boldsymbol{p})\ \alpha_{\boldsymbol{p}}) \\ &= \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ (u(\boldsymbol{p})-\alpha_{\boldsymbol{p}})\ v(\boldsymbol{p}) \end{aligned} \tag{C.5}$$

Development of the energy functional

$$J_\epsilon(g) = \frac{1}{2}\int_\Omega \lambda_\epsilon^2 \left(\partial_{ij}^2 u + t\ \partial_{ij}^2 v\right)^2 d\Omega = \frac{1}{2}\int_\Omega \lambda_\epsilon^2 \left(\partial_{ij}^2 u\right)^2 d\Omega + \frac{t^2}{2}\int_\Omega \lambda_\epsilon^2 \left(\partial_{ij}^2 v\right)^2 d\Omega + t\int_\Omega \lambda_\epsilon^2 \left(\partial_{ij}^2 u\ \partial_{ij}^2 v\right) d\Omega \tag{C.6}$$

$$\partial_t J_\epsilon(g) = t\int_\Omega \lambda_\epsilon^2 \left(\partial_{ij}^2 v\right)^2 d\Omega + \int_\Omega \lambda_\epsilon^2 \left(\partial_{ij}^2 u\ \partial_{ij}^2 v\right) d\Omega \tag{C.7}$$

$$\partial_t J_\epsilon(g)|_{t=0} = \int_\Omega \lambda_\epsilon^2 \left(\partial_{ij}^2 u\ \partial_{ij}^2 v\right) d\Omega \tag{C.8}$$

Development of the minimization problem

Solving the structural modeling problem is thus equivalent to finding a solution function u verifying

$$\forall v, \int_\Omega \lambda_\epsilon^2 \ \left(\partial_{ij}^2 u\ \partial_{ij}^2 v\right) d\Omega + \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ (u(\boldsymbol{p})-\alpha_{\boldsymbol{p}})v(\boldsymbol{p}) = 0, \tag{C.9}$$

which is equivalent to Equation (2.56) (p.82).

D Approximation with the bases functions: equations details

Spaces U and V

For readability, Equations (2.5) and (2.6) (p.60) are repeated here:

$$U = \{u(\boldsymbol{x}) = \sum_{l=1}^{N_U} \Phi_l(\boldsymbol{x})\ u_l = \boldsymbol{\Phi}^T \cdot \boldsymbol{U} \mid \boldsymbol{x}\in\Omega\}, \tag{D.1}$$

$$V = \{v(\boldsymbol{x}) = \sum_{l=1}^{N_V} \Psi_l(\boldsymbol{x})\ v_l = \boldsymbol{\Psi}^T \cdot \boldsymbol{V} \mid \boldsymbol{x}\in\Omega\}. \tag{D.2}$$

Development of the data term

$$\begin{aligned}\sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2\ (u(\boldsymbol{p})-\alpha_{\boldsymbol{p}})\ v(\boldsymbol{p}) &= \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2\ u(\boldsymbol{p})\ v(\boldsymbol{p}) - \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2\ \alpha_{\boldsymbol{p}}\ v(\boldsymbol{p}) \\ &\approx \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2\ (\boldsymbol{\Phi}^T(\boldsymbol{p})\cdot\boldsymbol{U})\cdot(\boldsymbol{\Psi}^T(\boldsymbol{p})\cdot\boldsymbol{V}) - \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2\ \alpha_{\boldsymbol{p}}\ (\boldsymbol{\Psi}^T(\boldsymbol{p})\cdot\boldsymbol{V}) \\ &= \boldsymbol{V}^T\cdot\left(\sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2\ \left(\boldsymbol{\Psi}(\boldsymbol{p})\cdot\boldsymbol{\Phi}^T(\boldsymbol{p})\right)\right)\cdot\boldsymbol{U} - \boldsymbol{V}^T\cdot\left(\sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2\ \alpha_{\boldsymbol{p}}\ \boldsymbol{\Psi}(\boldsymbol{p})\right)\end{aligned} \tag{D.3}$$

Development of the energy term

$$\begin{aligned}\int_\Omega \lambda_\epsilon^2\left(\partial_{ij}^2 u\ \partial_{ij}^2 v\right)d\Omega &\approx \int_\Omega \lambda_\epsilon^2(\partial_{ij}^2\boldsymbol{\Phi}^T\cdot\boldsymbol{U})\cdot(\partial_{ij}^2\boldsymbol{\Psi}^T\cdot\boldsymbol{V})d\Omega \\ &= \int_\Omega \lambda_\epsilon^2(\boldsymbol{V}^T\cdot\partial_{ij}^2\boldsymbol{\Psi})\cdot(\partial_{ij}^2\boldsymbol{\Phi}^T\cdot\boldsymbol{U})d\Omega \\ &= \boldsymbol{V}^T\cdot\left(\int_\Omega \lambda_\epsilon^2\left(\partial_{ij}^2\boldsymbol{\Psi}\cdot\partial_{ij}^2\boldsymbol{\Phi}^T\right)d\Omega\right)\cdot\boldsymbol{U}\end{aligned} \tag{D.4}$$

Development of the minimization problem

The minimization problem (Equation (2.56), p.82) is written as

$$\forall\boldsymbol{V},\ \boldsymbol{V}^T\cdot\left(\int_\Omega \lambda_\epsilon^2\left(\partial_{ij}^2\boldsymbol{\Psi}\cdot\partial_{ij}^2\boldsymbol{\Phi}^T\right)d\Omega\right)\cdot\boldsymbol{U}+\boldsymbol{V}^T\cdot\left(\sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2\ \left(\boldsymbol{\Psi}(\boldsymbol{p})\cdot\boldsymbol{\Phi}^T(\boldsymbol{p})\right)\right)\cdot\boldsymbol{U} = \boldsymbol{V}^T\cdot\left(\sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2\ \alpha_{\boldsymbol{p}}\ \boldsymbol{\Psi}(\boldsymbol{p})\right), \tag{D.5}$$

which is equivalent to the linear system of Equation (2.57) (p.82).

E Least squares equivalence in a Bubnov-Galerkin scheme: equations details

Bubnov-Galerkin scheme

In a BG scheme, system (2.58) (p.82) becomes

$$\left(\sum_{\omega\in\boldsymbol{\Omega}}\int_\omega \lambda_\epsilon^2\left(\partial_{ij}^2\boldsymbol{\Phi}\cdot\partial_{ij}^2\boldsymbol{\Phi}^T\right)d\omega + \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2\ \left(\boldsymbol{\Phi}(\boldsymbol{p})\cdot\boldsymbol{\Phi}^T(\boldsymbol{p})\right)\right)\cdot\boldsymbol{U} = \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2\ \alpha_{\boldsymbol{p}}\ \boldsymbol{\Phi}(\boldsymbol{p}). \tag{E.1}$$

Quadrature with one point of evaluation

If the approximation is made that the term $\left(\partial_{ij}^2\boldsymbol{\Phi}\cdot\partial_{ij}^2\boldsymbol{\Phi}^T\right)$ is piecewise constant between each subdomain ω, then

$$\forall\omega\in\boldsymbol{\Omega},\ \forall\boldsymbol{p}\in\omega,\quad \int_\omega \left(\partial_{ij}^2\boldsymbol{\Phi}\cdot\partial_{ij}^2\boldsymbol{\Phi}^T\right)d\omega \approx \nu_\omega\left(\partial_{ij}^2\boldsymbol{\Phi}(\boldsymbol{p})\cdot\partial_{ij}^2\boldsymbol{\Phi}^T(\boldsymbol{p})\right), \tag{E.2}$$

with ν_ω the volume of the subdomain ω. The system (E.1) is then equivalent to

$$\left(\sum_{\omega\in\boldsymbol{\Omega},\ \boldsymbol{p}\in\omega} \lambda_\epsilon^2\nu_\omega\left(\partial_{ij}^2\boldsymbol{\Phi}(\boldsymbol{p})\cdot\partial_{ij}^2\boldsymbol{\Phi}^T(\boldsymbol{p})\right) + \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2\ \left(\boldsymbol{\Phi}(\boldsymbol{p})\cdot\boldsymbol{\Phi}^T(\boldsymbol{p})\right)\right)\cdot\boldsymbol{U} = \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2\ \alpha_{\boldsymbol{p}}\ \boldsymbol{\Phi}(\boldsymbol{p}). \tag{E.3}$$

In the case of a nodal basis shape functions, with each subdomain $\omega_{\overline{\boldsymbol{p}}} \in \boldsymbol{\Omega}$ respectively centered on a node $\overline{\boldsymbol{p}} \in \boldsymbol{N}$, this equation reduces to

$$\left(\sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}} \lambda_\epsilon^2 \nu_{\overline{\boldsymbol{p}}} \left(\partial_{ij}^2 \boldsymbol{\Phi}(\overline{\boldsymbol{p}}) \cdot \partial_{ij}^2 \boldsymbol{\Phi}^T(\overline{\boldsymbol{p}}) \right) + \sum_{\boldsymbol{p} \in \boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ \left(\boldsymbol{\Phi}(\boldsymbol{p}) \cdot \boldsymbol{\Phi}^T(\boldsymbol{p}) \right) \right) \cdot \boldsymbol{U} = \sum_{\boldsymbol{p} \in \boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ \alpha_{\boldsymbol{p}} \ \boldsymbol{\Phi}(\boldsymbol{p}). \tag{E.4}$$

Direct least squares approach

If the discretization of the domain Ω and the quadrature point approximation are applied directly on the standard minimization problem (Equation (2.55), p.81), then the functional J is a sum of squared terms:

$$J(u) \approx \frac{1}{2} \sum_{\forall \omega \in \boldsymbol{\Omega},\ \boldsymbol{p} \in \omega} \lambda_\epsilon^2 \nu_\omega \left(\partial_{ij}^2 u(\boldsymbol{p}) \right)^2 + \frac{1}{2} \sum_{\boldsymbol{p} \in \boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ (u(\boldsymbol{p}) - \alpha_{\boldsymbol{p}})^2. \tag{E.5}$$

In the same case of basis shape functions as above, this equation reduces to

$$J(u) \approx \frac{1}{2} \sum_{\forall \overline{\boldsymbol{p}} \in \boldsymbol{N}} \lambda_\epsilon^2 \nu_{\overline{\boldsymbol{p}}} \left(\partial_{ij}^2 \boldsymbol{\Phi}(\overline{\boldsymbol{p}})^T \cdot \boldsymbol{U} \right)^2 + \frac{1}{2} \sum_{\boldsymbol{p} \in \boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ \left(\boldsymbol{\Phi}(\boldsymbol{p})^T \cdot \boldsymbol{U} - \alpha_{\boldsymbol{p}} \right)^2. \tag{E.6}$$

A classical approach is to write each squared term individually as an equation in a least squares system. The DPV constraints equations and the energy approximation equations are written separately, giving system (2.60) (p.83) repeated here:

$$\left\{ \begin{array}{rll} \lambda_\epsilon \sqrt{\nu_{\overline{\boldsymbol{p}}}} \ \left(\partial_{ij}^2 \boldsymbol{\Phi}(\overline{\boldsymbol{p}})^T \cdot \boldsymbol{U} \right) & = 0, & \forall \overline{\boldsymbol{p}} \in \boldsymbol{N} \\ \lambda_{\boldsymbol{p}} \ \left(\boldsymbol{\Phi}(\boldsymbol{p})^T \cdot \boldsymbol{U} \right) & = \lambda_{\boldsymbol{p}} \ \alpha_{\boldsymbol{p}}, & \forall \boldsymbol{p} \in \boldsymbol{D} \end{array} \right. . \tag{E.7}$$

In matrix form, this system is written as

$$\boldsymbol{A} \cdot \boldsymbol{U} = \boldsymbol{B}. \tag{E.8}$$

The least squares solution to this problem is

$$\boldsymbol{A}^T \cdot \boldsymbol{A} \cdot \boldsymbol{U} = \boldsymbol{A}^T \cdot \boldsymbol{B}, \tag{E.9}$$

with

$$\boldsymbol{A}^T \cdot \boldsymbol{A} = \sum_{\overline{\boldsymbol{p}} \in \boldsymbol{N}} \lambda_\epsilon^2 \nu_{\overline{\boldsymbol{p}}} \left(\partial_{ij}^2 \boldsymbol{\Phi}(\overline{\boldsymbol{p}}) \cdot \partial_{ij}^2 \boldsymbol{\Phi}^T(\overline{\boldsymbol{p}}) \right) + \sum_{\boldsymbol{p} \in \boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ \left(\boldsymbol{\Phi}(\boldsymbol{p}) \cdot \boldsymbol{\Phi}^T(\boldsymbol{p}) \right), \tag{E.10}$$

and with

$$\boldsymbol{A}^T \cdot \boldsymbol{B} = \sum_{\boldsymbol{p} \in \boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ \alpha_{\boldsymbol{p}} \ \boldsymbol{\Phi}(\boldsymbol{p}). \tag{E.11}$$

The same system as the one given by Equation (E.3) is obtained, hence the equivalence.

F Normalizing the influence of the number of data constraints in a least squares system

If a DPV constraint is added twice in the modeling problem, the standard functional J can be written as

$$J(u) = \frac{1}{2} \int_\Omega \lambda_\epsilon^2 \left(\partial_{ij}^2 u \right)^2 d\Omega + \frac{1}{2} \left(\left(\sum_{\boldsymbol{p} \in (\boldsymbol{D} - \boldsymbol{p}')} \lambda_{\boldsymbol{p}}^2 \ (u(\boldsymbol{p}) - \alpha_{\boldsymbol{p}})^2 \right) + 2\lambda_{\boldsymbol{p}'}^2 \ (u(\boldsymbol{p}') - \alpha_{\boldsymbol{p}'})^2 \right), \tag{F.1}$$

with $\boldsymbol{p}'$ the data point added twice, $\alpha_{\boldsymbol{p}'}$ its expected iso-value, and $\lambda_{\boldsymbol{p}'}$ the associated constraint weight.

Normalizing the influence of these two repetitive constraints to act as one is handled by dividing the last term of Equation (F.1) (i.e., $2\lambda_{\boldsymbol{p}'}^2 \ (u(\boldsymbol{p}') - \alpha_{\boldsymbol{p}'})^2$) by two. In a least squares system, this

is equivalent to multiplying the equations of these two constraints by $\frac{1}{\sqrt{2}}$ respectively. This can be generalized for any number of repetition N_r of a same data constraint with a normalization factor of $\frac{1}{\sqrt{N_r}}$.

The same concept can also be applied to normalize the influence of the number of data constraints against the energy penalization (i.e., with a normalization factor of $\frac{1}{\sqrt{\#\boldsymbol{D}}}$ in the minimization problem). This is supposed to avoid an abnormal over-fit to the data when increasing their number in a regularly sampled setting. In practice, this effect is negligible even without the normalization.

G Lagrange multipliers: equations details

By using the approximation function u (i.e., defined with an arbitrary basis shape functions $\boldsymbol{\Phi}$) directly in the standard minimization problem (Equation (2.55), p.81), the functional J becomes

$$J(\boldsymbol{U}) = \frac{1}{2}\int_\Omega \lambda_\epsilon^2 \left(\partial_{ij}^2 \boldsymbol{\Phi}^T \cdot \boldsymbol{U}\right)^2 d\Omega + \frac{1}{2}\sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ (\boldsymbol{\Phi}^T(\boldsymbol{p})\cdot\boldsymbol{U} - \alpha_{\boldsymbol{p}})^2, \tag{G.1}$$

which is a Bubnov-Galerkin scheme where the coefficients of the test function $\boldsymbol{V}$ are chosen to be the same as the coefficients of the solution function $\boldsymbol{U}$.

Let the set $\boldsymbol{D}$ be subdivided into two sets $\boldsymbol{D}(soft)$ and $\boldsymbol{D}(hard)$ of soft and hard DPV constraints, respectively. The hard DPV constraints are enforced in the modeling problem using the Lagrange multipliers. The functional J is redefined with its newly added variables $\boldsymbol{\lambda}$ as

$$J(\boldsymbol{U},\boldsymbol{\lambda}) = \frac{1}{2}\int_\Omega \lambda_\epsilon^2 \left(\partial_{ij}^2 \boldsymbol{\Phi}^T \cdot \boldsymbol{U}\right)^2 d\Omega + \frac{1}{2}\sum_{\boldsymbol{p}\in\boldsymbol{D}(soft)} \lambda_{\boldsymbol{p}}^2 \ (\boldsymbol{\Phi}^T(\boldsymbol{p})\cdot\boldsymbol{U} - \alpha_{\boldsymbol{p}})^2 + \boldsymbol{\lambda}^T\cdot(\boldsymbol{\Theta}\cdot\boldsymbol{U}-\boldsymbol{\alpha}), \tag{G.2}$$

with $\boldsymbol{\lambda}$ and $\boldsymbol{\alpha}$ the vectors of unknown weights and known iso-values respectively associated to each hard constraint $\boldsymbol{p}\in\boldsymbol{D}(hard)$, and

$$\boldsymbol{\Theta} = \left[\boldsymbol{\Phi}(\boldsymbol{p})^T \ \mid \ \forall \boldsymbol{p}\in\boldsymbol{D}(hard)\right] \tag{G.3}$$

a matrix of dimensions $(\#\boldsymbol{D}(hard), \#\boldsymbol{N})$. The last term of J could be written as

$$\boldsymbol{\lambda}^T\cdot(\boldsymbol{\Theta}\cdot\boldsymbol{U}-\boldsymbol{\alpha}) = \sum_{\boldsymbol{p}\in\boldsymbol{D}(hard)} \lambda_{\boldsymbol{p}} \ \left(\boldsymbol{\Phi}^T(\boldsymbol{p})\cdot\boldsymbol{U} - \alpha_{\boldsymbol{p}}\right), \tag{G.4}$$

but where the weights $\lambda_{\boldsymbol{p}}^{hard}$ are unknown (in opposition to the other $\lambda_{\boldsymbol{p}}^{soft}$ weights).

A solution to the minimization problem of J is obtained by solving the system

$$\begin{cases} \partial_{\boldsymbol{U}} J(\boldsymbol{U},\boldsymbol{\lambda}) &= 0 \\ \partial_{\boldsymbol{\lambda}} J(\boldsymbol{U},\boldsymbol{\lambda}) &= 0 \end{cases}, \tag{G.5}$$

with

$$\partial_{\boldsymbol{U}} J(\boldsymbol{U},\boldsymbol{\lambda}) = \int_\Omega \lambda_\epsilon^2 \left(\partial_{ij}^2\boldsymbol{\Phi}\cdot\partial_{ij}^2\boldsymbol{\Phi}^T\right)\cdot\boldsymbol{U}d\Omega + \sum_{\boldsymbol{p}\in\boldsymbol{D}(soft)} \lambda_{\boldsymbol{p}}^2 \left(\left(\boldsymbol{\Phi}(\boldsymbol{p})\cdot\boldsymbol{\Phi}^T(\boldsymbol{p})\right)\cdot\boldsymbol{U} - \alpha_{\boldsymbol{p}}\boldsymbol{\Phi}(\boldsymbol{p})\right) + \boldsymbol{\Theta}^T\cdot\boldsymbol{\lambda} \tag{G.6}$$

and with

$$\partial_{\boldsymbol{\lambda}} J(\boldsymbol{U},\boldsymbol{\lambda}) = \boldsymbol{\Theta}\cdot\boldsymbol{U} - \boldsymbol{\alpha}. \tag{G.7}$$

When discretizing the domain of study as presented in Section 2.2.4.2 (p.82), this system becomes the same as system (2.72) (p.85), with

$$\boldsymbol{K} = \sum_{\omega\in\boldsymbol{\Omega}} \lambda_\epsilon^2 \left(\partial_{ij}^2\boldsymbol{\Phi}\cdot\partial_{ij}^2\boldsymbol{\Phi}^T\right) d\omega + \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \left(\boldsymbol{\Phi}(\boldsymbol{p})\cdot\boldsymbol{\Phi}^T(\boldsymbol{p})\right), \tag{G.8}$$

and with

$$\boldsymbol{F} = \sum_{\boldsymbol{p}\in\boldsymbol{D}} \lambda_{\boldsymbol{p}}^2 \ \alpha_{\boldsymbol{p}} \ \boldsymbol{\Phi}(\boldsymbol{p}). \tag{G.9}$$

H Plots of weight and MLS functions

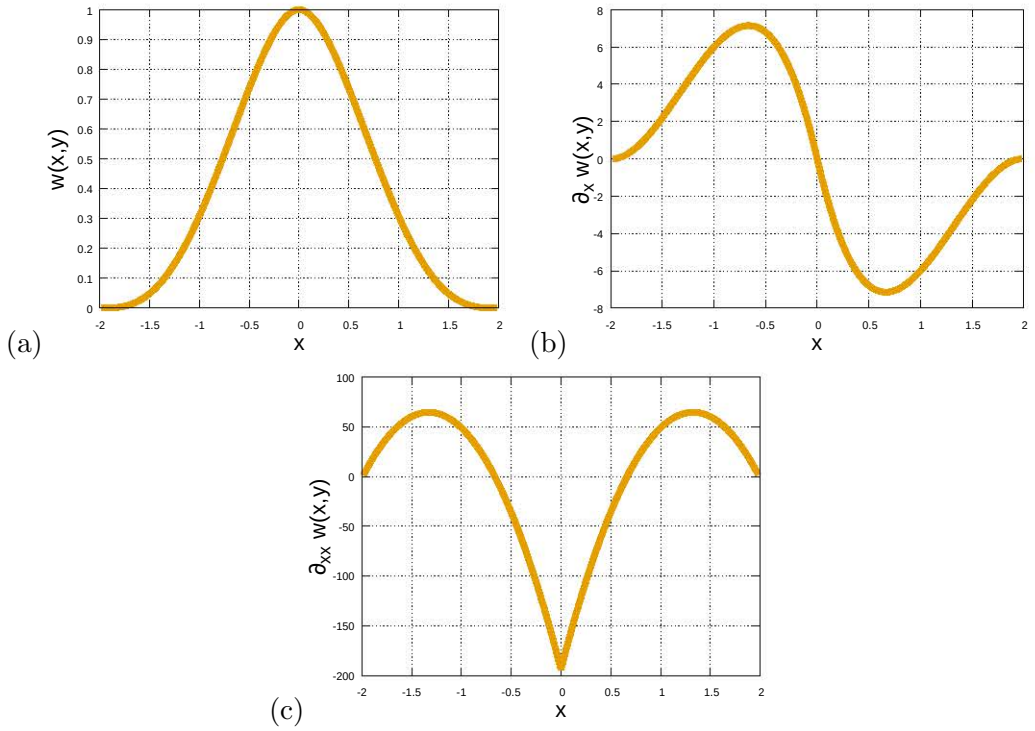


Figure H.1: *1D plots of the fourth order splines and their derivatives, with* $\overline{\boldsymbol{p}} = 0$ *and* $\rho_x = 1.99$. *(a) The weight function. (b) The first order derivative. (c) The second order derivative.*

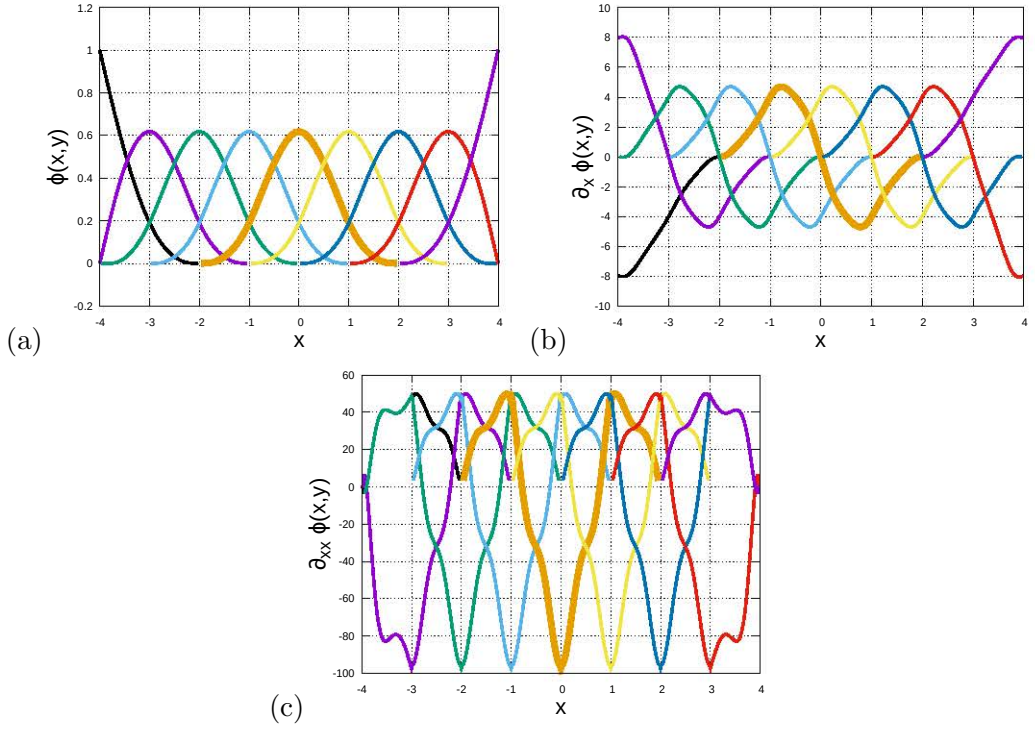


Figure H.2: *1D plots of the MLS functions and their derivatives on* 9 *regularly sampled nodes with* $\rho = 1.99$. *(a) The conventional MLS functions. (b) The first order derivatives. (c) The second order derivatives.*

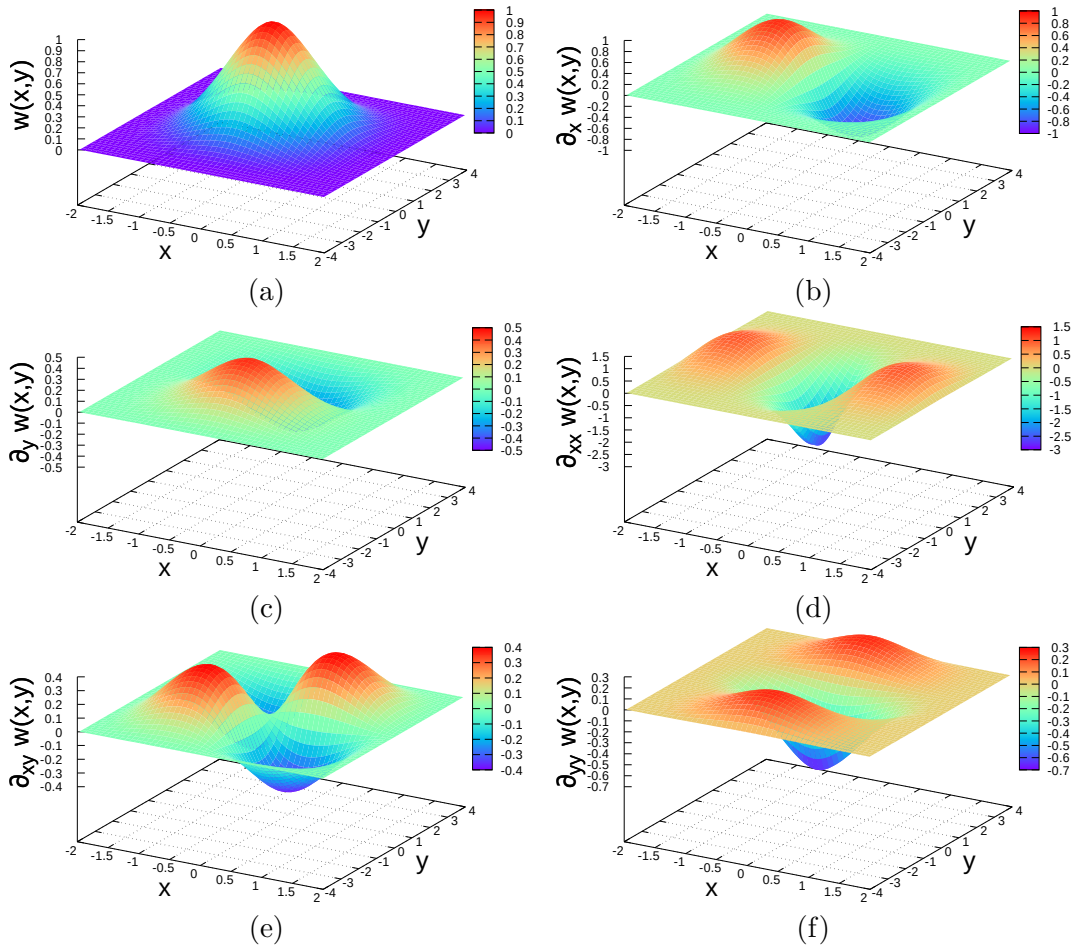


Figure H.3: *2D plots of the fourth order spline and its derivatives, with* $\overline{\boldsymbol{p}} = \{0, 0\}$, $\rho_x = 1.99$ *and* $\rho_y = 3.98$.

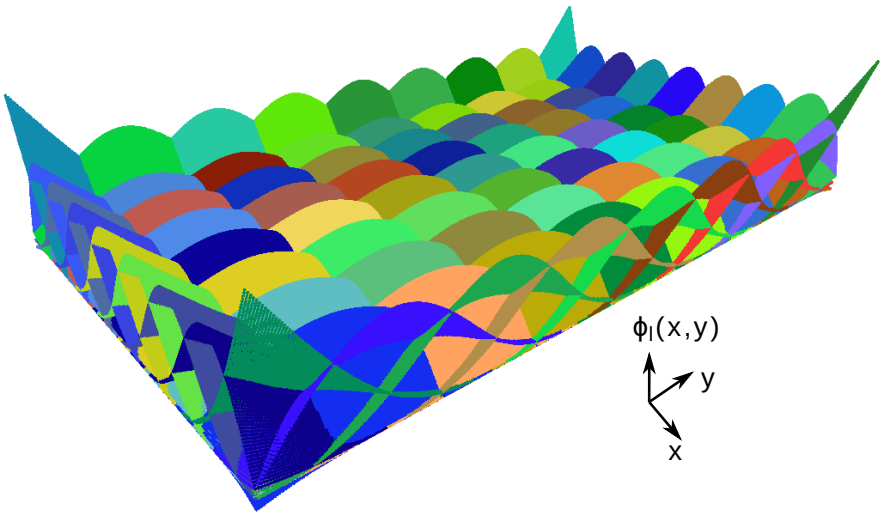


Figure H.4: *2D plot of the superposition of* 81 *MLS functions centered on* (9×9) *regularly sampled nodes, with a nodal spacing of* 1 *in the* x *axis and* 2 *in the* y *axis, and with* $\rho = 1.99$.

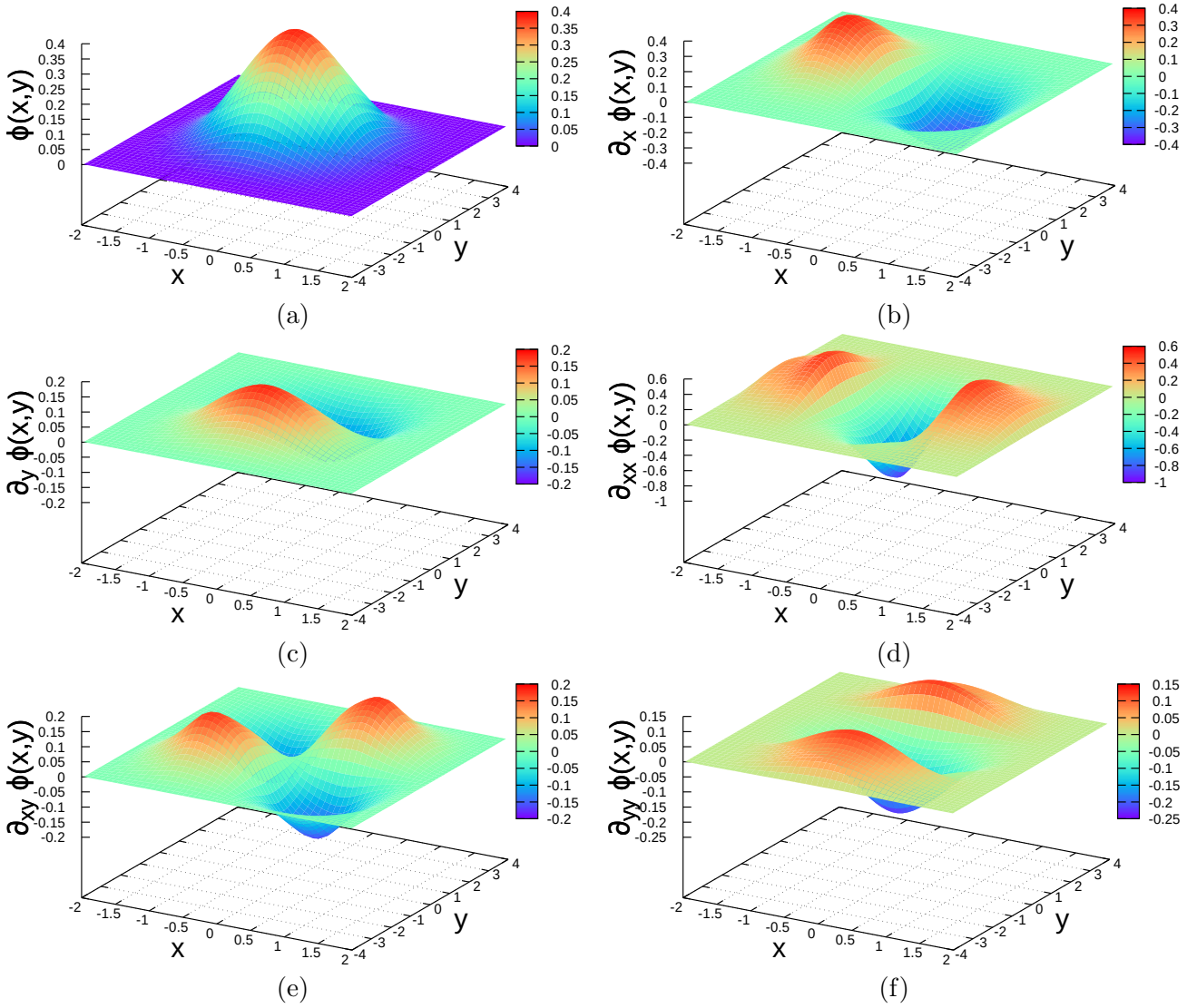


Figure H.5: *2D plots of the MLS functions and their derivatives on the center node ($\overline{\boldsymbol{p}} = \{0, 0\}$) of Figure H.4.*

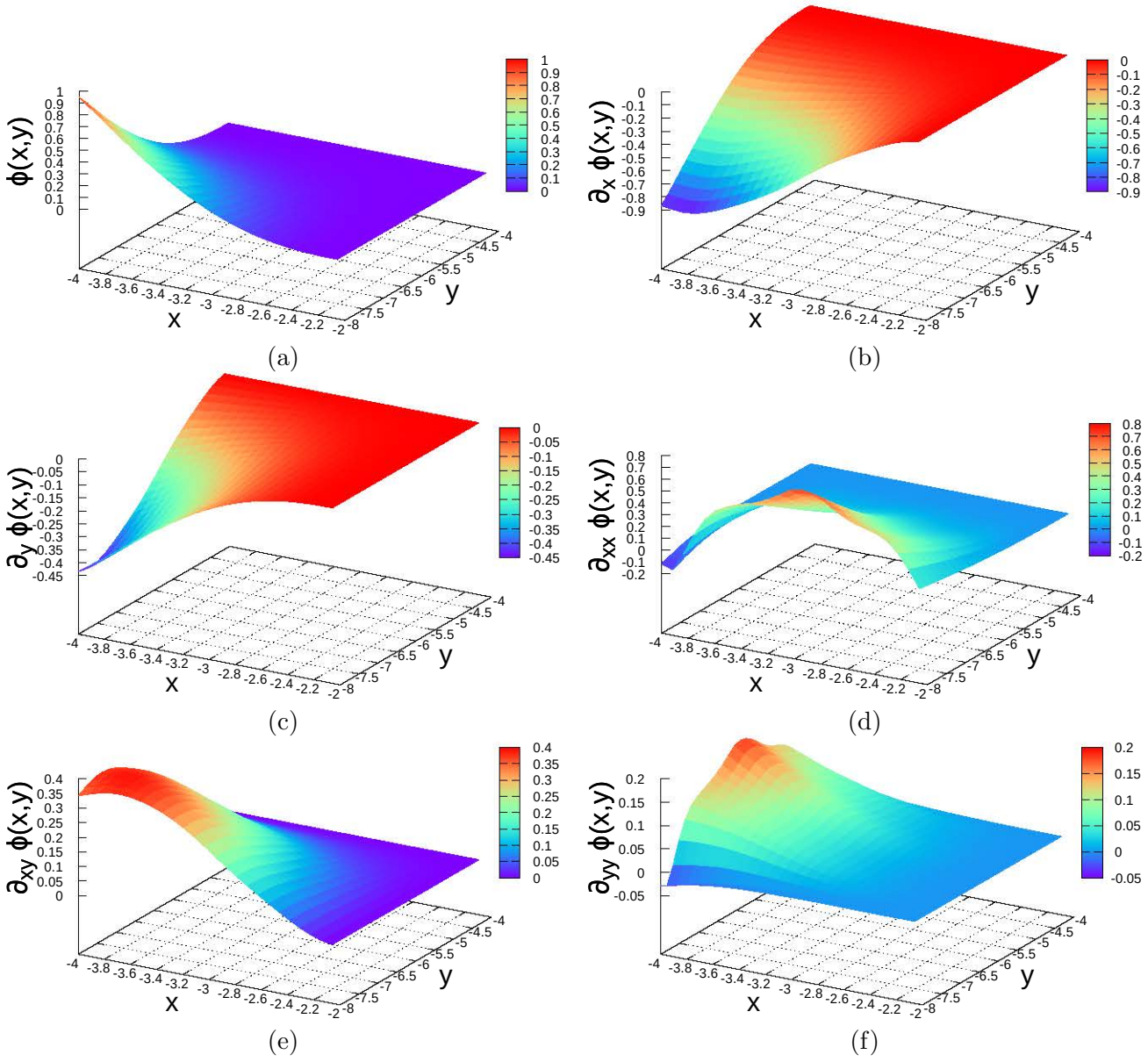


Figure H.6: *2D plots of the MLS functions and their derivatives on a corder node* ($\overline{\boldsymbol{p}} = \{-4, -4\}$) *of Figure H.4.*

I Integration of an homogeneous directional rigidity

Let's consider the integration term on the rotation angle θ in Equation (5.11) (p.146),

$$\forall \boldsymbol{x} \in \Omega, \quad \int_0^\pi \left(\begin{bmatrix} \cos\theta & \sin\theta \end{bmatrix} \cdot \begin{bmatrix} \partial^2_{xx} u & \partial^2_{xy} u \\ \partial^2_{xy} u & \partial^2_{yy} u \end{bmatrix} \cdot \begin{bmatrix} cos\theta \\ st \end{bmatrix} \right)^2 d\theta \tag{I.1}$$

$$= \int_0^\pi \left(\cos^2\theta \left(\partial^2_{xx} u\right) + 2\cos\theta \sin\theta \left(\partial^2_{xy} u\right) + \sin^2\theta \left(\partial^2_{yy} u\right) \right)^2 d\theta \tag{I.2}$$

$$= \int_0^\pi \Big(\cos^4\theta \left(\partial^2_{xx} u\right)^2 + 4\cos^2\theta \sin^2\theta \left(\partial^2_{xy} u\right)^2 + \sin^4\theta \left(\partial^2_{yy} u\right)^2 + 4\cos^3\theta sin\theta \left(\partial^2_{xx} u\right)\left(\partial^2_{xy} u\right) + 2\cos^2\theta \sin^2\theta \left(\partial^2_{xx} u\right)\left(\partial^2_{yy} u\right) + 4\cos\theta \sin^3\theta \left(\partial^2_{xy} u\right)\left(\partial^2_{yy} u\right) \Big) d\theta. \tag{I.3}$$

Each integral term is detailed separately, as

$$\int_0^\pi \cos^4\theta \left(\partial^2_{xx} u\right)^2 d\theta = \frac{3\pi}{8} \left(\partial^2_{xx} u\right)^2, \tag{I.4}$$

$$\int_0^\pi 4\cos^2\theta \sin^2\theta \left(\partial^2_{xy} u\right)^2 d\theta = \frac{4\pi}{8} \left(\partial^2_{xy} u\right)^2, \tag{I.5}$$

$$\int_0^\pi \sin^4\theta \left(\partial^2_{yy} u\right)^2 d\theta = \frac{3\pi}{8} \left(\partial^2_{yy} u\right)^2, \tag{I.6}$$

$$\int_0^\pi 4\cos^3\theta sin\theta \left(\partial^2_{xx} u\right)\left(\partial^2_{xy} u\right) d\theta = 0, \tag{I.7}$$

$$\int_0^\pi 2\cos^2\theta \sin^2\theta \left(\partial^2_{xx} u\right)\left(\partial^2_{yy} u\right) d\theta = \frac{2\pi}{8} \left(\partial^2_{xx} u\right)\left(\partial^2_{yy} u\right), \tag{I.8}$$

$$\int_0^\pi 4\cos\theta \sin^3\theta \left(\partial^2_{xy} u\right)\left(\partial^2_{yy} u\right) d\theta = 0. \tag{I.9}$$

$$\tag{I.10}$$

We then replace those terms in the above equality as

$$\forall \boldsymbol{x} \in \Omega, \quad \int_0^\pi \left(\begin{bmatrix} \cos\theta & \sin\theta \end{bmatrix} \cdot \begin{bmatrix} \partial^2_{xx} u & \partial^2_{xy} u \\ \partial^2_{xy} u & \partial^2_{yy} u \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & \sin\theta \end{bmatrix} \right)^2 d\theta \tag{I.11}$$

$$= \frac{3\pi}{8} \left(\partial^2_{xx} u\right)^2 + \frac{4\pi}{8} \left(\partial^2_{xy} u\right)^2 + \frac{3\pi}{8} \left(\partial^2_{yy} u\right)^2 + \frac{2\pi}{8} \left(\partial^2_{xx} u\right)\left(\partial^2_{yy} u\right) \tag{I.12}$$

$$= \frac{\pi}{4} \left(\left(\partial^2_{xx} u\right)^2 + \left(\partial^2_{yy} u\right)^2 + 2\left(\partial^2_{xy} u\right)^2 + \frac{1}{2}\left(\partial^2_{xx} u\right)^2 + \frac{1}{2}\left(\partial^2_{yy} u\right)^2 + \left(\partial^2_{xx} u\right)\left(\partial^2_{yy} u\right) \right) \tag{I.13}$$

$$= \frac{\pi}{4} \left(\left(\partial^2_{xx} u\right)^2 + \left(\partial^2_{yy} u\right)^2 + 2\left(\partial^2_{xy} u\right)^2 + \left(\left(\partial^2_{xx} u\right) + \left(\partial^2_{yy} u\right)\right)^2 \right), \tag{I.14}$$

$$\tag{I.15}$$

hence Equation (5.12) (p.146).

J Integration of an anisotropic directional rigidity

Let's consider the integration term on the rotation angle θ in Equation (5.14) (p.146),

$$\forall \boldsymbol{x} \in \Omega, \quad \int_0^{\pi} \left((\boldsymbol{v}_1 \cos\theta + \boldsymbol{v}_2 \sin\theta)^T \cdot \boldsymbol{H}(u) \cdot (\boldsymbol{v}_1 \cos\theta + \boldsymbol{v}_2 \sin\theta) \right)^2 d\theta \tag{J.1}$$

$$= \int_0^{\pi} \left(\cos^2\theta ({\boldsymbol{v}_1}^T \cdot \boldsymbol{H}(u) \cdot \boldsymbol{v}_1) + 2\cos\theta\sin\theta ({\boldsymbol{v}_1}^T \cdot \boldsymbol{H}(u) \cdot \boldsymbol{v}_2) + \sin^2\theta ({\boldsymbol{v}_2}^T \cdot \boldsymbol{H}(u) \cdot \boldsymbol{v}_2) \right)^2 d\theta \tag{J.2}$$

$$= \int_0^{\pi} \left(\cos^2\theta (A) + 2\cos\theta\sin\theta (B) + \sin^2\theta (C) \right)^2 d\theta \tag{J.3}$$

$$= \int_0^{\pi} \cos^4\theta (A)^2 + \sin^4\theta (C)^2 + 4\cos^2\theta\sin^2\theta (B)^2 + 4\cos^3\theta\sin\theta (AB) + 2\cos^2\theta\sin^2\theta (AC) + 4\cos\theta\sin^3\theta (BC) d\theta \tag{J.4}$$

$$= \frac{3\pi}{8}(A)^2 + \frac{3\pi}{8}(C)^2 + 4\frac{\pi}{8}(B)^2 + 2\frac{\pi}{8}(AC) d\theta \tag{J.5}$$

$$= \frac{\pi}{4} \left((A)^2 + (C)^2 + 2(B)^2 + \frac{1}{2}(A+C)^2 \right) d\theta, \tag{J.6}$$

where the Hessian matrix $\boldsymbol{H}$ is symetric.

The squared terms A, B, C and $(A+C)$ are explicited separately as

$$A = \begin{bmatrix} \cos\alpha & \sin\alpha \end{bmatrix} \cdot \begin{bmatrix} \partial^2_{xx} u & \partial^2_{xy} u \\ \partial^2_{xy} u & \partial^2_{yy} u \end{bmatrix} \cdot \begin{bmatrix} \cos\alpha \\ \sin\alpha \end{bmatrix} = \cos^2\alpha \ \partial^2_{xx} u + \sin^2\alpha \ \partial^2_{yy} u + \sin 2\alpha \ \partial^2_{xy} u, \tag{J.7}$$

$$B = \begin{bmatrix} \cos\alpha & \sin\alpha \end{bmatrix} \cdot \begin{bmatrix} \partial^2_{xx} u & \partial^2_{xy} u \\ \partial^2_{xy} u & \partial^2_{yy} u \end{bmatrix} \cdot \begin{bmatrix} -a\sin\alpha \\ a\cos\alpha \end{bmatrix} = a \left(\cos\alpha \ \sin\alpha (\partial^2_{yy} u - \partial^2_{xx} u) + \cos 2\alpha \ \partial^2_{xy} u \right), \tag{J.8}$$

$$C = \begin{bmatrix} -a\sin\alpha & a\cos\alpha \end{bmatrix} \cdot \begin{bmatrix} \partial^2_{xx} u & \partial^2_{xy} u \\ \partial^2_{xy} u & \partial^2_{yy} u \end{bmatrix} \cdot \begin{bmatrix} -a\sin\alpha \\ a\cos\alpha \end{bmatrix} = a^2 \left(\sin^2\alpha \ \partial^2_{xx} u + \cos^2\alpha \ \partial^2_{yy} u - \sin 2\alpha \ \partial^2_{xy} u \right), \tag{J.9}$$

$$(A+C) = (\cos^2\alpha + a^2\sin^2\alpha) \ \partial^2_{xx} u + (\sin^2\alpha + a^2\cos^2\alpha) \ \partial^2_{yy} u + \sin 2\alpha (1 - a^2) \ \partial^2_{xy} u, \tag{J.10}$$

hence Equation (5.15) (p.146).

Bibliography

A. Adamson & M. Alexa [2006]. Anisotropic point set surfaces. *Computer Graphics Forum*, **25**(4):717–724.

J. Ahlberg, E. Nilson & J. Walsh [1967]. *The theory of splines and its applications*. Academic Press, New York.

S. Alcaraz, R. Lane, K. Spragg, S. Milicich, F. Sepulveda & G. Bignall [2011]. 3D geological modelling using new leapfrog geothermal software. In: *Proceedings of the 36th Workshop on Geothermal Reservoir Engineering, Stanford, CA, USA*, vol. 31.

M. Alexa & A. Adamson [2009]. Interpolatory point set surfaces—convexity and hermite data. *ACM Transactions on Graphics (TOG)*, **28**(2):20.

M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin & C. Silva [2001]. Point set surfaces. In: T. Ertl, K. Joy & A. Varshney (Eds.), *Proceedings of the IEEE Visualization Conference*, pp. 21–28.

P. Anquez, G. Caumon, J. Pellerin & B. Lévy [2017]. Automatic sealing and simplification of 3D geological surface models using topology recovery. In: *79th EAGE Conference and Exhibition 2017*.

P. Anquez, J. Pellerin, M. Irakarama, P. Cupillard, B. Lévy & G. Caumon [2019]. Automatic correction and simplification of geological maps and cross-sections for numerical simulations. *Comptes Rendus Geoscience*.

ARANZ Geo [2019]. Leapfrog geo. http://www.leapfrog3d.com/products/leapfrog-geo. Accessed 15.02.2019.

S. Atluri, J. Cho & H.-G. Kim [1999a]. Analysis of thin beams, using the meshless local petrov–Galerkin method, with generalized moving least squares interpolations. *Computational Mechanics*, **24**(5):334–347.

S. N. Atluri, H.-G. Kim & J. Y. Cho [1999b]. A critical assessment of the truly meshless local Petrov-Galerkin (MLPG), and local boundary integral equation (LBIE) methods. *Computational mechanics*, **24**(5):348–372.

S. N. Atluri & S. Shen [2002]. The meshless local Petrov-Galerkin (MLPG) method: A simple & less-costly alternative to the finite element and boundary element methods. *CMES - Computer Modeling in Engineering and Sciences*, **3**(1):11–51.

S. N. Atluri & T. Zhu [1998]. A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational mechanics*, **22**(2):117–127.

C. Aug [2004]. *Modélisation géologique 3D et caractérisation des incertitudes par la méthode du champ de potentiel*. Ph.D. thesis, École Nationale Supérieure des Mines de Paris.

I. Babuška, U. Banerjee & J. E. Osborn [2003]. Meshless and generalized finite element methods: A survey of some major results. pp. 1–20. Springer.

I. Babuska & J. M. Melenk [1995]. The partition of unity finite element method. Tech. rep., École polytechnique fédérale de Zurich.

I. Babuška & J. M. Melenk [1997]. The partition of unity method. *International journal for numerical methods in engineering*, **40**(4):727–758.

F. BaDughaish [2013]. A meshless surface-fitting method for fault detection. In: *IPTC 2013: International Petroleum Technology Conference*.

K.-J. Bathe & E. L. Wilson [1976]. *Numerical methods in finite element analysis*. Prentice-Hall Englewood Cliffs, NJ.

R. Beatson & L. Greengard [1997]. A short course on fast multipole methods. *New York*, pp. 1–37.

R. K. Beatson, W. Light & S. Billings [2001]. Fast solution of the radial basis function interpolation equations: Domain decomposition methods. *SIAM Journal on Scientific Computing*, **22**(5):1717–1740.

S. Beissel & T. Belytschko [1996]. Nodal integration of the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, **139**(1-4):49–74.

V. Belikov [1997]. The non-Sibsonian interpolation: A new method of interpolation of the values of a function on an arbitrary set of points. *Computational Mathematics and Mathematical Physics*, **37**(1):9–15.

T. Belytschko, L. Gu & Y. Lu [1994a]. Fracture and crack growth by Element Free Galerkin methods. *Modelling and Simulation in Materials Science and Engineering*, **2**(3A):519.

T. Belytschko, Y. Krongauz, J. Dolbow & C. Gerlach [1998]. On the completeness of meshfree particle methods. *International Journal for Numerical Methods in Engineering*, **43**(5):785–819.

T. Belytschko, Y. Krongauz, M. Fleming, D. Organ & W. K. S. Liu [1996a]. Smoothing and accelerated computations in the Element Free Galerkin method. *Journal of Computational and Applied Mathematics*, **74**(1):111–126.

T. Belytschko, Y. Krongauz, D. Organ, M. Fleming & P. Krysl [1996b]. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, **139**(1-4):3–47.

T. Belytschko, Y. Y. Lu & L. Gu [1994b]. Element-Free Galerkin methods. *International journal for numerical methods in engineering*, **37**(2):229–256.

T. Belytschko, N. Moës, S. Usui & C. Parimi [2001]. Arbitrary discontinuities in finite elements. *International Journal for Numerical Methods in Engineering*, **50**(4):993–1013.

J. R. Berger & A. Karageorghis [2001]. The method of fundamental solutions for layered elastic materials. *Engineering Analysis with Boundary Elements*, **25**(10):877–886.

M. Berger, P. Alliez, A. Tagliasacchi, L. M. Seversky, C. T. Silva, J. A. Levine & A. Sharf [2014]. State of the Art in Surface Reconstruction from Point Clouds. *Proceedings of the Eurographics 2014, Eurographics STARs*, **1**:161–185.

M. Berger, J. A. Levine, L. G. Nonato, G. Taubin & C. T. Silva [2013]. A Benchmark for Surface Reconstruction. *ACM Transactions on Graphics*, **32**(2):20:1–20:17.

B. Bhattacharyya [1969]. Bicubic spline interpolation as a method for treatment of potential field data. *Geophysics*, **34**(3):402–423.

J. F. Blinn [1982]. A generalization of algebraic surface drawing. *ACM transactions on graphics (TOG)*, **1**(3):235–256.

A. Boag, Y. Leviatan & A. Boag [1988]. Analysis of acoustic scattering from fluid cylinders using a multifilament source model. *The Journal of the Acoustical Society of America*, **83**(1):1–8.

H. G. Borgos, T. Skov, T. Randen & L. Sonneland [2003]. Automated geometry extraction from 3D seismic data. In: *SEG Technical Program Expanded Abstracts 2003*, pp. 1541–1544. Society of Exploration Geophysicists.

A. Botella, B. Lévy & G. Caumon [2016]. Indirect unstructured hex-dominant mesh generation using tetrahedra recombination. *Computational Geosciences*, **20**(3):437–451.

F. Brezzi [1974]. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. *Revue française d'automatique, informatique, recherche opérationnelle Analyse numérique*, **8**(R2):129–151.

I. C. Briggs [1974]. Machine contouring using minimum curvature. *Geophysics*, **39**(1):39–48.

M. D. Buhmann [2000]. Radial basis functions. *Acta Numerica 2000*, **9**:1–38.

F. Calakli & G. Taubin [2011]. SSD: Smooth signed distance surface reconstruction. *Computer Graphics Forum*, **30**(7):1993–2002.

P. Calcagno, J. P. Chilès, G. Courrioux & A. Guillen [2008]. Geological modelling from field data and geological knowledge. Part I. Modelling method coupling 3D potential-field interpolation and geological rules. *Physics of the Earth and Planetary Interiors*, **171**(1-4):147–157.

T. Carmichael & L. Ailleres [2016]. Method and analysis for the upscaling of structural data. *Journal of Structural Geology*, **83**:121–133.

J. C. Carr, R. K. Beatson & J. B. Cherrie [2001]. Reconstruction and representation of 3D objects with radial basis functions. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 67–76.

G. Caumon [2009]. *Vers une intégration des incertitudes et des processus en géologie numérique.*

Habilitation à diriger la recherche.

G. Caumon, P. Collon-Drouaillet, C. L. C. De Veslud, S. Viseur & J. Sausse [2009]. Surface-based 3D modeling of geological structures. *Mathematical Geosciences*, **41**(8):927–945.

G. Caumon, G. Gray, C. Antoine & M.-O. Titeux [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 Transactions on Geoscience and Remote Sensing*, **51**(3):1613–1621.

G. Caumon, F. Lepage, C. H. Sword & J.-L. Mallet [2004]. Building and editing a sealed geological model. *Mathematical Geology*, **36**(4):405–424.

G. Caumon, A.-L. Tertois & L. Zhang [2007]. Elements for stochastic structural perturbation of stratigraphic models. In: *EAGE Conference on Petroleum Geostatistics*.

B. Chauvin [2017]. *Applicability of the mechanics-based restoration: Boundary conditions, fault network and comparison with a geometrical method*. Ph.D. thesis, Université de Lorraine.

C. Chen [1995]. The method of fundamental solutions for non-linear thermal explosions. *Communications in Numerical Methods in Engineering*, **11**(8):675–681.

C. Chen, a. Karageorghis & Y.-S. Smyrlis [2008]. *The method of fundamental solutions - a meshless method*.

H. Chen, C. Min & F. Gibou [2007]. A supra-convergent finite difference scheme for the Poisson and heat equations on irregular domains and non-graded adaptive Cartesian grids. *Journal of Scientific Computing*, **31**(1-2):19–60.

W.-H. Chen & X.-M. Guo [2001]. Element Free Galerkin method for three-dimensional structural analysis. *Computer Modeling in Engineering and Sciences*, **2**(4):497–508.

Y. Chen, J. Lee & A. Eskandarian [2006]. *Meshless methods in solid mechanics*. Springer.

Z.-Q. Cheng, Y.-Z. Wang, B. Li, K. Xu, G. Dang & S.-Y. Jin [2008]. A survey of methods for moving least squares surfaces. *Volume Graphics*, pp. 9–23.

N. Cherpeau, G. Caumon & B. Lévy [2010]. Stochastic simulations of fault networks in 3D structural modeling. *Comptes Rendus Geoscience*, **342**(9):687–694.

J. P. Chilès, C. Aug, A. Guillen & T. Lees [2004]. Modelling the geometry of geological units and its uncertainty in 3D from structural data : The potential-field method. *Orebody Modeling and Strategic Mine Planning - Spectrum 14*, (July):22–24.

H. Ching & R. Batra [2001]. Determination of crack tip fields in linear elastostatics by the meshless local Petrov-Galerkin (MLPG) method. *CMES- Computer Modeling in Engineering and Sciences*, **2**(2):273–289.

V. Cingoski, N. Miyamoto & H. Yamashita [1998]. Element-Free Galerkin method for electromagnetic field computations. *IEEE Transactions on Magnetics*, **34**(5):3236–3239.

C. S. Co, S. D. Porumbescu & K. I. Joy [2004]. Meshless isosurface generation from multiblock data. In: *Proceedings of the Sixth Joint Eurographics-IEEE TCVG conference on Visualization*, pp. 273–282. Eurographics Association.

A. J. Coleman, C. A.-L. Jackson, O. B. Duffy & M. A. Nikolinakou [2018]. How, where, and when do radial faults grow near salt diapirs? *Geology*.

P. Collon, W. Steckiewicz-Laurent, J. Pellerin, G. Laurent, G. Caumon, G. Reichart & L. Vaute [2015]. 3D geomodelling combining implicit surfaces and Voronoi-based remeshing: A case study in the Lorraine coal basin (France). *Computers & Geosciences*, **77**:29–43.

R. Courant [1950]. *Dirichlet's principle, conformal mapping, and minimal surfaces*. Interscience.

E. Cowan, R. Beatson, H. Ross, W. Fright, T. McLennan, T. Evans, J. Carr, R. Lane, D. Bright, A. Gillman, P. Oshust & M. Titley [2003]. Practical implicit geological modelling. *5th International Mining Geology Conference*, (8):89–99.

E. J. Cowan, R. K. Beatson, W. R. Fright, T. J. Mclennan & T. J. Mitchell [2002]. Rapid geological modelling. *Applied Structural Geology for Mineral Exploration and Mining, International Symposium*, (September):23–25.

T. A. Cross & P. W. Homewood [1997]. Amanz Gressly's role in founding modern stratigraphy. *Geological Society of America Bulletin*, **109**(12):1617–1630.

S. L. Crouch [1983]. *Boundary element methods in solid mechanics, with applications in rock mechanics and geological engineering*. BOOK. George Allen & Unwin.

S. Cuomo, a. Galletti, G. Giunta & a. Starace [2013]. Surface reconstruction from scattered point via RBF interpolation on GPU. *Computer Science and Information Systems (FedCSIS), 2013*

Federated Conference on, pp. 433–440.
C. Dahlstrom [1969]. Balanced cross sections. *Canadian Journal of Earth Sciences*, **6**(4):743–757.
S. De & K. J. Bathe [2000]. The method of finite spheres. *Computational Mechanics*, **25**(4):329–345.
S. De & K.-J. Bathe [2001]. The method of finite spheres with improved numerical integration. *Computers & Structures*, **79**(22-25):2183–2196.
B. R. De Araújo, D. S. Lopes, P. Jepp, J. A. Jorge & B. Wyvill [2015]. A survey on implicit surface polygonization. *ACM Computing Surveys (CSUR)*, **47**(4):1–39.
E. A. de Kemp & K. B. Sprague [2003]. Interpretive tools for 3-D structural geological modeling. Part i: Bezier-based curves, ribbons and grip frames. *GeoInformatica*, **7**(1):55–71.
M. de la Varga, A. Schaaf & F. Wellmann [2019]. GemPy 1.0: Open-source stochastic geological modeling and inversion. *Geoscientific Model Development*.
T. K. Dey & J. Sun [2005]. An adaptive MLS surface for reconstruction with guarantees. In: *Symposium on Geometry processing*, pp. 43–52.
G. Dhatt & G. Touzot [1981]. *Une présentation de la méthode des éléments finis*. Presses Université Laval.
G. A. Dilts [1999]. Moving-least-squares-particle hydrodynamics - I. consistency and stability. *International Journal for Numerical Methods in Engineering*, **44**(8):1115–1155.
H. Q. Dinh, H. Q. Dinh, G. Turk, G. Turk, G. Slabaugh & G. Slabaugh [2001]. Reconstructing Surfaces Using Anisotropic Basis Functions. *International Conference on Computer Vision (ICCV)*, pp. 606–613.
P. Dirac [1958]. The principles of quantum mechanics, 4th edn. Clarendon.
C. A. Duarte & J. T. Oden [1996]. Hp clouds - an h-p meshless method. *Numerical methods for partial differential equations*, **12**(6):673–706.
O. Dubrule [1984]. Comparing splines and kriging. *Computers and Geosciences*, **10**(2-3):327–338.
M. Duchaineau, M. Wolinsky, D. E. Sigeti, M. C. Miller, C. Aldrich & M. B. Mineev-Weinstein [1997]. ROAMing terrain: Real-time optimally adapting meshes. In: *Visualization'97., Proceedings*, pp. 81–88. IEEE.
J. Duchon [1977]. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In: *Constructive theory of functions of several variables*, pp. 85–100. Springer.
J. Edwards [2017]. *Construction de modèles stratigraphiques à partir de données éparses*. Ph.D. thesis, Université de Lorraine.
V. Estellers, M. Scott & S. Soatto [2016]. Robust surface reconstruction. *SIAM Journal on Imaging Sciences*, **9**(4):2073–2098.
G. E. Fasshauer & J. G. Zhang [2007]. On choosing "optimal" shape parameters for RBF approximation. *Numerical Algorithms*, **45**(1-4):345–368.
R. P. Fedkiw, T. Aslam, B. Merriman & S. Osher [1999]. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of computational physics*, **152**(2):457–492.
H. Fossen [2016]. *Structural geology*. Cambridge University Press.
L. Fox, P. Henrici & C. Moler [1967]. Approximations and bounds for eigenvalues of elliptic operators. *SIAM Journal on Numerical Analysis*, **4**(1):89–102.
T. Frank, A.-L. Tertois & J.-L. Mallet [2007]. 3D-reconstruction of complex geological interfaces from irregularly distributed and noisy point data. *Computers & Geosciences*, **33**(7):932–943.
N. Fremming [2002]. 3D geological model construction using a 3D grid. In: *ECMOR VIII-8th European Conference on the Mathematics of Oil Recovery*.
T.-P. Fries & H.-G. Matthias [2004]. Classification and overview of meshfree methods. *Department of Mathematics and Computer Science, Technical Univ of Braunschweig*, p. 64.
Geovariances [2019]. Minestis. https://www.geovariances.com/en/software/minestis-mineral-resource-estimation/. Accessed 15.02.2019.
H. Gjøystdal, J. Reinhardsen & K. Åstebøl [1985]. Computer representation of complex 3-d geological structures using a new "solid modeling" technique. *Geophysical Prospecting*, **33**(8):1195–1211.
G. Godefroy [2018]. *Modélisation cinématique et stochastique des failles à partir de données éparses pour l'analyse des incertitudes structurales*. Ph.D. thesis, Université de Lorraine.
J. P. Gois, D. F. Trevisan, H. C. Batagelo, I. Macêdo, G. Diogo, F. Trevisan & H. Costa [2013]. Generalized hermitian radial basis functions implicits from polygonal mesh constraints. *The*

Visual Computer, **29**(6-8):651–661.

A. Gomes, I. Voiculescu, J. Jorge, B. Wyvill & C. Galbraith [2009]. *Implicit curves and surfaces: Mathematics, data structures and algorithms*. Springer Science & Business Media.

Í. G. Gonçalves, S. Kumaira & F. Guadagnin [2017]. A machine learning approach to the potential-field method for implicit modeling of geological structures. *Computers & Geosciences*, **103**:173–182.

L. Grose, G. Laurent, L. Aillères, R. Armit, M. Jessell & G. Caumon [2017]. Structural data constraints for implicit modeling of folds. *Journal of Structural Geology*, **104**:80–92.

Y. Gu & G. Liu [2001a]. A boundary point interpolation method (BPIM) using radial function basis. In: *First MIT Conference on Computational Fluid and Solid Mechanics, Boston, MA*, pp. 1590–1592.

Y. Gu & G. Liu [2002]. A hybrid boundary point interpolation method (HBPIM) and its coupling with EFG method. In: *Proceedings of the 1st Asian Workshop on Meshfree Methods*, vol. 16, p. 18.

Y. Gu & G.-R. Liu [2001b]. A meshless local Petrov-Galerkin (MLPG) method for free and forced vibration analyses for solids. *Computational Mechanics*, **27**(3):188–198.

G. Guennebaud & M. Gross [2007]. Algebraic point set surfaces. *ACM Transactions on Graphics*, **26**(3):23.

J. Guo, L. Wu, W. Zhou, J. Jiang & C. Li [2016]. Towards automatic and topologically consistent 3D regional geological modeling from boundaries and attitudes. *ISPRS International Journal of Geo-Information*, **5**(2):17.

W. Hackbusch [1999]. A sparse matrix arithmetic based on {H}-matrices. part I: Introduction to {H}-matrices. *Computing*, **62**(2):89–108.

M. J. Hillier, E. M. Schetselaar, E. A. de Kemp & G. Perron [2014]. Three-Dimensional Modelling of Geological Surfaces Using Generalized Interpolation with Radial Basis Functions. *Mathematical Geosciences*, **46**(8):931–953.

Ø. Hjelle & M. Dæhlen [2005]. Multilevel least squares approximation of scattered data over binary triangulations. *Computing and Visualization in Science*, **8**(2):83–91.

O. Hjelle & S. A. Petersen [2011]. A Hamilton-Jacobi framework for modeling folds in structural geology. *Mathematical Geosciences*, **43**(7):741–761.

Ø. Hjelle, S. A. Petersen & A. M. Bruaset [2013]. A numerical framework for modeling folds in structural geology. *Mathematical Geosciences*, **45**(3):255–276.

L. Holden, P. Mostad, B. F. Nielsen, J. Gjerde, C. Townsend & S. Ottesen [2003]. Stochastic structural modeling. *Mathematical geology*, **35**(8):899–914.

H. Hoppe, T. DeRose, T. Duchamp, J. McDonald & W. Stuetzle [1992]. *Surface reconstruction from unorganized points*, vol. 26. ACM.

M. R. Hudec & M. P. Jackson [2007]. Terra Infirma: Understanding salt tectonics. *Earth-Science Reviews*, **82**(1-2):1–28.

E. Husson [2013]. *Interaction géodynamique/karstification et modélisation géologique 3D des massifs carbonatés: Implication sur la distribution prévisionnelle de la karstification. Exemple des paléokarsts crétacés à néogènes du Languedoc Montpelliérain*. Ph.D. thesis, Université Montpellier 2.

S. R. Idelsohn & E. Onate [2006]. To mesh or not to mesh. that is the question.... *Computer methods in applied mechanics and engineering*, **195**(37-40):4681–4696.

S. R. Idelsohn, E. Onate, N. Calvo & F. Del Pin [2003]. The meshless finite element method. *International Journal for Numerical Methods in Engineering*, **58**(6):893–912.

Intrepid-Geophysics [2018]. Geomodeller. http://www.intrepid-geophysics.com/ig/index.php?page=geomodeller. Accessed 24.10.2018.

M. Irakarama, G. Laurent, J. Renaudeau & G. Caumon [2018a]. Finite difference implicit modeling of geological structures. In: *80th EAGE Conference and Exhibition 2018*.

M. Irakarama, G. Laurent, J. Renaudeau & G. Caumon [2018b]. Finite difference implicit structural modeling of geological structures. In: *2018 Ring Meeting*. ASGA.

C. Jacquemyn, M. D. Jackson & G. J. Hampson [2018]. Surface-based geological reservoir modelling using grid-free NURBS curves and surfaces. *Mathematical Geosciences*, pp. 1–28.

M. Jessell, L. Aillères, E. D. Kemp, M. Lindsay, F. Wellmann, M. Hillier, G. Laurent, T. Carmichael & R. Martin [2014]. Next Generation Three-Dimensional Geologic Modeling and Inversion. *SEG Special Publication 18*, pp. 261–272.

M. Karimi-Fard & L. J. Durlofsky [2016]. A general gridding, discretization, and coarsening methodology for modeling flow in porous formations with discrete geological features. *Advances in water resources*, **96**:354–372.

J. O. Kaven, R. Mazzeo & D. D. Pollard [2009]. Constraining surface interpolations using elastic plate bending solutions with applications to geologic folding. *Mathematical Geosciences*, **41**(1):1.

M. Kazhdan [2005]. Reconstruction of solid models from oriented point sets. In: *Proceedings of the third Eurographics symposium on Geometry processing*, p. 73. Eurographics Association.

M. Kazhdan, M. Bolitho & H. Hoppe [2006]. Poisson surface reconstruction. *Eurographics Symposium on Geometry Processing*, pp. 61–70.

M. Kazhdan & H. Hoppe [2013]. Screened Poisson surface reconstruction. *ACM Transactions on Graphics*, **32**(3):1–13.

H. Khazal, H. Bayesteh, S. Mohammadi, S. S. Ghorashi & A. Ahmed [2016]. An extended Element Free Galerkin method for fracture analysis of functionally graded materials. *Mechanics of Advanced Materials and Structures*, **23**(5):513–528.

R. H. Knight, R. G. Lane, H. J. Ross, a. P. G. Abraham & J. Cowan [2007]. Implicit ore delineation. *Proceedings of Exploration 07: Fifth Decennial International Conference on Mineral Exploration*, pp. 1165–1169.

R. Kress, A. Mohsen & B. Brosowski [1986]. On the simulation source technique for exterior problems in acoustics. *Mathematical methods in the applied sciences*, **8**(1):585–597.

P. Krysl & T. Belytschko [1999]. The element free Galerkin method for dynamic propagation of arbitrary 3-D cracks. *International Journal for Numerical Methods in Engineering*, **44**(6):767–800.

V. D. Kupradze [1964]. A method for the approximate solution of limiting problems in mathematical physics. *USSR Computational Mathematics and Mathematical Physics*, **4**(6):199–205.

E. Labrunye [2004]. *Extraction automatique d'information géologique à partir d'images sismiques tridimensionnelles*. Ph.D. thesis, Vandoeuvre-les-Nancy, INPL.

C. Lajaunie, G. Courrioux & L. Manuel [1997]. Foliation fields and 3D cartography in geology: Principles of a method based on potential interpolation. *Mathematical Geology*, **29**(4):571–584.

F. Lallier [2012]. *Corrélation stratigraphique stochastique de puits*. Ph.D. thesis, Université de Lorraine.

P. Lancaster & K. Salkauskas [1981]. Surfaces generated by moving least squares methods. *Mathematics of computation*, **37**(155):141–158.

G. Laurent [2016]. Iterative thickness regularization of stratigraphic layers in discrete implicit modeling. *Mathematical Geosciences*, pp. 1–23.

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

D. Ledez [2003]. *Modélisation d'objets naturels par formulation implicite*. Ph.D. thesis, Vandoeuvre-les-Nancy, INPL.

A. M. Lemon & N. L. Jones [2003]. Building solid models from boreholes and user-defined cross-sections. *Computers & Geosciences*, **29**(5):547–555.

B. Levy & J.-L. Mallet [1999]. Discrete Smooth Interpolation: Constrained discrete fairing for arbitrary meshes. *GOCAD Consortium*.

M. D. Lindsay, M. W. Jessell, L. Ailleres, S. Perrouty, E. de Kemp & P. G. Betts [2013]. Geodiversity: Exploration of 3D geological model space. *Tectonophysics*, **594**(April 2016):27–37.

T. Liszka, C. Duarte & W. Tworzydlo [1996]. hp-Meshless cloud method. *Computer Methods in Applied Mechanics and Engineering*, **139**(1-4):263–288.

G. Liu & Y. Gu [2001a]. A local point interpolation method for stress analysis of two-dimensional solids. *Structural Engineering and Mechanics*, **11**(2):221–236.

G. Liu & Y. Gu [2001b]. A local radial point interpolation method (lrpim) for free vibration analyses of 2-d solids. *Journal of Sound and vibration*, **246**(1):29–46.

G. Liu & Y. Gu [2003]. A meshfree method: Meshfree weak–strong (MWS) form method, for 2-D solids. *Computational Mechanics*, **33**(1):2–14.

G.-R. Liu & Y. Gu [2001c]. A point interpolation method for two-dimensional solids. *International Journal for Numerical Methods in Engineering*, **50**(4):937–951.

G. R. Liu & Y. Gu [2005]. *An introduction to meshfree method and their programming*. Springer.

W. K. Liu, Y. Chen, R. A. Uras & C. T. Chang [1996]. Generalized multiple scale reproducing kernel particle methods. *Computer Methods in Applied Mechanics and Engineering*, **139**(1-4):91–

157.

W. K. Liu, S. Jun & Y. F. Zhang [1995]. Reproducing kernel particle methods. *International journal for numerical methods in fluids*, **20**(8-9):1081–1106.

W. E. Lorensen & H. E. Cline [1987]. Marching cubes: A high resolution 3D surface construction algorithm. In: *ACM siggraph computer graphics*, vol. 21, pp. 163–169. ACM.

L. B. Lucy [1977]. A numerical approach to the testing of the fission hypothesis. *The astronomical journal*, **82**:1013–1024.

I. Macêdo, J. P. Gois & L. Velho [2009]. Hermite interpolation of implicit surfaces with radial basis functions. *Proceedings of SIBGRAPI 2009 - 22nd Brazilian Symposium on Computer Graphics and Image Processing*, pp. 1–8.

F. Maerten [2010]. *Geomechanics to solve geological structure issues: Forward, inverse and restoration modeling*. Ph.D. thesis, Université Montpellier 2.

F. Maerten [2018]. Meshless representation of a geologic environment. *Owner: Schlumberger*, (US Patent 10088596).

L. Maerten & F. Maerten [2006]. Chronologic modeling of faulted and fractured reservoirs using geomechanically based restoration: Technique and industry applications. *AAPG bulletin*, **90**(8):1201–1226.

J. L. Mallet [1988]. Three-dimensional graphic display of disconnected bodies. *Mathematical Geology*, **20**(8):977–990.

J.-L. Mallet [1989]. Discrete smooth interpolation. *ACM Transactions on Graphics (TOG)*, **8**(2):121–144.

J. L. Mallet [1992]. Discrete smooth interpolation. *Computer-aided Design*, **24**(4):178–191.

J. L. Mallet [1997]. Discrete modeling for natural objects. *Mathematical Geology*, **29**(2):199–219.

J.-L. Mallet [2002]. *Geomodeling*. Oxford University Press, Inc.

J.-L. Mallet [2014]. *Elements of mathematical sedimentary geology: The GeoChron model*. EAGE publications.

J. G. Manchuk & C. V. Deutsch [2019]. Boundary modeling with moving least squares. *Computers & Geosciences*.

J. Manson, G. Petrova & S. Schaefer [2008]. Streaming surface reconstruction using wavelets. *Computer Graphics Forum*, **27**(5):1411–1420.

M. Marin, C. Castagnac, O. Ferrer & J. Renaudeau [2018]. Physical models as analogues for basin and petroleum systems modeling: Example of temperature and pressure estimations around salt bodies. In: *GeoMod2018*.

R. Martin & J. B. Boisvert [2017]. Iterative refinement of implicit boundary models for improved geological feature reproduction. *Computers & Geosciences*, **109**:1–15.

C. Massiot & G. Caumon [2010]. Accounting for axial directions, cleavages and folding style during 3D structural modeling. In: *Proc. 30th Gocad Meeting, Nancy*.

G. Matheron [1981]. Splines and kriging: Their formal equivalence. *Down-to-Earth Statistics: Solutions Looking for Geological Problems*, **8**:77–95.

M. Maxelon [2004]. *Developing a three-Dimensional structural model of the lower lepontine nappes – Central Alps, Switzerland and Northern Italy*. Ph.D. thesis, ETH Zurich.

A. Mazuyer [2018]. *Estimation de l'état de contrainte initial in situ dans les réservoirs par approche inverse*. Ph.D. thesis, Université de Lorraine.

D. H. McLain [1976]. Two dimensional interpolation from random data. *The Computer Journal*, **19**(2):178–181.

B. Mederos, M. Lage, S. Arouca, F. Petronetto, L. Velho, T. Lewiner & H. Lopes [2007]. Regularized implicit surface reconstruction from points and normals. *Journal of the Brazilian Computer Society*, **13**(4):7–16.

J. Melenk & I. Babuška [1996]. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, **139**(1-4):289–314.

U. T. Mello & M. E. Henderson [1997]. Techniques for including large deformations associated with salt and fault motion in basin modeling. *Marine and Petroleum Geology*, **14**(5):551–564.

N. Moës & T. Belytschko [2002]. Extended finite element method for cohesive crack growth. *Engineering fracture mechanics*, **69**(7):813–833.

N. Moës, J. Dolbow & T. Belytschko [1999]. A finite element method for crack growth without remeshing. *Int J Numer Methods Eng*, **46**(February):131–150.

M. Mongillo [2011]. Choosing basis functions and shape parameters for radial basis function methods. *SIAM Undergraduate Research Online*, **4**:190–209.

G. Morra, P. Chatelain, P. Tackley & P. Koumoutsakos [2007]. Large scale three-dimensional boundary element simulation of subduction. In: *International Conference on Computational Science*, pp. 1122–1129. Springer.

B. Morse, T. Yoo, P. Rheingans, D. Chen & K. Subramanian [2001]. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. *Proceedings International Conference on Shape Modeling and Applications*, pp. 89–98.

Y. X. Mukherjee & S. Mukherjee [1997]. The boundary node method for potential problems. *International Journal for Numerical Methods in Engineering*, **40**(5):797–815.

M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross & M. Alexa [2004]. Point based animation of elastic, plastic and melting objects. In: *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 141–151. Eurographics Association.

S. Muraki [1991]. Volumetric shape description of range data using "blobby model". *ACM SIGGRAPH computer graphics*, **25**(4):227–235.

Y. Nagai, Y. Ohtake & H. Suzuki [2009]. Smoothing of partition of unity implicit surfaces for noise robust surface reconstruction. *Eurographics Symposium on Geometry Processing*, **28**(5):1339–1348.

B. Nayroles, G. Touzot & P. Villon [1992]. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational mechanics*, **10**(5):307–318.

V. P. Nguyen, T. Rabczuk, S. Bordas & M. Duflot [2008]. Meshless methods : A review and computer implementation aspects. *Mathematics and computers in simulation*, **79**(3):763–813.

H. Nishimura [1985]. Object modeling by distribution function and a method of image generation. *Trans Inst Electron Commun Eng Japan*, **68**:718.

J. Nocedal & S. J. Wright [2006]. *Numerical optimization 2nd*. Springer.

Y. Ohtake, A. Belyaev & H. Seidel [2003]. A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions. *Shape Modeling International*, pp. 153–161.

Y. Ohtake, A. Belyaev & H.-P. Seidel [2006]. Sparse surface reconstruction with adaptive partition of unity and radial basis functions. *Graphical Models*, **68**(1):15–24.

E. Onate, S. Idelsohn, O. Zienkiewicz & R. Taylor [1996a]. A finite point method in computational mechanics. applications to convective transport and fluid flow. *International journal for numerical methods in engineering*, **39**(22):3839–3866.

E. Onate, S. Idelsohn, O. Zienkiewicz, R. Taylor & C. Sacco [1996b]. A stabilized finite point method for analysis of fluid mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, **139**(1):315–346.

D. Organ, M. Fleming, T. Terry & T. Belytschko [1996]. Continuous meshless approximations for nonconvex bodies by diffraction and transparency. *Computational mechanics*, **18**(3):225–235.

S. Osher & J. A. Sethian [1988]. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of computational physics*, **79**(1):12–49.

Paradigm [2019]. Skua-gocad. http://www.pdgm.com/products/skua-gocad/. Accessed 15.02.2019.

M. N. Parquer, P. Collon & G. Caumon [2017]. Reconstruction of channelized systems through a conditioned reverse migration method. *Mathematical Geosciences*, **49**(8):965–994.

J. Pellerin, B. Lévy, G. Caumon & A. Botella [2014]. Automatic surface remeshing of 3D structural models at specified resolution: A method based on Voronoi diagrams. *Computers & Geosciences*, **62**:103–116.

M. Philippon, C. L. C. De Veslud, F. Gueydan, J.-P. Brun & G. Caumon [2015]. 3D geometrical modelling of post-foliation deformations in metamorphic terrains (Syros, Cyclades, Greece). *Journal of Structural Geology*, **78**:134–148.

L. Piegl & W. Tiller [1997]. The NURBS book. 1997. *Monographs in Visual Communication*.

F. Pigeonneau [2011]. Simulation numérique à l'échelle macroscopique par la méthode des éléments finis.

J. Pouderoux, J.-C. Gonzato, I. Tobor & P. Guitton [2004]. Adaptive hierarchical rbf interpolation for creating smooth digital elevation models. In: *Proceedings of the 12th annual ACM international workshop on Geographic information systems*, pp. 232–240. ACM.

A. Poullikkas, A. Karageorghis & G. Georgiou [2002]. The method of fundamental solutions for three-dimensional elastostatics problems. *Computers & structures*, **80**(3):365–370.

M. J. Pyrcz, O. Catuneanu & C. V. Deutsch [2005]. Stochastic surface-based modeling of

turbidite lobes. *AAPG bulletin*, **89**(2):177–191.

J. G. Ramsay [1967]. *Folding and fracturing of rocks*. McGraw-Hill, New York/London.

J. Renaudeau, M. Irakarama, G. Laurent, F. Maerten & G. Caumon [2019a]. Implicit modelling of geological structures: A Cartesian grid method handling discontinuities with ghost points. *Boundary Elements and other Mesh Reduction Methods XXXXI*, **122**:189.

J. Renaudeau, F. Maerten & E. Malvesin [2018]. Geologic structural model generation. *Owner: Schlumberger*, (US Patent 20180347320).

J. Renaudeau, E. Malvesin, F. Maerten & G. Caumon [2019b]. Implicit structural modeling by minimization of the bending energy with moving least squares functions. *Mathematical Geosciences*. Accepted paper.

M. Rouhani, A. D. Sappa & E. Boyer [2015]. Implicit B-spline surface reconstruction. *IEEE Transactions on Image Processing*, **24**(1):22–32.

J. Ruiu [2015]. *Modélisation d'objets sédimentaires par des surfaces paramétriques et application à l'analyse d'image*. Ph.D. thesis, Université de Lorraine.

J. Ruiu, G. Caumon & S. Viseur [2016]. Modeling channel forms and related sedimentary objects using a boundary representation based on non-uniform rational b-splines. *Mathematical Geosciences*, **48**(3):259–284.

O. Schall, M. Samozino, B. Falcidieno & N. Thalmann [2005]. Surface from scattered points a brief survey of recent developments. In: *1st International Workshop towards Semantic Virtual Environments*, pp. 138–147. MIRALab.

M. Scheuerer, R. Schaback & M. Schlather [2013]. Interpolation of spatial data–a stochastic or a deterministic problem? *European Journal of Applied Mathematics*, **24**(4):601–629.

Schlumberger [2010]. *Petrel 2010: Structural modeling course*.

Schlumberger [2018]. Petrel. https://www.software.slb.com/products/petrel. Accessed 15.02.2019.

Schlumberger [2019]. Petromod. https://www.software.slb.com/products/petromod. Accessed 15.02.2019.

C. Shen, J. F. O'Brien & J. R. Shewchuk [2004]. Interpolating and approximating implicit surfaces from polygon soup. In: *ACM Transactions on Graphics*, vol. 23, p. 896. ACM.

J. Shewchuk [2002]. What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures (preprint). *University of California at Berkeley*, **73**:137.

R. Sibson [1980]. A vector identity for the Dirichlet tessellation. In: *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 87, pp. 151–155. Cambridge University Press.

J. Sladek, P. Stanak, Z. Han, V. Sladek & S. Atluri [2013]. Applications of the MLPG method in engineering & sciences: A review. *Comput Model Eng Sci*, **92**:423–475.

J. Slak & G. KOSEC [2019]. Fast generation of variable density node distributions for mesh-free methods. *Boundary Elements and other Mesh Reduction Methods XXXXI*, **122**:163.

L. Souche, G. Iskenova, F. Lepage, D. Desmarest *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*.

K. B. Sprague & E. A. De Kemp [2005]. Interpretive tools for 3-D structural geological modelling. Part II: Surface design from sparse spatial data. *GeoInformatica*, **9**(1):5–32.

T. Strouboulis, I. Babuška & K. Copps [2000]. The design and analysis of the generalized finite element method. *Computer methods in applied mechanics and engineering*, **181**(1):43–69.

N. Sukumar [1998]. *The natural element method in solid mechanics*. Ph.D. thesis, Northwestern University.

C. H. Sword Jr [1991]. Building flexible interactive geologic models. In: *SEG Technical Program Expanded Abstracts 1991*, pp. 1465–1467. Society of Exploration Geophysicists.

A. Tabarraei & N. Sukumar [2007]. Adaptive computations using material forces and residual-based error estimators on quadtree meshes. *Computer Methods in Applied Mechanics and Engineering*, **196**(25-28):2657–2680.

A.-L. Tertois [2007]. *Création et modification de modèles géologiques par champs de potentiel . Application au modèle GeoChron*. Ph.D. thesis, Vandoeuvre-les-Nancy, INPL.

I. Tobor, P. Reuter & C. Schlick [2004]. Efficient reconstruction of large scattered geometric datasets using the partition of unity and radial basis functions. Tech. rep., UNION Agency.

G. Turk & J. F. O'brien [1999]. Variational implicit surfaces. Tech. rep., Georgia Institute of Technology.

S. A. Vollgger, A. R. Cruden, L. Ailleres & E. J. Cowan [2015]. Regional dome evolution and its control on ore-grade distribution: Insights from 3D implicit modelling of the Navachab gold deposit, Namibia. *Ore Geology Reviews*, **69**:268–284.

G. Wahba [1990]. *Spline models for observational data*, vol. 59. Society for Industrial and Applied Mathematics.

J. Wang & G. Liu [2002]. A point interpolation meshless method based on radial basis functions. *International Journal for Numerical Methods in Engineering*, **54**(11):1623–1648.

J. F. Wellmann, F. G. Horowitz & K. Regenauer-Lieb [2011]. Towards a quantification of uncertainties in 3-d geological models. In: *IAMG 2011 Conference "Mathematical geosciences at the crossroads of theory and practice", Salzburg, Austria*, pp. 745–758.

J. F. Wellmann, F. G. Horowitz, E. Schill & K. Regenauer-Lieb [2010]. Towards incorporating uncertainty of structural data in 3D geological inversion. *Tectonophysics*, **490**(3-4):141–151.

H. Wendland [1995]. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, **4**(1):389–396.

H. Wendland [2004]. Solving large generalized interpolation problems efficiently. *Advances in Constructive Approximation (M Neamtu and EB Saff, eds), Nashboro Press, Brentwood, TN*, pp. 509–518.

P. Wessel & D. Bercovici [1998]. Interpolation with splines in tension: A Green's function approach. *Mathematical Geology*, **30**(1):77–93.

X. Wu, M. Yu & W. Q. Xia [2005a]. Implicit fitting and smoothing using radial basis functions with partition of unity. In: *Computer Aided Design and Computer Graphics, 2005. Ninth International Conference on*, pp. 10–pp. IEEE.

Y. Wu, G. Liu & Y. Gu [2005b]. Application of meshless local Petrov-Galerkin (MLPG) approach to simulation of incompressible flow. *Numerical Heat Transfer, Part B: Fundamentals*, **48**(5):459–475.

Z. Wu [1992]. Hermite-birkhoff interpolation of scattered data by radial basis functions. *Approximation Theory and its Applications*, **8**(2):1–10.

G. Wyvill, C. McPheeters & B. Wyvill [1986]. Soft objects. In: *Advanced Computer Graphics*, pp. 113–128. Springer.

A. Yavari, A. Kaveh, S. Sarkani & H. A. R. Bondarabady [2001]. Topological aspects of meshless methods and nodal ordering for meshless discretizations. *International Journal for Numerical Methods in Engineering*, **52**(9):921–938.

R. Yokota, L. A. Barba & M. G. Knepley [2010]. Petrbf - a parallel o(n) algorithm for radial basis function interpolation with Gaussians. *Computer Methods in Applied Mechanics and Engineering*, **199**(25):1793–1804.

X. Zhang, X.-H. Liu, K.-Z. Song & M.-W. Lu [2001]. Least-squares collocation meshless method. *International Journal for Numerical Methods in Engineering*, **51**(9):1089–1100.

T. Zhu, J. Zhang & S. Atluri [1998]. A meshless local boundary integral equation (LBIE) method for solving nonlinear problems. *Computational Mechanics*, **22**(2):174–186.

Formulation continue du problème de modélisation implicite de structures géologiques discrétisée avec des méthodes de réduction de maillage

Résumé : La modélisation structurale consiste à approximer les structures géologiques du sous-sol en un modèle numérique afin d'en visualiser la géométrie et d'y effectuer des calculs d'estimation et de prédiction. L'approche implicite de la modélisation structurale utilise des données de terrain interprétées pour construire une fonction volumétrique sur le domaine d'étude qui représente la géologie. Cette fonction doit honorer les observations, interpoler entre ces dernières, et extrapoler dans les zones sous-échantillonnées tout en respectant les concepts géologiques. Les méthodes actuelles portent cette interpolation soit sur les données, soit sur un maillage. Ensuite, le problème de modélisation est posé selon la discrétisation choisie : par krigeage dual sur les points de donnée ou en définissant un critère de rugosité sur les éléments du maillage. Dans cette thèse, nous proposons une formulation continue de la modélisation structurale par méthodes implicites. Cette dernière consiste à minimiser une somme de fonctionnelles arbitraires. Les contraintes de donnée sont imposées avec des fonctionnelles discrètes, et l'interpolation est contrôlée par des fonctionnelles continues. Cette approche permet de (i) développer des liens entre les méthodes existantes, (ii) suggérer de nouvelles discrétisations d'un même problème de modélisation, et (iii) modifier le problème de modélisation pour mieux honorer certains cas géologiques sans dépendre de la discrétisation. Nous portons également une attention particulière à la gestion des discontinuités telles que les failles et les discordances. Les méthodes existantes nécessitent soit la création de zones volumétriques avec des géométries complexes, soit la génération d'un maillage volumétrique dont les éléments sont conformes aux surfaces de discontinuité. Nous montrons, en explorant des méthodes sans maillage locales et des concepts de réduction de maillage, qu'il est possible d'assurer l'interpolation des structures tout en réduisant les contraintes liées à la gestion des discontinuités. Deux discrétisations de notre problème de minimisation sont suggérées : l'une utilise les moindres carrés glissants avec des critères optiques pour la gestion des discontinuités, et l'autre utilise des fonctions issues de la méthode des éléments finis avec le concept de nœuds fantômes pour les discontinuités. Une étude de sensibilité et une comparaison des deux méthodes sont proposées en 2D, ainsi que quelques exemples en 3D. Les méthodes développées dans cette thèse ont un grand impact en termes d'efficacité numérique et de gestion de cas géologiques complexes. Par exemple, il est montré que notre problème de minimisation au sens large apporte plusieurs solutions pour la gestion de cas de plis sous-échantillonnés et de variations d'épaisseur dans les couches stratigraphiques. D'autres applications sont également présentées tels que la modélisation d'enveloppe de sel et la restauration mécanique.

Mots-clés : Méthodes numériques, Modélisation implicite, Équations continues, Méthodes sans maillage

---

Continuous formulation of implicit structural modeling discretized with mesh reduction methods

Abstract: Implicit structural modeling consists in approximating geological structures into a numerical model for visualization, estimations, and predictions. It uses numerical data interpreted from the field to construct a volumetric function on the domain of study that represents the geology. The function must fit the observations, interpolate in between, and extrapolate where data are missing while honoring the geological concepts. Current methods support this interpolation either with the data themselves or using a mesh. Then, the modeling problem is posed depending on these discretizations: performing a dual kriging between data points or defining a roughness criterion on the mesh elements. In this thesis, we propose a continuous formulation of implicit structural modeling as a minimization of a sum of generic functionals. The data constraints are enforced by discrete functionals, and the interpolation is controlled by continuous functionals. This approach enables to (i) develop links between the existing methods, (ii) suggest new discretizations of the same modeling problem, and (iii) modify the minimization problem to fit specific geological issues without any dependency on the discretization. Another focus of this thesis is the efficient handling of discontinuities, such as faults and unconformities. Existing methods require either to define volumetric zones with complex geometries, or to mesh volumes with conformal elements to the discontinuity surfaces. We show, by investigating local meshless functions and mesh reduction concepts, that it is possible to reduce the constraints related to the discontinuities while performing the interpolation. Two discretizations of the minimization problem are then suggested: one using the moving least squares functions with optic criteria to handle discontinuities, and the other using the finite element method functions with the concept of ghost nodes for the discontinuities. A sensitivity analysis and a comparison study of both methods are performed in 2D, with some examples in 3D. The developed methods in this thesis prove to have a great impact on computational efficiency and on handling complex geological settings. For instance, it is shown that the minimization problem provides the means to manage under-sampled fold structures and thickness variations in the layers. Other applications are also presented such as salt envelope surface modeling and mechanical restoration.

Keywords: Numerical methods, Implicit modeling, Continuous equations, Meshless methods