



**HAL**  
open science

# Étude d'algorithmes de restauration d'images sismiques par optimisation de forme non linéaire et application à la reconstruction sédimentaire

Mathieu Gilardet

► **To cite this version:**

Mathieu Gilardet. Étude d'algorithmes de restauration d'images sismiques par optimisation de forme non linéaire et application à la reconstruction sédimentaire. Algorithme et structure de données [cs.DS]. Université de Pau et des Pays de l'Adour, 2013. Français. NNT : . tel-00952964

**HAL Id: tel-00952964**

**<https://theses.hal.science/tel-00952964>**

Submitted on 28 Feb 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Étude d'algorithmes de restauration d'images sismiques par optimisation de forme non linéaire et application à la reconstruction sédimentaire

## THÈSE

présentée et soutenue publiquement le 19 décembre 2013

pour l'obtention du

Doctorat de l'Université de Pau et des Pays de l'Adour  
(spécialité informatique)

par

Mathieu GILARDET

### Composition du Jury :

<i>Rapporteurs :</i>	Yannick Berthoumieu	Professeur Institut Polytechnique ENSEIRB-Matméca - Bordeaux
	Christian Gout	Professeur Institut National des Sciences Appliquées - Rouen
<i>Examineur :</i>	Roland Martin	Ingénieur de Recherche HDR Observatoire Midi Pyrénées - Toulouse
<i>Directeur de thèse :</i>	Dimitri Komatitsch	Directeur de Recherche CNRS - Marseille
<i>Encadrants de thèse :</i>	Sébastien Guillon	Ingénieur TOTAL S.A. - Pau
	Bruno Jobard	Maître de Conférences Université de Pau et des Pays de l'Adour - Pau



# Remerciements

Je tiens d'abord à remercier Nameen Keskes, pour m'avoir permis d'intégrer l'équipe SISMAGE et d'aborder par cette occasion la discipline passionnante qu'est la recherche.

Je remercie ensuite Sebastien Guillon qui durant ces trois dernières années, m'a initié à des sujets nouveaux et enrichissants, en orientant mon cheminement avec patience, exigence et justesse. Cette expérience m'a permis de mûrir ma réflexion et donné l'envie de continuer à explorer les sciences.

Je remercie également Dimitri Komatitsch pour m'avoir à travers de nombreuses remarques, amené à me poser les bonnes questions, approfondir et affiner ma réflexion, finalement prendre le recul nécessaire au quotidien de tout chercheur.

Je remercie enfin Bruno Jobard de m'avoir introduit à SISMAGE à travers mon stage de Master 2, ce qui aura permis cet aboutissement au delà de mes espérances initiales.

Je remercie ensuite Roland Martin et Yannick Bertoumieu pour avoir accepté d'être rapporteurs du manuscrit (malgré un calendrier serré), ainsi que Roland Martin pour leur participation au jury de soutenance. L'ensemble de leurs remarques constructives ont témoigné de l'intérêt porté à ce travail.

Je ne peux oublier Anne-Gaelle et Stephano, qui ont beaucoup œuvré (sans compter les heures) pour me faire progresser en mathématiques, et à qui je dois aujourd'hui d'avoir un regard plus serein sur la discipline.

Une pensée particulière pour tous mes proches et amis, avec qui j'ai partagé tant de moments ces dernières années, et qui ont su me sortir le nez du guidon lorsque j'en avais besoin. Je garderais un souvenir particulier des apéros du jeudi soir, de nos arsouilles sur routes sinueuses, ou encore de quelques voyages, et vous souhaite à tous le meilleur pour l'avenir.

Enfin, tellement de producteurs de musique ont contribué de manière indirecte à l'aboutissement de cette thèse, que je ne saurais les omettre de ce préambule. J'adresse donc une pensée particulière aux artistes de *Warp Records*, *Ad Noiseam*, *Rephlex Records*, *Planet  $\mu$*  ou encore *Jarring Effects*.





*à Elvire.*



# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 Présentation de la restauration sismique</b>	<b>5</b>
1.1 Caractéristiques principales des images sismiques	5
1.2 Interprétation d'images sismiques	7
1.3 La restauration : un outil précieux d'aide à l'interprétation	7
1.4 Description succincte du processus de restauration	10
1.4.1 Restauration des failles	11
1.4.2 Mise à plat d'un horizon	12
1.5 État de l'art de la restauration	12
1.5.1 Méthodes de restauration géométriques	12
1.5.2 Méthodes de restauration physiques	14
1.6 Création d'un nouvel outil de restauration	15
<b>2 Déformation de modèles numériques</b>	<b>17</b>
2.1 Introduction résumée aux techniques de déformation	17
2.2 Formalisme général d'une déformation de modèle numérique	22
2.2.1 Choix des points de contraintes	24
2.2.1.1 Déformation directe du modèle numérique $\mathcal{P}$	25
2.2.1.2 Déformation de l'espace $\mathcal{S}$ englobant le modèle numérique $\mathcal{P}$	27
2.2.1.3 Applications qui déforment le modèle $\mathcal{P}$ par un squelette	30
2.2.2 Calcul du déplacement à tous les points de contraintes	31
2.2.2.1 Déformation sous contraintes géométriques	32
2.2.2.2 Déformation sous contraintes physiques	33
2.2.3 Interpolation de la déformation pour l'ensemble du domaine	33
2.2.3.1 Interpolants locaux	34

2.2.3.2	Interpolants globaux . . . . .	37
2.3	Choix de nos solutions . . . . .	38
2.3.1	Choix d'un ensemble de points de contraintes . . . . .	38
2.3.2	Estimation des déplacements . . . . .	39
2.3.3	Interpolation . . . . .	40
<b>3</b>	<b>Présentation de notre algorithme de restauration</b> _____	<b>41</b>
3.1	Processus de restauration . . . . .	42
3.1.1	Restauration d'une faille . . . . .	43
3.1.2	Mise à plat de l'horizon . . . . .	44
3.2	Écriture des contraintes de déformation . . . . .	46
3.2.1	Introduction des contraintes de restauration . . . . .	46
3.2.2	Scénario de restauration de la faille . . . . .	48
3.2.2.1	Contrainte de continuité de l'horizon . . . . .	48
3.2.2.2	Contrainte de contact à la faille . . . . .	49
3.2.3	Scénario de mise à plat de l'horizon . . . . .	52
3.2.4	Introduction des contraintes géométriques . . . . .	53
3.2.4.1	Préservation des formes . . . . .	53
3.2.4.2	Préservation des longueurs d'arêtes . . . . .	63
3.3	Optimisation des contraintes de déformation . . . . .	65
3.3.1	Optimisation par minimisation pondérée des contraintes . . . . .	66
3.3.1.1	Formulation du système d'équations . . . . .	66
3.3.1.2	Résolution du système non-linéaire par la méthode itérative de <i>Gauss-Newton</i> . . . . .	67
3.3.1.3	Pondération des différentes contraintes . . . . .	69
3.3.2	Optimisation avec respect strict de contraintes . . . . .	70
3.3.2.1	Formulation des contraintes imposées . . . . .	71
3.3.2.2	Généralisation des notations . . . . .	73
3.3.2.3	Formulation du système d'équations . . . . .	75
3.3.2.4	Résolution du système d'équation . . . . .	75
3.4	Comparaison et choix d'une méthode d'optimisation . . . . .	78
3.5	Pré-conditionnement du problème de restauration . . . . .	80
3.5.1	Pré-conditionnement d'un algorithme de minimisation . . . . .	81
3.5.2	Observation de convergence vers des minima locaux lors de la restauration de failles . . . . .	81

3.5.3	Pré-conditionnement de l'algorithme d'optimisation pour la restauration de la faille . . . . .	84
3.6	Convergence et critère d'arrêt du processus itératif d'optimisation . . . . .	85
3.6.1	Étude de la convergence pour le scénario typique de restauration d'une faille . . . . .	85
3.6.2	Étude de la convergence pour le scénario de mise à plat de l'horizon . . . . .	88
3.6.3	Analyse plus précise de ces résultats . . . . .	90
3.6.4	Conditions d'arrêt automatique . . . . .	90
3.6.5	Étude de la convergence pour le scénario complet de restauration, avec critère d'arrêt automatique . . . . .	91
3.7	Impact du maillage sur la déformation et choix d'un maillage optimisé . . . . .	93
3.7.1	Cas d'application avec deux maillages impactant différemment la déformation . . . . .	94
3.7.2	Causes de l'impact du choix du maillage sur la déformation . . . . .	96
3.7.3	Limite d'utilisation d'un maillage optimal . . . . .	97
3.7.4	Méthode permettant de déterminer un maillage "optimisé" acceptable . . . . .	98
3.7.4.1	Analyse de l'impact du raffinement . . . . .	98
3.7.4.2	Choix d'un maillage optimisé . . . . .	102
<b>4</b>	<b>Processus de restauration et outils dérivés</b> _____	<b>105</b>
4.1	Processus de restauration . . . . .	105
4.2	Outils dérivés . . . . .	107
4.2.1	Correspondance des pointés entre l'espace initial et l'espace restauré . . . . .	107
4.2.1.1	Fonctions de correspondance d'un espace à un autre . . . . .	108
4.2.1.2	Pointé d'horizons dans l'espace restauré . . . . .	110
4.2.2	Restauration multi-failles et multi-horizons . . . . .	111
4.2.2.1	Restauration d'une section multi-faillée . . . . .	111
4.2.2.2	Restauration d'une section contenant plusieurs horizons . . . . .	114
4.3	Conclusions . . . . .	118
	<b>Conclusions</b> _____	<b>119</b>
	<b>Bibliographie</b> _____	<b>123</b>



# Introduction

L'exploration pétrolière est un processus composé de plusieurs étapes majeures, dont le but est d'acquérir une connaissance solide du sous-sol. La première étape consiste à pouvoir représenter ce que l'on ne peut observer et l'on procède alors à une acquisition sismique pour obtenir une "photographie" de la région étudiée. On procède dans ce but à l'émission d'ondes artificielles (de nature sismiques ou acoustiques), dont les réflexions sur les différentes couches géologiques sont écoutées à la surface (tout comme lors d'une échographie), ce qui permet d'obtenir un premier enregistrement du sous-sol. Cet enregistrement est ensuite traité en vue de créer une représentation compréhensible. Typiquement, il permet d'obtenir des images sismiques bidimensionnelles (coupes verticales appelées sections sismiques) ou tridimensionnelles (volumes sismiques appelés cubes) qui illustrent l'organisation des différentes couches.

Une fois obtenues, ces images sont interprétées par des géophysiciens pour déduire la morphologie du sous-sol et reconstituer son histoire, tout cela dans le but de déterminer la probabilité de présence de réservoirs à hydrocarbures. Ce travail consiste à détecter différents objets géologiques, comme les "horizons" qui symbolisent l'interface entre deux couches géologiques de nature différentes, ou les "failles" qui correspondent à des fractures de la roche composant ces couches. On extrait ainsi la structure du sous-sol qui est en fait une représentation simplifiée de l'imagerie sismique considérée. Cette structure permet ensuite de formuler des hypothèses sur l'histoire du sous-sol. Si des fractures sont observées, on doit être en mesure de décrire les différents phénomènes tectoniques qui ont conduit à leur formation. En ce sens, outre l'extraction de la structure du sous-sol, le travail d'interprétation consiste à formuler des hypothèses décrivant les différentes déformations qui se sont produites au cours du temps dans le sous-sol.

Aujourd'hui, l'ensemble de ce processus d'interprétation s'effectue sur des plateformes logicielles mettant à disposition une large gamme d'outils. Certains d'entre eux sont dédiés à la caractérisation des failles et d'horizons, comme les outils assistés de pointé ou les propagateurs automatiques, d'autres sont dédiés à la modélisation du modèle structurel, sur des images sismiques (par un ensemble de polygones) ou bien dans l'espace 3D (par un ensemble de surfaces), enfin, un dernier outil permet de valider la cohérence de ce modèle avec les hypothèses de déformation avancées : la restauration.

La validation d'une interprétation par sa restauration, consiste à appliquer l'hypothèse de déformation formulée au modèle structurel considéré. On déforme alors le modèle conformément aux



lois énoncées par l'hypothèse testée et l'on escompte obtenir un modèle représentant le sous-sol dans une configuration antérieure aux différentes déformations produites. Si une telle rétro-déformation, respectueuse des différentes lois, permet d'obtenir un modèle géologiquement cohérent, on considère alors que l'interprétation est juste dans son ensemble. Dans le cas inverse, elle est à revoir.

La précision de l'interprétation est essentielle tant en phase d'exploration (pour déterminer la probabilité de présence d'hydrocarbures), qu'en phase d'exploitation (pour simuler l'écoulement des fluides). Aussi, la restauration est un outil précieux. Toutefois, certains géologues et géophysiciens pensent qu'il pourrait intervenir plus tôt, et en particulier se transformer en outil d'aide à l'interprétation. En effet, l'interprétation d'images sismiques est un processus long et coûteux. Aussi, il pourrait d'une part être intéressant d'assister la création du modèle structurel et la formulation des hypothèses de déformation par les mêmes outils qui assureront sa validation ; et d'autre part cela permettrait de voir le processus d'interprétation comme une série de raffinements, plutôt qu'une série d'essais et d'erreurs chronophages.

Lorsque les géologues découvrent une image sismique, ils effectuent quelques pointés évidents pour y déterminer une structure grossière, puis ils commencent à réfléchir sur la manière dont le sous-sol s'est déformé, ce qui leur permet par exemple de corrélérer des structures discontinues. Un outil de restauration leur permettrait à cet instant de tester interactivement ces différentes hypothèses. Ils pourraient déterminer la plus probable, ce qui permettrait d'effectuer des pointés non-évidents sur l'image sismique, et par raffinements successifs, nous obtiendrions alors une interprétation finie. L'objectif de travail a donc été d'intégrer cet outil de restauration aux phases préliminaires de l'interprétation.

Dans le chapitre 1, nous présenterons les aspects géologiques nécessaires à la compréhension de ce document. Nous y détaillerons le processus d'acquisition sismique permettant d'obtenir une imagerie numérique du sous-sol, puis décrirons comment s'effectuent les pointés de failles et d'horizons. Nous illustrerons par quelques exemples certaines situations qui peuvent s'avérer délicates lors d'une interprétation préliminaire, et expliquerons comment la restauration permet de simplifier ces cas. Nous donnerons ensuite au lecteur un aperçu des différentes méthodes de restauration existantes et verrons que le sujet est riche, par des applications en 2D, en 3D, qui traitent différents cas, avec plusieurs degrés de réalisme. Cet aperçu nous permettra toutefois de mesurer que les méthodes décrites requièrent toutes au minimum une interprétation finie : c'est-à-dire l'ensemble des pointés de failles et horizons, mais aussi l'hypothèse de déformation correspondante. Or comme nous souhaitons appliquer les principes de la restauration aux étapes préliminaires de l'interprétation, nous ne possédons à ce stade qu'un ensemble de pointés grossiers et fragmentés. Nous avons en conséquence choisi de développer notre propre outil de restauration, en nous appuyant sur des critères classiques établis par la littérature.

Puisque restaurer une image sismique consiste à la déformer pour rétablir l'état originel de la sismique visualisée, cette démarche vise à concevoir un outil de déformation d'image. Nous allons donc dans le chapitre 2 nous intéresser de manière générale à la déformation de modèles numériques. Nous verrons au travers d'une série d'exemples que ce sujet couvre des applications variées et des domaines

très différents. Nous proposerons alors un formalisme commun permettant de décomposer toute déformation en trois étapes, qui sont l'application de contraintes sur un ensemble de points représentant le modèle, le calcul du déplacement de ces points, et la déformation du modèle par interpolation des ces déplacements. Pour chacune de ces étapes, nous présenterons plusieurs approches existantes et choisirons celles qui nous semblent les plus pertinentes dans notre application de restauration d'une image sismique.

Ayant conçu notre outil de restauration, nous exposerons le processus permettant de calculer la déformation dans le chapitre 3. Nous y décrirons alors en détail la manière dont nous traduisons les différentes contraintes régissant la déformation par un ensemble d'équations. L'ensemble de ces contraintes produira un ensemble de termes à minimiser et nous les regrouperons dans une formulation aux moindres carrés, qui s'avèrera non linéaire. Cette non linéarité empêchera toute résolution directe du système d'équations, aussi nous utiliserons une approche itérative de type *Gauss-Newton*, désormais classique pour optimiser des problèmes non linéaires de minimisation conjointe. Ce chapitre sera également l'occasion d'aborder diverses subtilités mathématiques ou informatiques liées à notre processus, qui permettront de déterminer ses paramètres de fonctionnement optimum.

Nous aurons alors décrit l'ensemble de la solution mise en œuvre pour restaurer des images sismiques et illustrerons dans le chapitre 4 diverses applications de notre travail. Nous montrerons au travers de cas classiques issus du processus d'interprétation, comment notre outil permet d'améliorer la détection d'objets géologiques, qui étaient ambigus sur l'image sismique non restaurée. Nous verrons également qu'il est possible, et plus simple, de pointer ces objets directement dans l'image restaurée. Nous généraliserons enfin le champ d'application de notre outil, en présentant des scénarios de restauration faisant intervenir plusieurs failles et plusieurs horizons.

Nous conclurons finalement ce manuscrit en discutant des apports de notre travail, de ses limites et possibles évolutions futures.



# Chapitre 1

## Présentation de la restauration sismique

### 1.1 Caractéristiques principales des images sismiques

Les géosciences regroupent l'ensemble des disciplines qui s'intéressent au sous-sol. Dans ce domaine, le sujet d'étude présente la particularité de ne pas être physiquement accessible, car enterré, hormis en un certain nombre de lieux où il est possible de faire des sondages géologiques (carottages), mais ceux-ci restent ponctuels et reviennent de plus très cher. Aussi, afin de fournir un support de travail aux métiers ayant besoin de visualiser et manipuler le sous-sol, les techniques modernes d'imagerie permettent de l'observer, puis de le reconstruire dans des modèles numériques.

On obtient une image sismique par émission de signaux acoustiques dans le sous-sol, provenant d'une source active (explosion, vibreur) ou d'une écoute passive (bruit sismique naturel), puis acquisition et traitement de ces signaux, réfléchis et convertis aux différentes interfaces géologiques constituant le modèle. Les ondes émises traversent les différents milieux du sous-sol à des vitesses variables. Lors de la rencontre avec des interfaces géologiques (typiquement un changement de milieu, un saut entre deux couches), une partie de ces ondes se réfléchit, tandis qu'une autre se réfracte et se transmet. Ainsi, l'intensité du signal écouté, le délai après émission, la forme d'onde, tout comme la position à laquelle on l'écoute, fournit une série d'informations localisées sur la composition du sous-sol. Ce processus est illustré sur la figure 1.1.a. Puis, en faisant varier les positions d'émission et de réception, on obtient des données supplémentaires à des angles d'incidence ou distance source-récepteurs différentes, que l'on agrège, tel qu'illustré en figure 1.1.b.

Ayant obtenu suffisamment d'informations sur l'ensemble de la zone étudiée, on peut créer un modèle numérique de la géologie du sous-sol. Après traitement et recombinaison de ces signaux, on génère finalement des coupes sismiques 2D (figure 1.2.a), ou des cubes sismiques 3D (figure 1.2.b). Ces données 2D ou 3D peuvent ensuite être visualisées sous forme d'image afin d'étudier le sous-sol. L'étape suivante consiste à interpréter ces images afin de qualifier les structures observables.

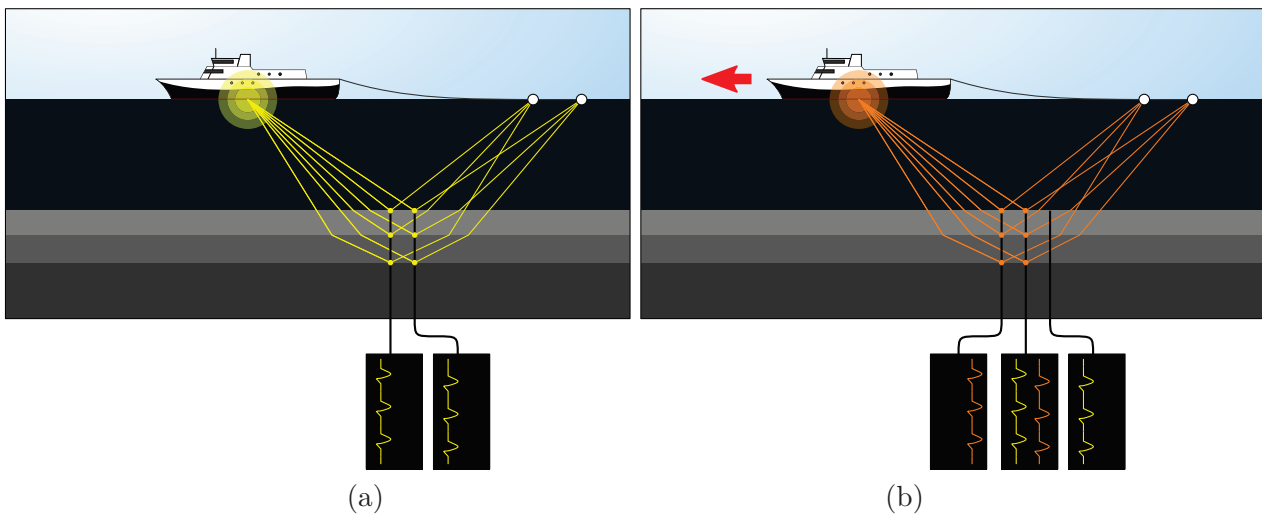


FIGURE 1.1 – **Processus typique d’acquisition de données sismiques** : (a) Émission et acquisition de signaux pour quelques positions, (b) déplacement du dispositif et acquisition en de nouvelles positions.

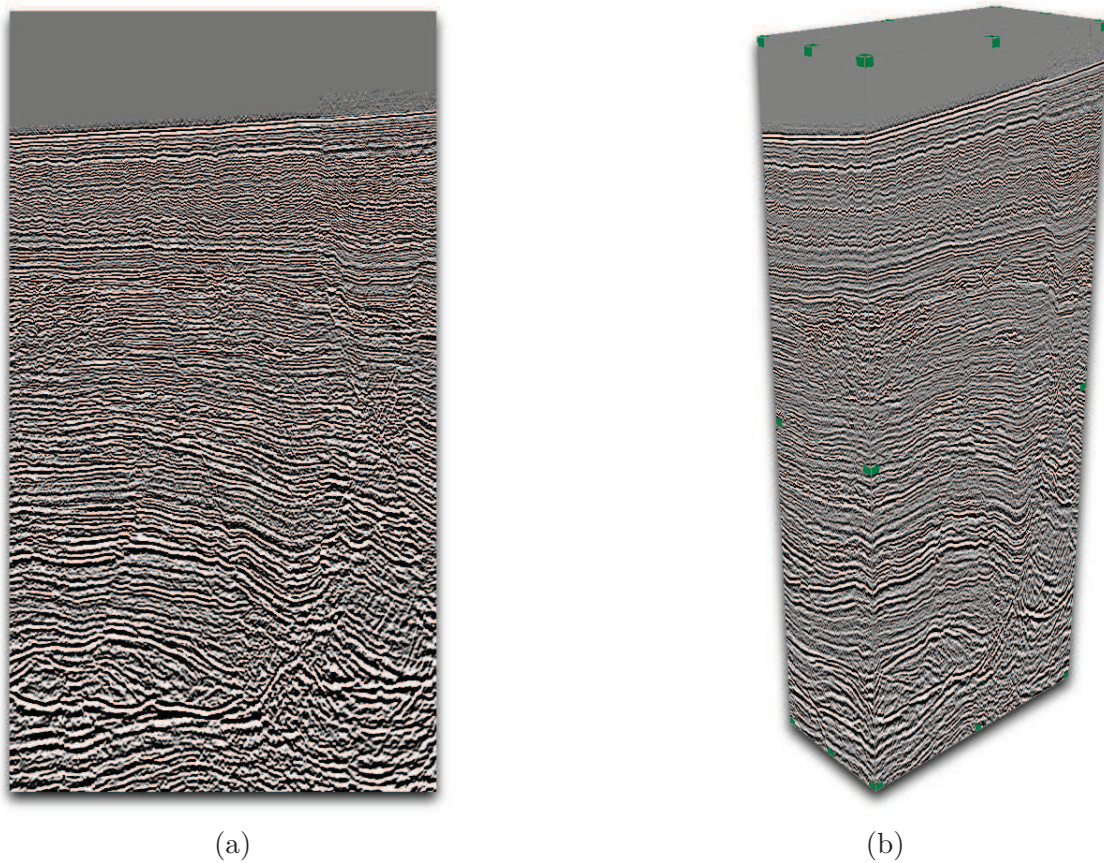


FIGURE 1.2 – **Exemples typiques d’images sismiques** : (a) Section sismique 2D, (b) cube sismique 3D.

## 1.2 Interprétation d'images sismiques

Dans l'industrie pétrolière, l'interprétation sismique est le métier qui analyse les images obtenues pour déterminer la structure du sol et sa composition. Dans cette étape, on cherche à identifier des évènements géologiques, les dater, puis expliquer les mécanismes qui ont conduit à la formation de ces évènements observés. Dans le domaine de l'exploration énergétique, cette analyse est nécessaire pour essayer de déceler les zones dans lesquelles des réservoirs à hydrocarbures se sont formés.

Le processus consiste dans un premier temps à identifier les évènements géologiques qui structurent le sous-sol. On s'intéresse en particulier à deux types d'évènements : les failles (discontinuités verticales ou sub-verticales causées par des mouvements tectoniques dans le passé), et les horizons (interfaces horizontales ou quasi-horizontales entre deux milieux de nature à peu près homogène, c'est-à-dire les interfaces entre couches géologiques différentes). Typiquement, on part de l'image sismique sur la figure 1.3.a, puis on pointe les failles sur la figure 1.3.b et les horizons sur la figure 1.3.c. Une fois cette structure globale extraite, on peut alors l'interpréter pour reconstruire l'histoire passée du sous-sol.

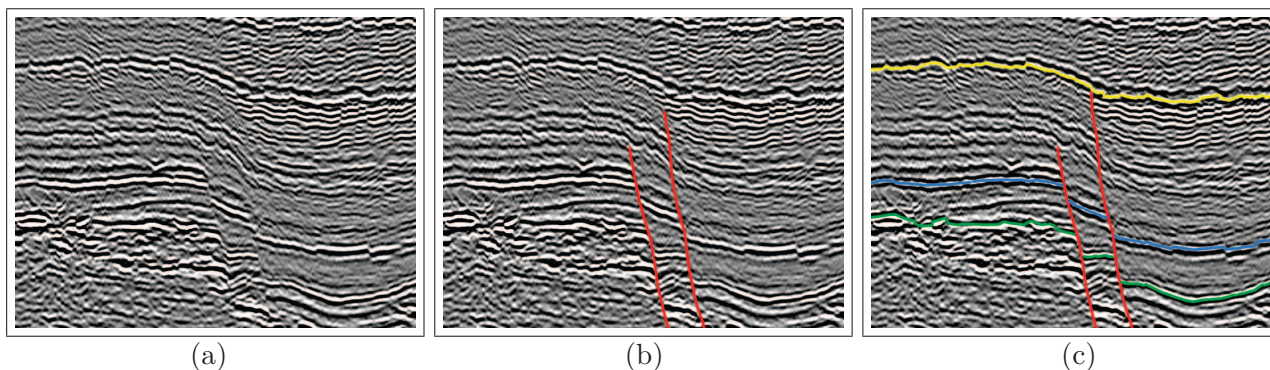


FIGURE 1.3 – Pointés typiques d'évènements sur une image sismique : (a) Image sismique initiale, (b) pointé des failles, (c) pointé des horizons. L'identification des évènements géologiques est terminée. Son interprétation par des géologues peut alors commencer.

## 1.3 La restauration : un outil précieux d'aide à l'interprétation

À qualité d'image équivalente, l'identification d'évènements peut être plus ou moins difficile selon le scénario géologique rencontré. Dans le cas de la sismique sur la figure 1.4.a, les horizons sont aisément lisibles et aucune faille importante n'est présente. Le pointé illustré sur la figure 1.4.b est donc immédiat et simple.

En revanche, la sismique sur figure 1.5.a présente une faille importante et de grande taille. À cause de cette discontinuité, corréler les horizons observés de part et d'autre est délicat. Cette situation conduit donc à une incertitude du pointé illustrée sur la figure 1.5.b.

En plus d'introduire de l'incertitude dans les pointés, la présence de failles peut rendre difficile l'observation de faciès particuliers. La sismique de la figure 1.6.a présente un motif subtil (surligné en jaune) que les géologues interprètent comme étant un complexe chenalisant. Détecter un tel évènement



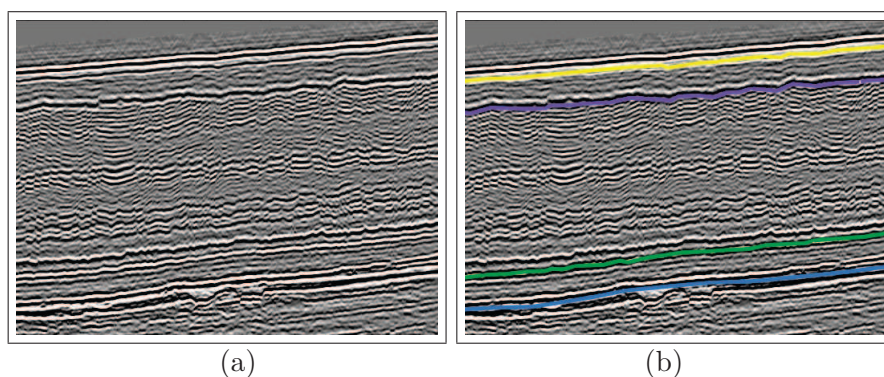


FIGURE 1.4 – **Pointé sur une sismique présentant un faciès horizontal et continu** : (a) Sismique initiale où des horizons sont aisément distinguables, (b) pointé d’horizons.

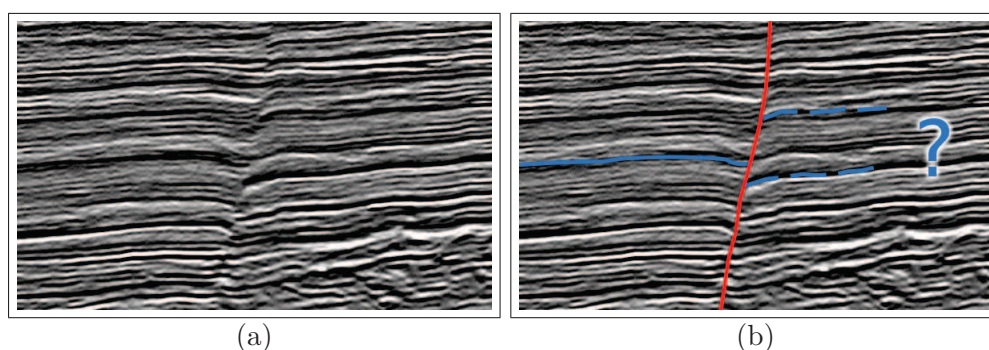


FIGURE 1.5 – **Difficulté de pointé dans le cas d’une sismique faillée** : (a) Sismique initiale, (b) pointé d’un horizon. À cause de la faille en rouge, la corrélation de l’horizon est difficile.

géologique présente un vif intérêt en phase d’interprétation pour reconstituer l’histoire du sous-sol. À présent, nous présentons sur la figure 1.6.b le même chenal, mais dans une zone où il est segmenté par un jeu de failles. Délimiter précisément le contour du chenal est alors rendu plus difficile, surtout dans les deux zones comprises entre les failles.

On observe à travers ces exemples que la présence de discontinuités sur la sismique rend plus difficile l’identification d’évènements géologiques, et donc la compréhension du système sédimentaire. En effet, ces discontinuités, ou flexures observées, sont la conséquence de déformations (phénomènes tectoniques structuraux) qui se sont produites dans le sous-sol au cours du temps et qui sont postérieures aux dépôts sédimentaires que l’on souhaite analyser. La figure 1.7 illustre ce mécanisme de déformation et explique comment un système sédimentaire peut être déformé pour arriver à un état actuel observable sur la sismique.

Le processus de restauration consiste alors, à partir d’une image observée, à identifier les phénomènes de déformation, puis à appliquer l’ensemble des déformations inverses permettant de revenir à une image illustrant le système de dépôts initial. La figure 1.8 illustre ce processus de restauration. Tout d’abord, un pointé des évènements géologiques majeurs est réalisé (comme la faille rouge, et l’horizon bleu sur la figure 1.8.b). Ensuite les deux déformations liées à la faille et à la flexure sont inversées afin de reconstruire l’image 1.8.d représentant l’état avant déformation. On retrouve le système présenté

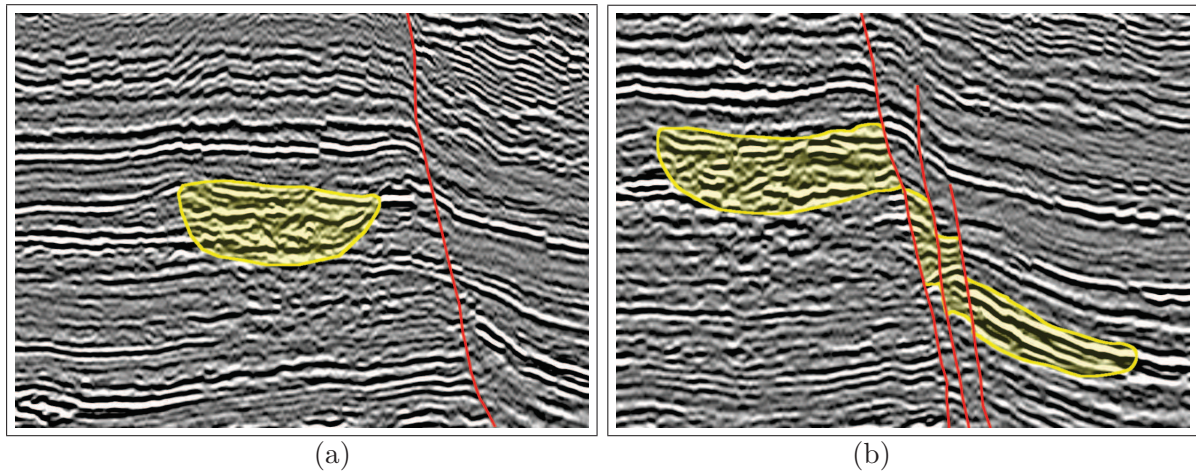


FIGURE 1.6 – Illustration des difficultés de l'interprétation en environnement faillé : (a) Section présentant un chenal simple à visualiser, (b) section présentant un chenal difficile à visualiser à cause de la présence de failles.

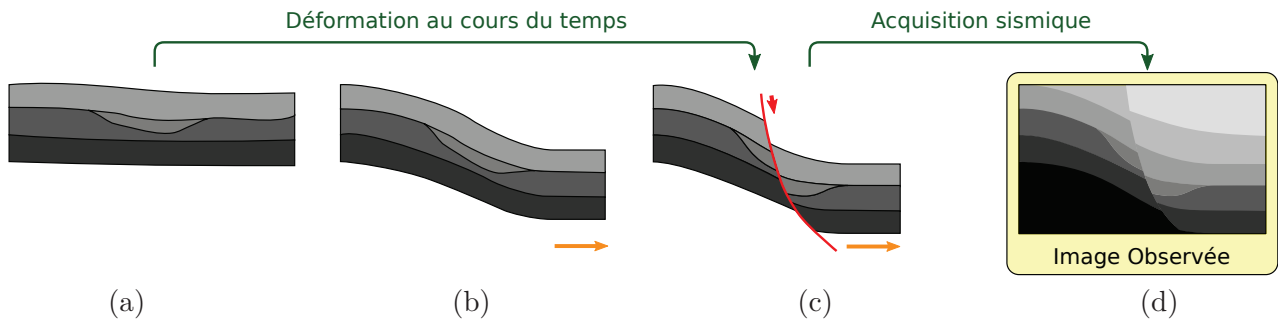


FIGURE 1.7 – Déformation du sous-sol au cours du temps : (a) Avant déformation, les différents dépôts et faciès particuliers sont aisément identifiables. (b) Un phénomène d'extension se produit dans le sous-sol et conduit à sa déformation. (c) La déformation atteint un point de rupture, causant la création d'une faille. Le faciès particulier aisé à identifier avant déformation devient plus délicat à observer. (d) Après acquisition sismique, on obtient une image du sous-sol contenant l'ensemble des déformations qui s'y sont produites.

sur la figure 1.7.a.

Sur cette image restaurée, l'identification d'évènements géologiques subtils devient plus aisée. Par exemple, le pointé du chenal (base en jaune et toit en vert sur la figure 1.9.b) est beaucoup plus simple sur l'image restaurée. Finalement, si on assure une bijectivité du processus de déformation, alors ce nouveau pointé pourra aisément être reporté sur l'image observée (cf figure 1.9.c). L'outil de restauration devient alors une aide cruciale, non seulement pour comprendre les mécanismes géologiques de déformation, mais aussi pour le pointé des évènements subtils.



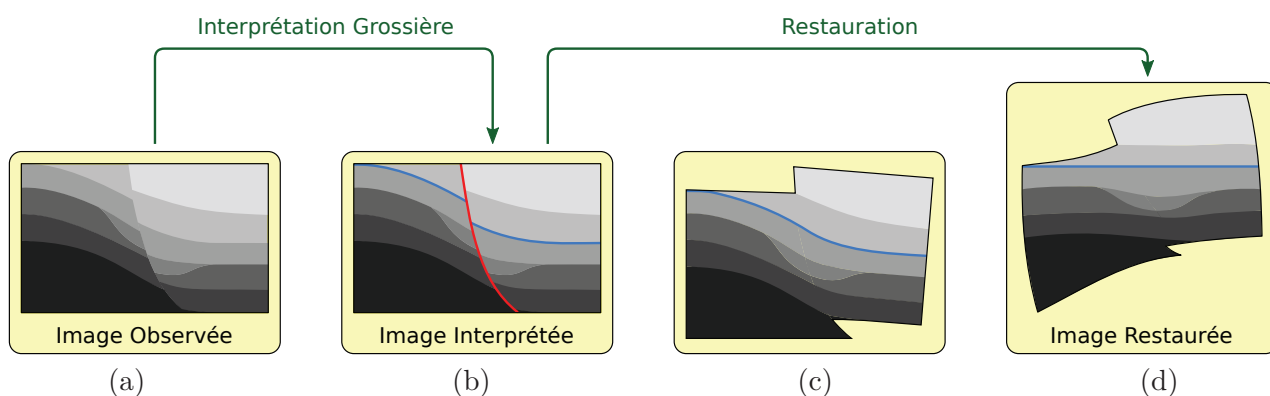


FIGURE 1.8 – **Restauration typique d’une image observée** : (a) Image observée, (b) image où sont grossièrement pointés une faille et un horizon, (c) restauration de la discontinuité, (d) remise à plat de l’horizon. On obtient alors une image restaurée.

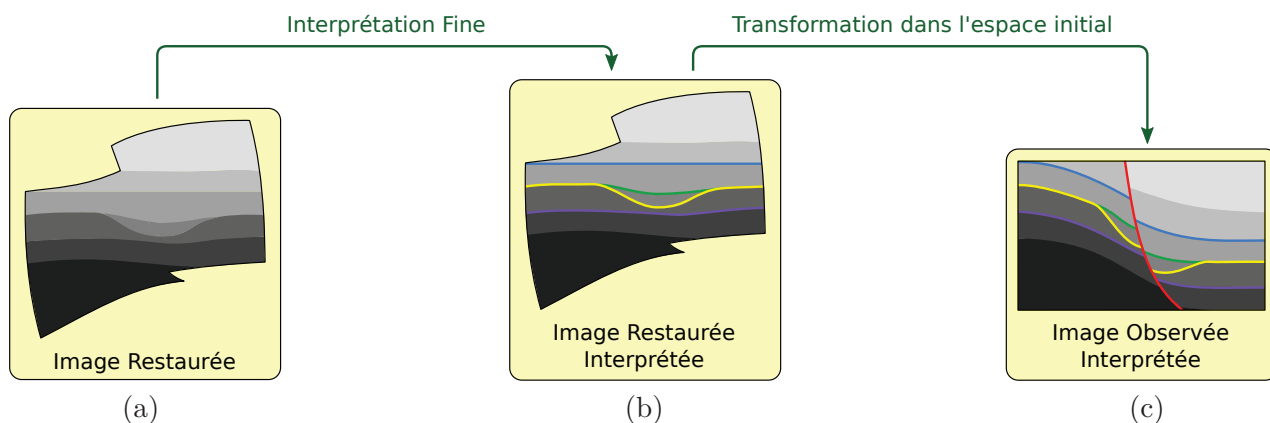


FIGURE 1.9 – **Interprétation dans l’image restaurée** : (a) Image restaurée où tous les objets à identifier sont plus aisément observables et présentent moins de doute, (b) pointé fin et interprétation dans l’image restaurée, (c) export des pointés vers l’image observée.

## 1.4 Description succincte du processus de restauration

Comme nous venons de le voir, l’objectif de la restauration consiste à déformer une image sismique observée, en appliquant l’ensemble des transformations inverses qui se sont produites de manière cumulative au cours du temps. Ceci permet alors de fournir aux géologues des images plus simples à interpréter.

Les déformations qui se produisent dans le sous-sol sont des forces d’extension ou de compression. Dans les deux cas, ces forces créent une flexure des horizons, puis si l’on atteint un point de rupture, causent la formation d’une faille. Ces deux processus, extension et compression, sont illustrés sur la figure 1.10 avec dans un premier temps le mécanisme de flexure (figure 1.10.b) puis de génération de la faille (figure 1.10.c). Ainsi la restauration consistera à rétablir la continuité de la pile sédimentaire en zones faillées, puis inverser les flexures en remettant à plat un horizon de référence.

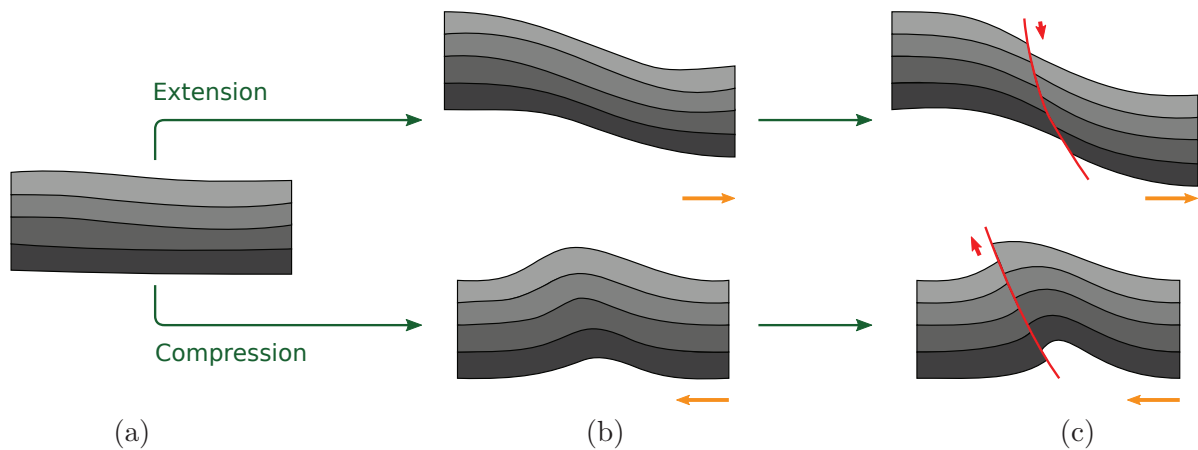


FIGURE 1.10 – **Processus de déformation du sous-sol** : (a) État initial. (b) Flexure des horizons, en haut par extension, en bas par compression. (c) Formation d’une faille, en haut par extension, en bas par compression.

### 1.4.1 Restauration des failles

La restauration d’une faille pose le problème de rétablir la continuité de la pile sédimentaire dans une image sismique. Géologiquement, on considère qu’un bloc est statique, tandis que l’autre s’est déplacé, formant ainsi la faille, puis l’accentuant. Sur la figure 1.10.c, ce sont les blocs de droite qui se sont déplacés.

Restaurer un tel modèle consiste à faire glisser la partie mobile du modèle le long de la faille, de manière à restaurer la continuité d’un marqueur identifié, et présent de chaque côté de la faille. Idéalement, un tel marqueur est un horizon, car il sera particulièrement simple à remettre en cohérence. Cette opération ne requiert en effet que de confondre les deux points appartenant à la fois aux extrémités de l’horizon et à la faille.

Suivant la géométrie de la faille, faire glisser la partie mobile du modèle consiste en une simple translation si la faille est parfaitement rectiligne, ou une rotation si la faille décrit un arc, dont le rayon de courbure est constant. Pour toutes les autres géométries possibles de la faille (typiquement, une courbure irrégulière), la restauration engendrera une déformation de la partie mobile. Ce dernier cas est à-priori le plus répandu.

Il est donc important de contrôler cette déformation par un ensemble de critères à respecter. Par exemple, [Dah69] énonce la loi de conservation des volumes, stipulant que ces derniers ne doivent pas être altérés par la déformation. Ce genre de critères sera alors introduit dans le mécanisme de restauration comme un ensemble de contraintes à respecter.

Il est à noter qu’en pratique, le mécanisme de restauration ne concerne souvent pas une faille unique, mais plutôt un ensemble de failles. Le processus devra alors être itératif pour restaurer l’ensemble des discontinuités : cela nécessitera de la part de l’interpréteur d’être en mesure de classer les failles par ordre chronologique d’apparition afin de les restaurer dans l’ordre inverse, ou bien de tester interactivement plusieurs ordres de restauration pour déterminer le plus probable.

### 1.4.2 Mise à plat d'un horizon

La restauration d'un horizon vise à traiter le problème des flexures, illustré sur la figure 1.10.b, où différentes couches se sont successivement déposées par sédimentation et présentait initialement une géométrie plane. Les différents phénomènes de déformation (de nature compressive ou extensive) ont causé la formation des structures bombées ou étirées qui l'on peut observer aujourd'hui.

Étant donné un horizon de référence, inverser la flexure consiste à remettre à plat cet horizon. Dans le cas 2D (sur des images sismiques), restaurer consiste à déplier une polyligne représentant l'horizon de manière à la faire coïncider avec une droite horizontale, tout en conservant sa longueur. Dans le cas 3D (cubes sismiques), l'horizon est représenté par une surface que l'on doit projeter sur un plan, idéalement sans la déformer (ce qui permet ainsi de conserver son aire). Ce processus implique alors que la surface soit dite "dépliable", ce qui en pratique est rarement le cas. Aussi les algorithmes de restauration d'horizon minimisent la déformation lors de l'opération.

Comme cela a déjà été mentionné pour les failles dans le cas général, cette déformation n'est pas un mécanisme linéaire et impactera le reste de l'image ou du cube sismique. Il sera important d'introduire sur l'ensemble de la région considérée un ensemble de contraintes, comme par exemple la préservation de la surface, ou bien la préservation des volumes.

## 1.5 État de l'art de la restauration

La restauration est un sujet qui est fréquemment et extensivement abordé dans la littérature. Ce domaine a fortement évolué au cours de ces cinquante dernières années et diverses approches ont été abordées. Les évolutions sont venues tant d'avancées théoriques sur les modèles et les approches, que d'avancées dans le domaine de l'informatique et dans l'augmentation phénoménale de la puissance des ordinateurs, qui a rendu progressivement certains calculs et certaines approches possibles.

Il existe ainsi actuellement des approches de restauration d'images sismiques, de surfaces, ou de volumes, qui s'appliquent à des hypothèses de déformation d'extension ou de compression, et font intervenir des critères géométriques ou physiques. De nos jours, il est habituel de classer les méthodes de restauration en deux catégories : les méthodes géométriques qui ne font intervenir que des grandeurs géométriques (longueurs, aires, rotations...) et les méthodes physiques qui introduisent en plus des grandeurs décrivant la physique du problème (masse, rigidité c'est-à-dire capacité ou résistance à la déformation...) en plus de considérer sa géométrie. Nous allons donc lister et résumer ci-dessous les principales caractéristiques de ces différentes méthodes, ainsi que leurs objectifs principaux et les critères qu'elles utilisent habituellement.

### 1.5.1 Méthodes de restauration géométriques

Les premières méthodes de restauration par *section balancing* [Dah69, Hos79], proposent de tester la validité d'une section sismique interprétée, pour laquelle on formule l'hypothèse d'une déformation par compression (*flexural slip*). La restauration consiste alors à aplanir un horizon de référence coupé par une faille, en respectant la loi de conservation des volumes (dont les principes remontent à [Cha10]),

c'est-à-dire en déformant la section de manière à préserver les longueurs des autres horizons ainsi que les aires des différentes couches déformées. S'il est possible de déterminer un modèle restauré, géologiquement crédible tout en respectant ces critères, on peut alors considérer que l'interprétation est valide. Dans le cas contraire, les zones présentant des difficultés lors de la restauration (impossibilité de conserver les épaisseurs de couches et les longueurs d'horizons par exemple) donnent une indication sur l'endroit où l'interprétation est à-priori mauvaise.

Dans [Gib83, WJM86, RK89], les auteurs étendent l'outil de *section balancing* au scénario géologique d'une extension. Cette méthode permet de restaurer une faille en faisant glisser une couche du sous-sol le long de celle-ci. En plus de la loi de préservation des volumes, on contraint alors les déplacements selon un vecteur de cisaillement vertical (*vertical shear*) ou bien orthogonal à la faille (*inclined shear*).

Puis ces différentes méthodes ont été regroupées dans des processus automatisés, plus généraux permettant la restauration d'une section sismique en traitant les deux cas [MR90, MTE90].

Les méthodes dites de *surface unfolding* ou *map restoration* ont elles pour but de valider l'interprétation d'un horizon représenté par une surface 3D. Pour cela, elle mettent à plat l'horizon en fermant les lèvres de failles (c'est-à-dire en s'assurant que la surface restaurée ne présente ni vide ni recouvrement, sachant qu'un vide ou un recouvrement entre deux couches n'est pas réaliste d'un point de vue géologique et donc pas acceptable) et minimisent la déformation. Les zones fortement déformées suggèrent alors une mauvaise interprétation locale de l'horizon. [Bar85] propose de traiter le scénario de la flexure en supposant que le pli formé par l'horizon présente une forme approximativement cylindrique. Un cylindre étant un objet dont la surface est strictement dépliable (c'est-à-dire dont on peut dessiner le patron sur un plan), on peut ainsi déplier l'horizon en l'étirant dans le sens normal au pli tout en conservant les aires. Les lèvres de failles sont ensuite fermées une par une en appliquant sur différentes parties de la surface un déplacement horizontal constant selon une direction normale à la faille (*simple shear*, ou cisaillement simple).

Une autre méthode maintenant classique de *surface unfolding* [Gui91, GGDO91, GG93] propose de traiter le cas de flexures plus difficiles à déplier et qui ne peuvent donc pas être traitées par un modèle cylindrique. Une première étape consiste à appliquer sur tous les triangles composant la surface de l'horizon la rotation de plus faible amplitude qui rend ces triangles parallèles à un plan horizontal. Puis ces triangles sont projetés sur le plan horizontal de référence. Cette opération ayant produit un ensemble de triangles ayant perdu leur connectivité et se chevauchant, une seconde étape va consister à optimiser l'agencement de manière à réduire autant que possible les vides et les chevauchements. Cette phase d'optimisation consiste à déterminer la translation horizontale et la rotation minimale à appliquer à chaque triangle de manière à minimiser la somme des distances des noeuds devant se connecter. Finalement une troisième étape consiste à recréer la connectivité des triangles en fixant la position de connexion au barycentre de chaque ensemble de noeuds devant se connecter. Les triangles sont alors déformés, mais grâce à l'étape d'optimisation du réagencement cette déformation garantit une altération minimum de l'aire de chaque triangle. Il est à noter que ces travaux ont été étendus par [WKBR97], qui proposent de garantir strictement la conservation de l'aire de chaque triangle lors de l'étape de régénération du maillage. On peut également citer [RXS00], où les auteurs proposent une

méthode de restauration 3D utilisant cet algorithme pour mettre à plat les horizons. Leur contribution traite le cas du scénario géologique d’extension et ils proposent d’aplanir chaque triangle non plus avec une transformation (translation et rotation) conservant l’aire de chaque triangle, mais par projection sur un plan horizontal, selon une direction donnée (*simple shear*). Cette direction peut être choisie soit constante soit variable selon l’hypothèse géologique que formule l’utilisateur.

Enfin, les critères géométriques sont également présents dans des méthodes de restauration volumique. Dans [LTGM97], les auteurs proposent de restaurer un modèle volumique non faillé composé de plusieurs couches dont les contacts sont des horizons (*multisurface unfolding*). Leur approche consiste à déformer un maillage du modèle de manière à optimiser trois critères : l’horizontalité des interfaces, l’aire des interfaces, et les volumes des couches. On retrouve également ces mêmes critères dans la méthode de restauration volumique proposée par [GJS<sup>+</sup>02].

### 1.5.2 Méthodes de restauration physiques

Les méthodes présentées ci-dessus déterminent la déformation nécessaire à la restauration selon des lois purement géométriques (conservation des aires, des volumes, minimisation de la déformation par minimisation de la rotation des triangles). Lors de l’exploitation de réservoirs à hydrocarbures interviennent des simulations d’écoulements de fluides en milieu poreux (roche poreuse). Connaître précisément la déformation qui s’est produite dans les couches géologiques au cours du temps permet d’améliorer le réalisme de ces simulations, en modélisant par exemple l’effet de la présence de failles sur l’écoulement des fluides à travers la structure. Aussi, lorsque le modèle géologique le permet, c’est-à-dire lorsque l’on a bonne confiance en son interprétation et que l’on peut y déterminer des paramètres physiques (masse, rigidité...), des approches de restauration décrivant la mécanique de la déformation sont envisageables. De ce fait, d’autres méthodes plus complexes et plus coûteuses mais plus réalistes ont été développées plus récemment en cherchant à déterminer cette déformation selon des lois géomécaniques.

Dans [DSCM02], les auteurs proposent de restaurer une section sismique en modélisant la déformation selon une loi physique d’élasticité linéaire. Ils implémentent en particulier la relaxation dynamique [Und83] qui fait intervenir les grandeurs physiques de masse et de rigidité. [DSCM03] ont ensuite étendu ces travaux au cas de la restauration volumique.

Les méthodes de restauration géomécaniques récentes [Mur05, MLG06, MM06, DRCM10] modélisent quant-à-elles la conservation des masses et du mouvement selon la loi élastique linéaire de Hooke [JC76], qui est adaptée pour des déformations de faible amplitude. L’utilisation de la loi de Hooke nécessite d’appliquer au modèle du sous-sol considéré des propriétés mécaniques décrites par les coefficients d’élasticité de Lamé, qui font intervenir la rigidité (module de Young) et le taux de contraction du matériau (coefficient de Poisson). Ces propriétés mécaniques s’obtiennent par analyse lithologique ou inversion pétro-élastique des données sismiques [Doy07], car pour un grand nombre de types de roches ces coefficients élastiques sont connus (mesurés en laboratoire). Une fois déterminé le type de roche pour chaque couche du modèle on connaît donc ces coefficients.

Enfin, pour améliorer la modélisation de la déformation inverse, les travaux de recherche actuels s’orientent vers la prise en compte de nouveaux paramètres et comportements encore plus réalistes et donc plus complexes et coûteux. On peut par exemple citer les travaux récents de [DRSF<sup>+</sup>11] dans lesquels les auteurs introduisent dans la restauration le taux de compaction du sous-sol, qui n’est pas pris en compte dans les techniques basées sur l’élasticité linéaire mentionnées ci-dessus.

## 1.6 Création d’un nouvel outil de restauration

Parmi les différentes méthodes (géométriques et géomécaniques) de restauration présentées plus haut, aucune ne nous semble pleinement satisfaisante pour répondre à notre besoin d’un outil d’aide à l’interprétation d’images sismiques.

En premier lieu, nous avons pu constater que les méthodes géométriques actuelles sont principalement dédiées à la validation d’interprétation. Ces méthodes nécessitent donc un jeu complet de pointés de failles et d’horizons, ainsi qu’une hypothèse géologique de déformation (décrivant si la nature de la déformation produite est extensive, ou compressive). Or notre outil consiste précisément à assister la création de ces différentes données et nous ne disposons à ce stade que d’une interprétation préliminaire grossière (au moins un pointé d’horizon).

D’autre part, les méthodes géomécaniques ont l’inconvénient de nécessiter une large connaissance du sous-sol étudié et en particulier un ensemble de données physiques (masse, rigidité. . .). Ces données, complexes à obtenir ou modéliser, ne sont en aucun cas disponibles lors de l’interprétation préliminaire d’images sismiques. De plus, ces méthodes nécessitent (de par leur complexité) un coût de calcul important, donc incompatible avec notre application requérant une méthode de restauration quasi temps-réel.

Pour ces raisons, nous avons choisi de développer un nouvel outil de restauration d’images sismiques et allons nous appuyer sur les méthodes géométriques existantes en intégrant les critères classiques. Ainsi, cet outil devra permettre de déformer une image sismique tout en respectant les critères de conservation des longueurs et des aires, décrits plus haut.

Dans ce but, nous allons mettre en œuvre un outil de déformation d’image et proposons dans un premier temps de s’intéresser à ce domaine, en dédiant le chapitre 2 à l’étude de la déformation de modèles numériques.



## Chapitre 2

# Déformation de modèles numériques

L'opération de restauration des images sismiques qui nous intéresse dans ce travail nécessite l'usage d'une déformation de modèles numériques. Nous entendons par modèle numérique, un modèle défini par un ensemble de points. Il peut s'agir par exemple d'un maillage triangulaire dans le cas d'objets 3D pour les films d'animation ou d'un ensemble de pixels dans le cas d'images et notamment d'images sismiques.

Une déformation désigne une transformation géométrique plus générale que les transformations géométriques affines (translation, rotation et homothétie). Contrairement aux transformations affines qui modifient la position, l'orientation et la taille d'un modèle numérique tout en conservant sa forme globale, une déformation modifie la forme même du modèle à laquelle elle s'applique (d'où son nom). Logiquement, la définition d'une déformation requiert plus de paramètres que celui de transformations affines. Pour rendre cette définition aisée, de nombreuses techniques de contrôle de la déformation de modèles numériques ont été développées. Dans la suite de ce chapitre, nous commençons donc par exposer différents exemples d'applications faisant intervenir des déformations (section 2.1). Nous introduisons ensuite (section 2.2) un formalisme général exprimant la déformation d'un espace. Dans la dernière section (2.3) nous précisons les choix que nous avons adoptés pour répondre au cas particulier de la restauration d'images sismiques.

### 2.1 Introduction résumée aux techniques de déformation

Le thème de la déformation d'une structure est un sujet riche dans la pratique (mécanique, restauration. . .) ainsi que dans la littérature et dont les applications couvrent de nombreux domaines, aussi bien en 2D qu'en 3D. Nous allons en présenter ici quelques exemples.

Un premier exemple qui peut venir à l'esprit dans notre cadre de restauration d'images sismique est l'ensemble des techniques d'*image warping* [GM98]. On entend sous ce terme toute technique permettant de manipuler une image dans le but de la déformer. Cette déformation peut avoir pour objet par exemple la correction de distorsion [CF98] ou bien un but purement artistique comme la



déformation de perspectives pour améliorer la composition d'image [CAA10]. La figure 2.1 illustre un exemple de modification d'image afin de corriger une distorsion optique liée à une prise de vue.



FIGURE 2.1 – **Exemple de déformation d'image corrigeant la distorsion optique d'un objectif.** (a) Photographie initiale présentant une distorsion de la perspective suite à l'usage d'un objectif grand angle. (b) Photographie corrigée par une technique d'*image warping*.

Les techniques d'alignement d'image (*image registration*) utilisées entre autres en imagerie médicale [MV98] ou aérienne [KI03] sont un autre exemple d'*image warping*. On peut se référer par exemple à [Bro92] et [ZF03] pour un résumé détaillé du sujet. Le but est de composer une image finale à partir d'un ensemble d'images obtenues par des méthodes différentes (acquisition, modélisation...) ou des conditions d'acquisition d'image différentes (prise de vue, capteur...). Sur l'exemple de la figure 2.2 deux photographies sont prises depuis deux endroits distincts (figures 2.2.a et 2.2.b). Pour aligner ces images, un procédé automatique identifie des positions similaires dans chaque image (*landmark matching*) : ces positions sont l'ensemble de points rouges représentés sur l'image 2.2.c. Finalement, on détermine pour chacune des deux photographies la déformation qui remet en correspondance ces positions communes précédemment identifiées. La combinaison des deux images est alors possible et l'image finale obtenue est présentée sur le figure 2.2.c.

La modélisation 3D est également un thème très important pour la recherche et l'industrie dans lesquelles les techniques de déformation offrent aussi de nombreuses applications. On peut par exemple mentionner la conception assistée par ordinateur (CAO), dont le but est de créer des modèles numériques géométriques. Pour cela, un outil dédié, le modeler, permet de créer un ensemble d'objets géométriques simples appelés "primitives", par exemple des cubes, sphères, triangles ou des splines, que l'on transforme manuellement afin d'obtenir l'objet final de forme complexe que l'on désire créer comme modèle [SP86, Coq90, CR94]. Au même titre que les opérations de translation, de rotation ou d'homothétie, la déformation est une composante de base et un outil essentiel dans ce processus de création d'un objet complexe à partir d'un ensemble d'objets élémentaires. La figure 2.3 présente par exemple la déformation d'un cylindre permettant d'obtenir facilement une forme complexe en déplaçant manuellement un point de contrôle du cylindre. Dans cet exemple, la déformation honore

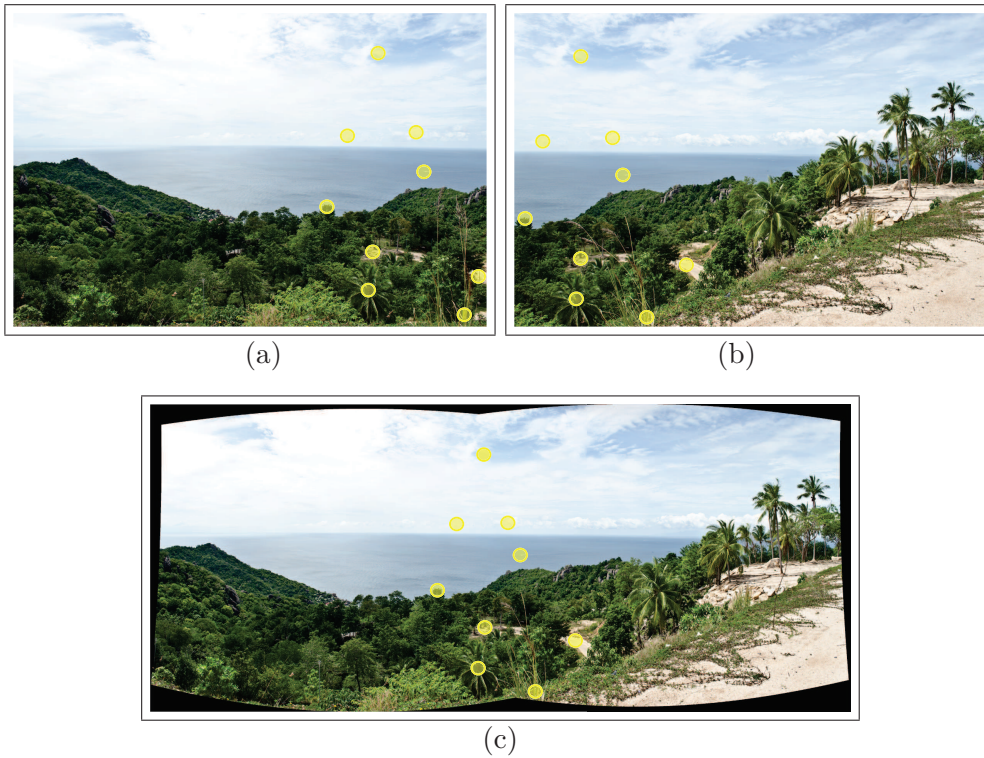


FIGURE 2.2 – Création d'un nouveau modèle par alignement d'image à partir d'une image brute non alignée. (a) et (b) Photographies initiales représentant la même scène prise depuis deux endroits distincts. (c) Image alignée obtenue après détection de positions similaires dans les deux photographies initiales.

une propriété de rigidité définie (illustrée par la carte de couleur), permettant de contrôler la zone où se produit la déformation, c'est-à-dire au milieu du cylindre.

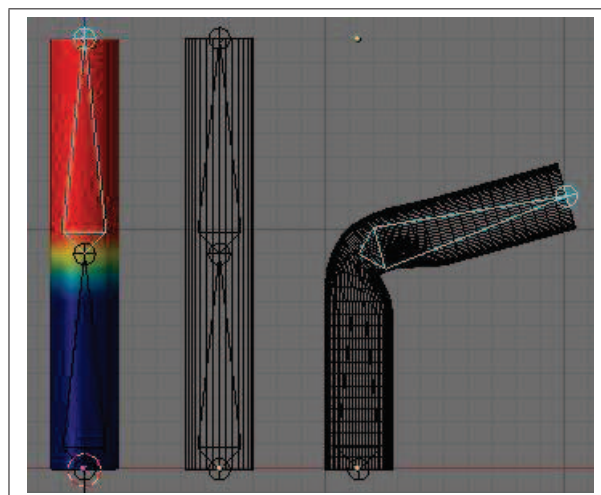


FIGURE 2.3 – Utilisation de l'outil de déformation dans le modèleur 3D de Conception Assistée par Ordinateur (CAO) libre *Blender* [Hes07].

Dans le domaine de l'animation (images ou films de synthèse), on s'intéresse à produire les déformations d'un modèle qui créent son mouvement au cours du temps, c'est-à-dire son animation. Les figures 2.5 et 2.6 présentent par exemple une déformation interactive dans laquelle l'utilisateur dispose de points de contrôle pour diriger la déformation. Ces points peuvent coïncider avec les points du maillage, comme cela est illustré sur la figure 2.4 pour une animation faciale [DN07, ED08]. Dans un souci d'optimisation, ces points peuvent s'organiser sous la forme d'une cage [JSW05] comme dans la figure 2.5 ou sous la forme d'un squelette [MtLTM88] comme dans la figure 2.6.

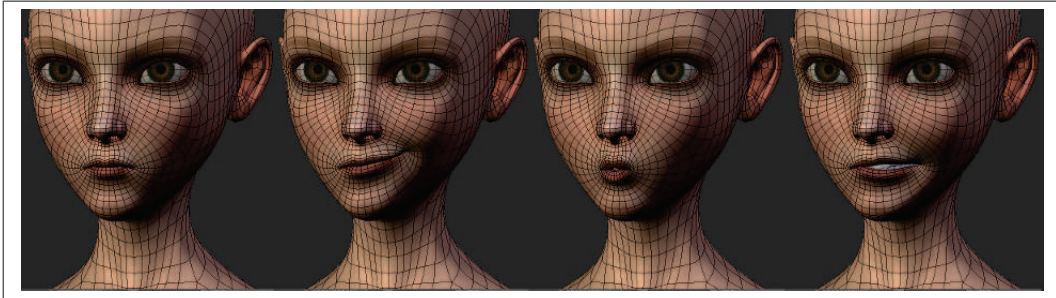


FIGURE 2.4 – Animation faciale issue du film en images de synthèses *Sintel* [LR10].

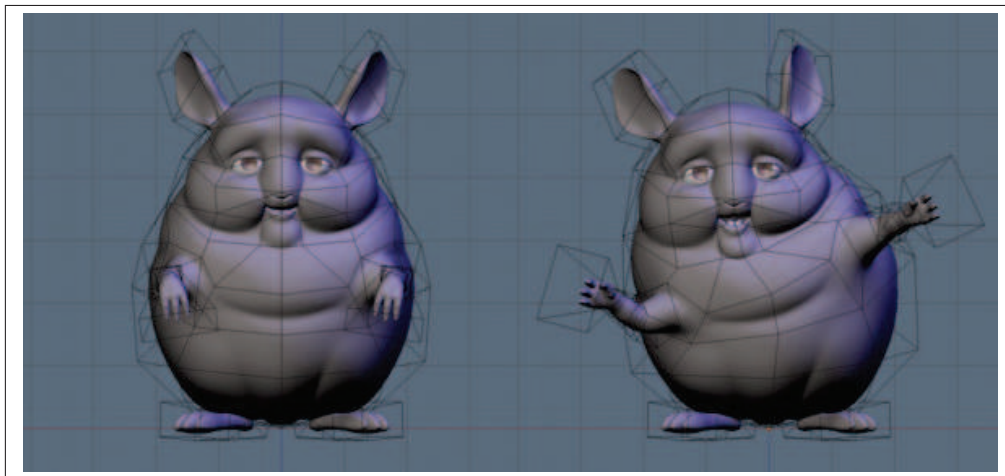


FIGURE 2.5 – Animation d'un personnage avec une cage issue du manuel en ligne de *Blender*.

Le thème de la déformation est également présent dans l'industrie textile, qui fait usage de modèles numériques pour concevoir des vêtements [LZY10]. De tels modèles peuvent alors être déformés, par exemple pour décliner plusieurs versions d'un produit, comme l'illustre la figure 2.7. La déformation est obtenue en habillant un sujet plus corpulent avec un modèle de référence.

On peut également mentionner le domaine de l'aéronautique, dans lequel la conception d'appareils (avions par exemple) nécessite de modéliser des déformations complexes [CCCJ04] comme l'interaction entre l'aile et un écoulement d'air. Dans l'exemple en figure 2.8, l'aile est soumise à des forces aéro-



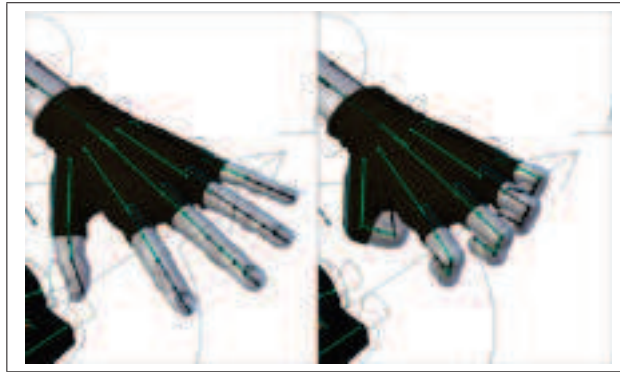


FIGURE 2.6 – Animation d’un personnage avec un squelette.

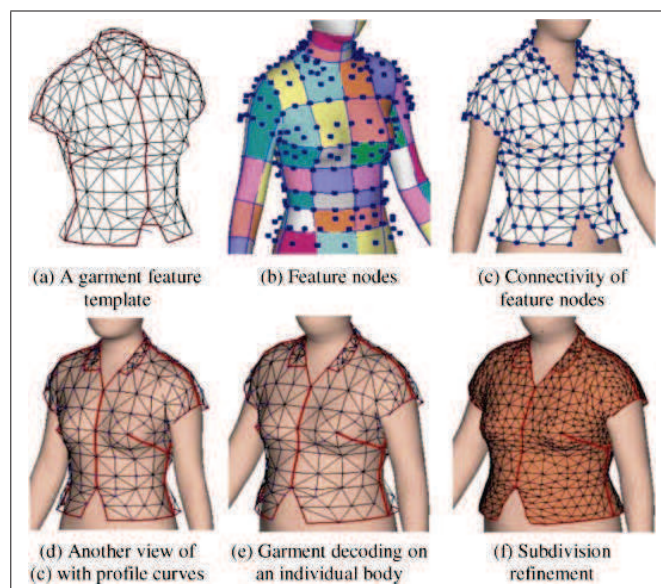


FIGURE 2.7 – Conception de vêtements par déclinaison de modèles existants [LZY10].

dynamiques qui la déforment et l’on peu ainsi mesurer quantitativement ses propriétés de flexibilité, aérodynamisme et portance du modèle fixé sur une base rigide.

La modélisation de déformations est également présente dans le domaine de la chirurgie [DAAM07, TS11]. La simulation permet à moindre risque aux étudiants de se familiariser à des gestes difficiles. Elle apporte également à la recherche la possibilité d’explorer de nouvelles possibilités pour des opérations et techniques chirurgicales futures. L’exemple de la figure 2.9 présente un modèle de déformation du tissu humain lors de son interaction avec un scalpel.

Enfin, comme nous l’avons mentionné et illustré au chapitre 1, les géosciences utilisent beaucoup de simulations de déformation pour le besoin de la restauration sismique et géologique.

Dans cette introduction, nous avons présenté quelques applications faisant intervenir une déformation. Malgré leurs buts variés et leurs domaines d’application différents, nous illustrons dans la suite

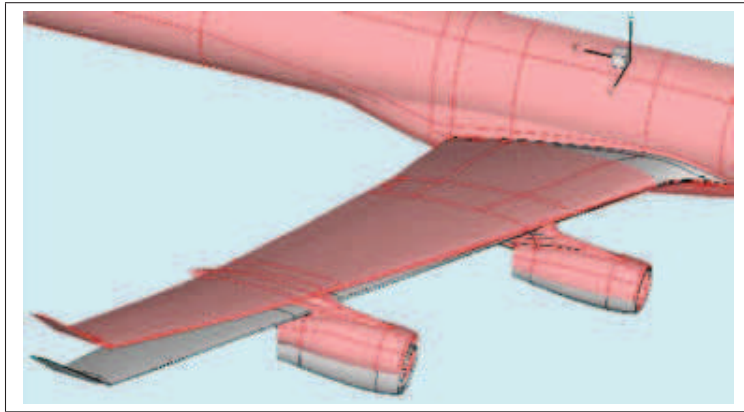


FIGURE 2.8 – Simulation aéroélastique [GL05].

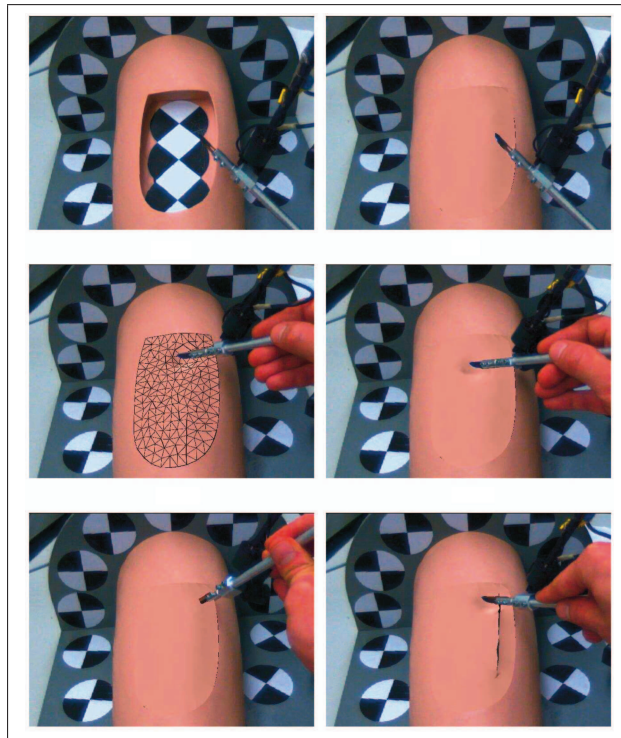


FIGURE 2.9 – Simulation d'opération chirurgicale avec un scalpel.

de ce chapitre le fait que tous ces processus de déformation respectent un même formalisme général. Nous présentons ce formalisme dans la section 2.2.

## 2.2 Formalisme général d'une déformation de modèle numérique

Les applications que nous avons présentées déforment toutes un modèle numérique, que l'on note  $\mathcal{P} = \{p_i(x_i, y_i)\}$  dans le cas 2D ou  $\mathcal{P} = \{p_i(x_i, y_i, z_i)\}$  dans le cas 3D. Les points  $p_i$  définissent le modèle numérique : il peut s'agir des pixels pour le cas d'une image ou des points d'un maillage d'un

objet 3D. Ce modèle  $\mathcal{P}$  est défini dans un espace noté  $\mathcal{S}$ , appartenant à  $\mathbb{R}^2$  pour le cas 2D ou  $\mathbb{R}^3$  pour le cas 3D. Aussi, déformer un objet  $\mathcal{P}$ , consiste à déformer l'espace  $\mathcal{S}$  dans lequel il est défini.

Dans un espace  $\mathcal{S}$ , initialement à l'équilibre, la déformation est introduite par un ensemble de contraintes perturbatrices, que nous appelons "contraintes externes", et qui s'appliquent sur un ensemble  $\{c_i\}$  de points donnés. Par exemple, dans l'application du recalage d'une image, on déforme cette dernière de façon à ce que certains points se déplacent à des endroits précis. On peut également penser à l'exemple d'une simulation physique en aérodynamique, où un modèle d'aile d'avion dans espace initialement à l'équilibre est soumis à une force aérodynamique nouvelle, perturbant l'équilibre initial.

Afin de permettre la déformation de l'espace par l'application des contraintes externes, il est nécessaire de caractériser la manière dont l'espace va se déformer. Dans ce but, l'espace possède des propriétés intrinsèques ainsi qu'un ensemble de règles de déformation appelées "contraintes internes", qui s'expriment en réaction aux contraintes externes.

Pour reprendre l'exemple de l'application du recalage d'une image, les contraintes internes peuvent imposer la déformation qui préserve au mieux la forme de l'image à recaler. Dans l'exemple de simulation physique, les contraintes internes vont traduire la réaction mécanique de l'aile (déformation légère) suite à la variation de la force aérodynamique. Sur le domaine considéré, on peut également s'intéresser à la variation de l'écoulement de l'air.

Contrairement aux contraintes externes qui ne s'appliquent qu'en un ensemble de points  $\{c_i\}$ , les contraintes internes impactent tout le domaine  $\mathcal{S}$ . De ce fait, afin de pouvoir calculer la déformation, il est nécessaire d'échantillonner  $\mathcal{S}$  par un ensemble de points. On complète ainsi l'ensemble de points  $\{c_i\}$  par un nouvel ensemble  $\{e_i\}$  décrivant  $\mathcal{S}$ , que l'on choisit. On appelle alors "points de contraintes"  $\{v_i\}$  l'union de ces deux ensembles, telle que :  $\{v_i\} = \{c_i\} \cup \{e_i\}$ .

Muni de ces points de contraintes  $\{v_i\}$ , on cherche à déterminer un nouvel état d'équilibre, qui résulte de l'application des contraintes externes et internes en ces points. Aussi, on exprime ces contraintes par un ensemble d'équations que l'on résout ensuite, afin d'obtenir en chaque point  $v_i$  un vecteur déplacement  $\vec{d}_i$  qui correspond à la déformation locale.

Enfin, le but recherché étant de déformer le modèle numérique  $\mathcal{P}$  et donc de déterminer la déformation sur l'ensemble des points  $\{p_i\}$ , une dernière étape est nécessaire car l'ensemble des points  $\{p_i\}$  est en général plus dense que celui des points de contraintes  $\{v_i\}$ . On détermine alors une fonction de déformation sur l'espace  $\mathcal{S}$ , par interpolation des déplacements déjà connus. Il ne reste plus qu'à évaluer cette fonction en tous les points  $p_i$  du modèle numérique  $\mathcal{P}$  pour obtenir sa version déformée.

Pour résumer, nous pouvons décomposer tout processus de déformation d'un espace  $\mathcal{S}$ , contenant un modèle numérique  $\mathcal{P}$  (en figure 2.10.a), en trois étapes :

- le choix d'un ensemble de points de contraintes  $\{v_i\}$ , décrivant l'espace à déformer  $\mathcal{S}$ , de manière à y appliquer les contraintes externes (en rouge sur la figure 2.10.b) et internes (tous les points

sur la figure 2.10.b),

- le calcul de vecteurs déplacement  $\{\vec{d}_i\}$  représentant la déformation aux points de contraintes  $\{v_i\}$ , suite à l’application des contraintes externes et internes (sur la figure 2.10.c),
- la définition d’une fonction d’interpolation à partir de cet ensemble discret de déplacements  $\{\vec{d}_i\}$ , permettant de calculer la déformation, en tout point  $p_i$  de l’espace  $\mathcal{S}$ . Sur la figure 2.10.d, la déformation pour tous les pixels est obtenue par l’application de la fonction interpolante des déplacements  $\{\vec{d}_i\}$ .

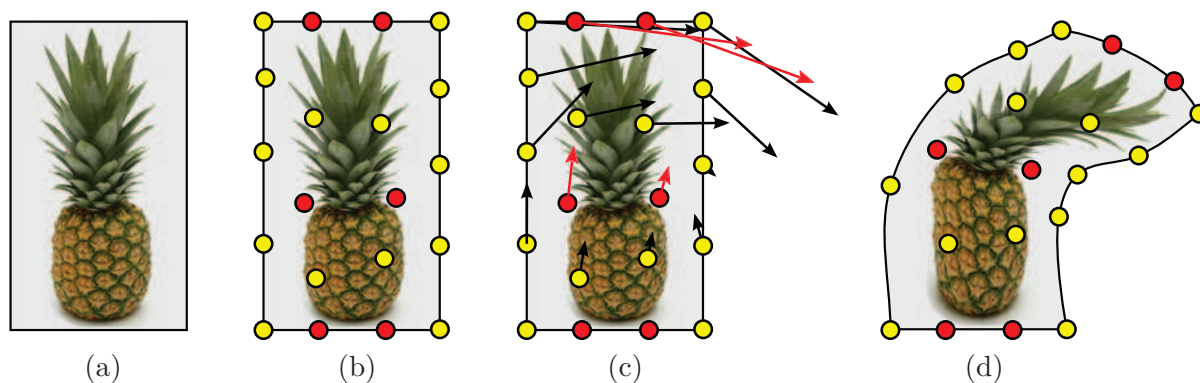


FIGURE 2.10 – **Étapes d’une déformation** : (a) Modèle  $\mathcal{P}$  dans un espace  $\mathcal{S}$ . Les  $\{p_i\}$  sont l’ensemble des pixels de l’image. (b) Choix de points de contraintes  $\{v_i\}$  (points jaunes et rouges) auxquels s’appliquent les contraintes internes. Sur le sous-ensemble de points  $\{c_i\}$  (en rouge) s’appliquent en particulier les contraintes externes. (c) Calcul de la déformation aux points  $\{v_i\}$ . (d) Interpolation de la déformation pour l’ensemble des pixels  $\{p_i\}$  de  $\mathcal{P}$ .

Nous présenterons dans la suite différentes approches permettant d’accomplir chacune de ces étapes, ainsi que des exemples d’application permettant d’illustrer ces approches.

### 2.2.1 Choix des points de contraintes

La première étape d’une opération de déformation consiste à choisir un ensemble de points  $\{v_i\}$ , auxquels s’appliquent les contraintes internes et externes qui décrivent la déformation souhaitée. Ce choix de points  $\{v_i\}$  est donc guidé par l’application imposant ces contraintes.

Considérant les applications typiques que l’on peut trouver dans la littérature, il existe trois principales méthodes pour déformer un modèle numérique  $\mathcal{P}$ . Dans la première catégorie, on déforme directement le modèle après avoir choisi les points de contrainte distribués sur celui-ci (tel que présenté en figure 2.11.a). Dans la seconde catégorie, on déforme l’espace  $\mathcal{S}$  englobant le modèle. Les points de contraintes sont alors choisis à l’extérieur du modèle (comme dans l’exemple de la figure 2.11.b). Enfin, dans la troisième catégorie, on déforme un squelette attaché au modèle  $\mathcal{P}$  (comme dans l’exemple de la figure 2.11.c).

Nous détaillons dans la suite ces trois méthodes de déformation et précisons le choix de points de contraintes qui en découle.

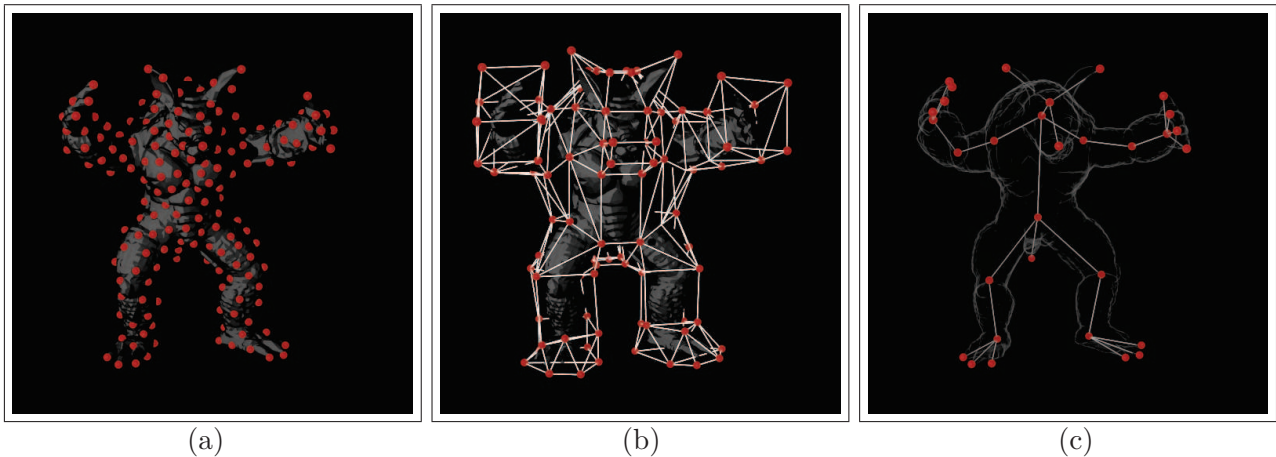


FIGURE 2.11 – **Choix de points de contraintes** : (a) Sur le modèle numérique. (b) À l’extérieur du modèle numérique. (c) À l’intérieur du modèle numérique.

### 2.2.1.1 Déformation directe du modèle numérique $\mathcal{P}$

Cette approche est généralement utilisée par les applications d’une part exigeantes concernant la manière de contrôler la déformation, et d’autre part cherchant à définir des déformations riches et complexes. Ce type d’application introduit des contraintes externes qui s’appliquent en diverses positions connues du modèle, et un jeu de contraintes internes détaillé. Le but est alors d’évaluer de la manière la plus précise possible la déformation sur  $\mathcal{P}$ .

Dans ce cas, les contraintes externes et internes sont directement appliquées à l’ensemble des points du modèle  $\mathcal{P}$ . Nous avons donc :  $\{v_i\} = \{p_i\}$ . Ce choix permet un contrôle total de la déformation, puisque chaque point de  $\mathcal{P}$  peut être soumis à une contrainte externe, ainsi que la possibilité de mettre oeuvre des déformations riches, puisque chaque  $\{p_i\}$  peut être déplacé. Nous illustrons sur la figure 2.12 une telle déformation riche et complexe par l’exemple d’une simulation déformant un carré de tissu [PTS09]. On remarque que le carré de textile rouge présente de nombreuses déformations locales de natures diverses, soulignant la nécessité de choisir tous les points du modèle numérique pour obtenir cette précision.



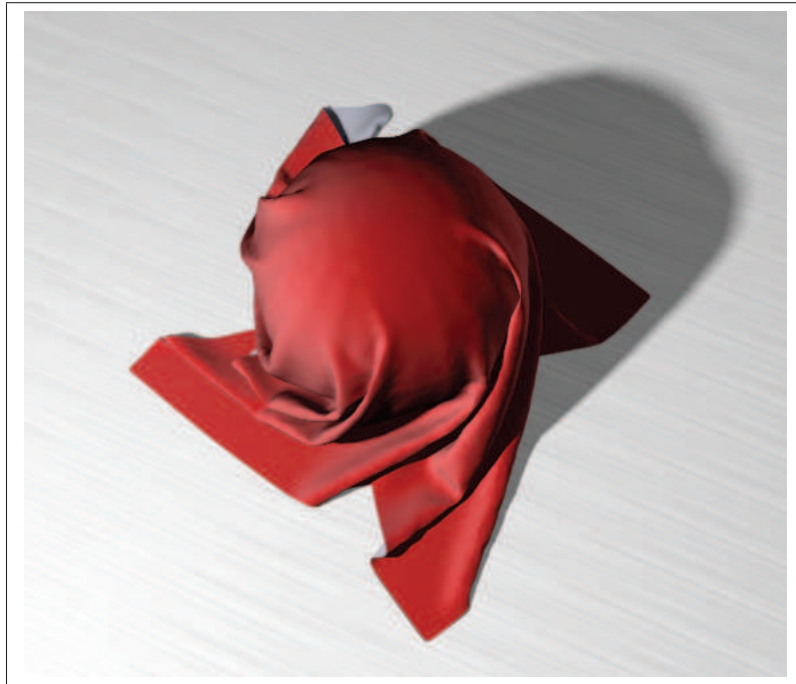


FIGURE 2.12 – Exemple d’une déformation riche en détails.

Il faut toutefois garder à l’esprit que le temps de calcul dépend fortement du nombre de points sur lesquels s’appliquent des contraintes. La complexité et la performance du processus d’optimisation dépend donc directement du nombre de points de contraintes. Il n’est donc pas rare de choisir les points de contraintes  $\{v_i\}$  tels qu’ils soient une décimation ou une approximation du modèle  $\mathcal{P}$ . On allège alors le coût de calcul au prix d’une perte de qualité du contrôle et de la précision de déformation. Dans de nombreuses applications toutefois, cette perte à des conséquences faibles voire négligeables. Un tel exemple est donné sur la figure 2.13, où les points de contraintes  $\{v_i\}$  choisis en rouge décrivent fidèlement l’aspect général du modèle  $\mathcal{P}$  tout en étant beaucoup moins nombreux que les points  $\{p_i\}$ . On remarque en effet que les points de contraintes ne représentent pas les détails fins de  $\mathcal{P}$ . Ce choix se révèle donc assez dense et suffisant pour une animation typique du modèle “Armadillo” (déplacement d’un bras...).

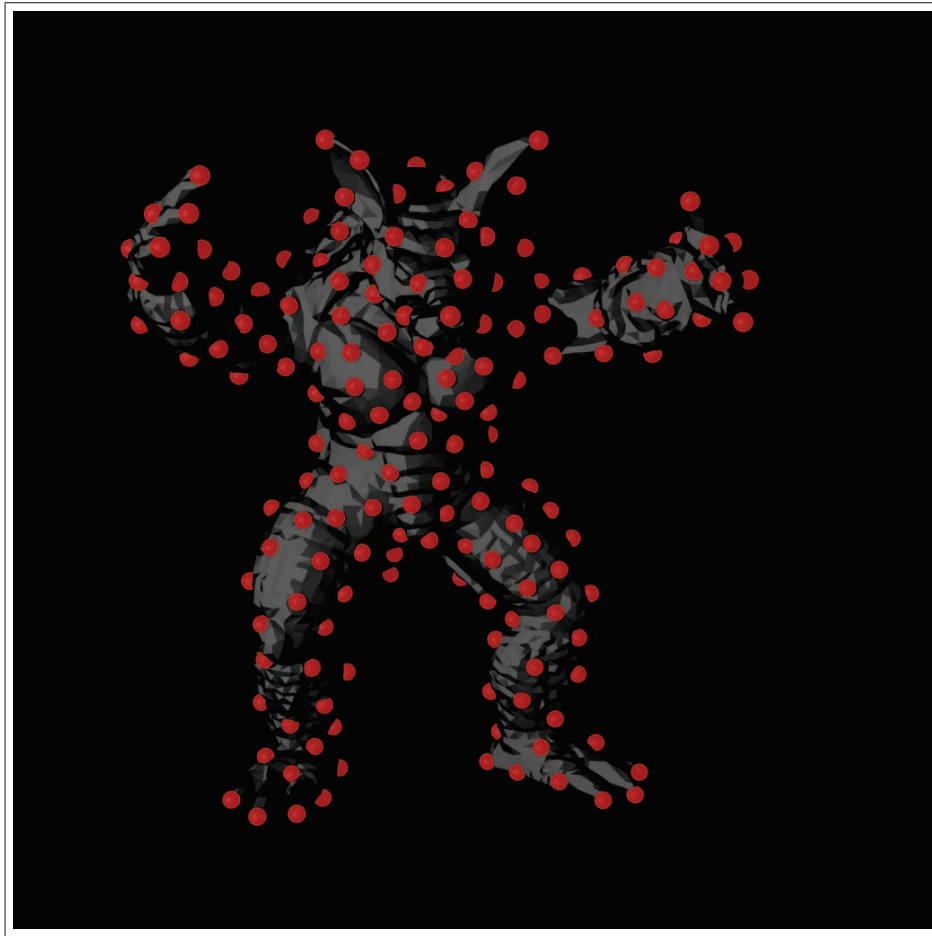


FIGURE 2.13 – Le modèle “Armadillo” tel que le choix de points de contraintes corresponde à une approximation (décimation) de  $\mathcal{P}$ .

### 2.2.1.2 Déformation de l’espace $\mathcal{S}$ englobant le modèle numérique $\mathcal{P}$

Les applications ayant un besoin d’interactivité, telles que la CAO ou l’animation, sont avant tout guidées par des considérations d’ergonomie et de performance afin de pouvoir être exécutées en temps réel. Elles utilisent des déformations faciles à contrôler et qui définissent un faible nombre de points de contraintes  $\{v_i\}$  pour être rapides.

L’approche dite “de déformation d’espace” propose d’abstraire le modèle numérique  $\mathcal{P}$  par l’espace  $\mathcal{S}$  qui l’englobe. Ce dernier se décrit et se manipule facilement par un petit nombre de points de contraintes  $\{v_i\}$ . Dans [Bar84] et [SP86], les auteurs s’intéressent à une méthode de déformation générale, indépendante du modèle. En conséquence, ils positionnent les points de contraintes  $\{v_i\}$  de manière à définir un treillis cubique de distribution spatiale homogène, dont la résolution est laissée au choix de l’utilisateur. Un exemple est illustré sur la figure 2.14.

Ce type de treillis cubique, régulier, convexe, permet de décrire facilement des transformations simples, c’est-à-dire pauvres en variations locales. En revanche, décrire une déformation riche implique

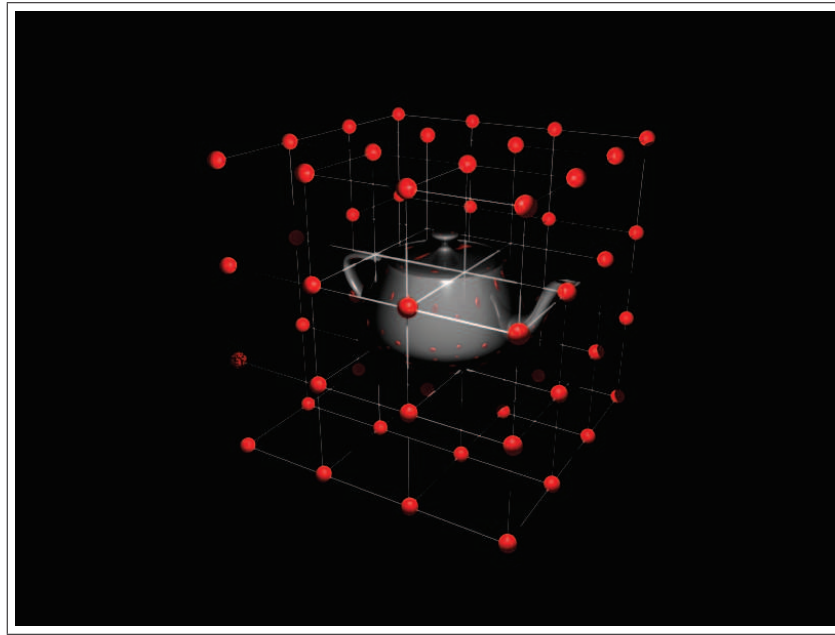


FIGURE 2.14 – Choix de  $\{v_i\}$  décrivant un treillis convexe cubique autour du modèle.

de relever la densité des points de contraintes, ce qui impacte négativement la performance en raison du coût de calcul. Aussi, d'autres approches proposent d'intervenir sur le positionnement de ces points, en prenant en compte la topologie du modèle. Elles permettent en particulier de définir des treillis convexes, que l'on appelle "cages" (voir figure 2.15). Dans [Coq90, MJ96] les auteurs s'affranchissent du cube et proposent de nouveaux types de mailles, tels les prismes ou les tétraèdres, et [JSW05] généralise cela à un choix de maille quelconque. Une cage permet un très bon contrôle du modèle à moindre coût. Toutefois, choisir la position des points de contraintes n'est pas trivial et rend la création de la cage difficile. Les modèles complexes souffrent particulièrement de ce point, comme le montre la figure 2.15, où un nombre important de points est nécessaire pour définir une cage s'adaptant à la géométrie d'un personnage.

Les techniques de déformation d'espace ont pour but de rendre ergonomiques les applications de déformations interactives (modélisation, animation...). Aussi dans ce type de déformations les points de contraintes ont pour vocation d'être manipulés. On leur impose alors une contrainte externe, se traduisant par un déplacement imposé (pouvant être nul si l'on désire avoir des points fixes). Il est fréquent que ces déplacements soient déterminés pour tous les points de contraintes choisis. Il s'agit typiquement de l'exemple d'une cage rudimentaire avec laquelle interagit un utilisateur. A la fin d'une session d'édition, le déplacement (nul ou non) est connu pour chaque point. Dans ce cas particulier, la déformation est déterminée et l'utilisation de contraintes internes n'a pas de sens.

Toutefois, afin de limiter le nombre de points pour contrôler la déformation, certains auteurs (par exemple [BHZN10]) introduisent des cages rigides traduisant l'énergie interne du modèle. Ainsi, imposer le déplacement d'un point de contraintes provoquera la déformation de la cage et donc le

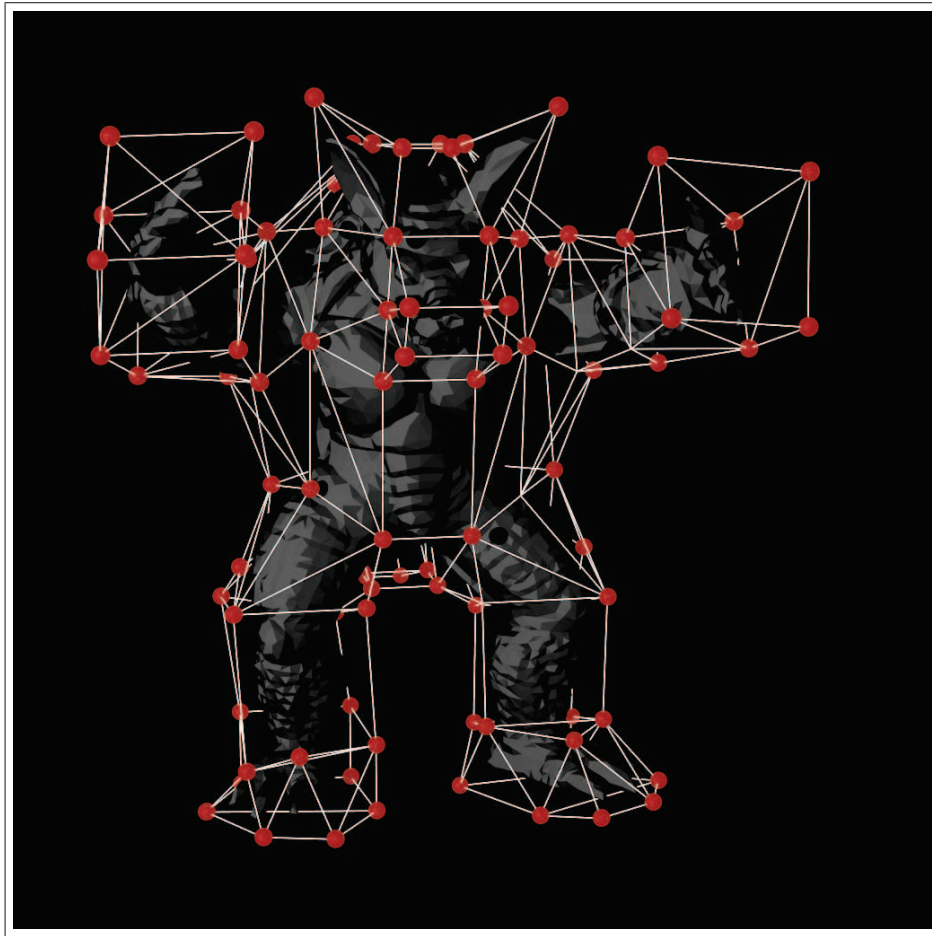


FIGURE 2.15 – Le modèle “Armadillo” approximé par un choix de points d’échantillonnage définissant une cage extérieure.

déplacement des autres points de contraintes. On se retrouve alors dans le cas classique, où des contraintes internes définissent la nature (rigide) de la déformation de la cage.

Finalement, ayant déformé l’espace  $\mathcal{S}$  par le biais de l’outil le décrivant (treillis, ou cage), on déterminera la déformation du modèle  $\mathcal{P}$  en interpolant la déformation du treillis ou de la cage. Ce processus sera détaillé dans la section 2.2.3.

### 2.2.1.3 Applications qui déforment le modèle $\mathcal{P}$ par un squelette

Dans les applications interactives qui déforment des modèles articulés (typiquement des personnages pour des animations), il est fréquent d'utiliser des "squelettes", plus simples à manipuler et performants que les cages. Un modèle rigide par parties est un assemblage de pièces rigides, où la déformation n'est possible que sur les articulations. Aussi, des approches proposent naturellement de positionner les points de contraintes  $\{v_i\}$  sur ces articulations [Kom88, MtLTM88, KŽ05, KCOZ06, FOKGM07] (figure 2.16).



FIGURE 2.16 – Le modèle “Armadillo” approximé par un choix de points de contrôle définissant un squelette.

Avec un objet articulé, le squelette présente de plus l'avantage de nécessiter moins de points qu'une cage, à contrôle de déformation équivalent. Son usage permet ainsi d'améliorer l'ergonomie et la performance. Ce point est illustré sur la figure 2.17, où la représentation de l'avant-bras d'Armadillo requiert 8 points pour le cas de la cage mais seulement 2 dans le cas du squelette. Animer l'avant-bras nécessite de déplacer 4 points dans le cas de la cage là où le squelette permet de n'en manipuler qu'un seul.

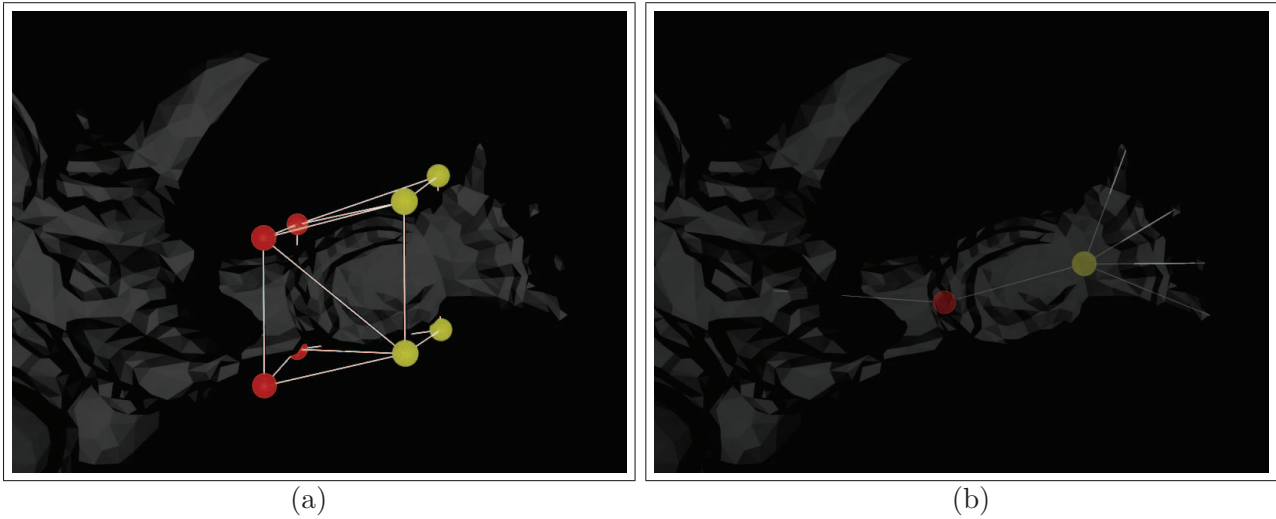


FIGURE 2.17 – **Comparaison de la méthode des cages avec la méthode du squelette.** (a) Les points de contraintes décrivent une cage. On doit déplacer les quatre points jaunes pour animer l’avant-bras. (b) En plaçant les points de contraintes sur les articulations, il n’y a plus qu’un point jaune à déplacer.

À l’instar des cages, la création manuelle d’un squelette peut se révéler une opération difficile dans le cas de modèles complexes. Aussi, il est possible de recourir à des méthodes d’extraction automatique. On peut citer des approches basées sur un diagramme de Voronoï [HBK01] ou par contraction de maillage [ATC<sup>+</sup>08].

De la même manière qu’avec les méthodes de déformation d’espace, contraintes externes et internes s’appliqueront aux points  $\{v_i\}$ . Les contraintes externes traduisent le mouvement introduit par l’utilisateur, tandis que les contraintes internes permettent d’assurer la rigidité de l’armature du squelette.

Enfin, les points  $\{v_i\}$  étant choisis à l’intérieur de l’enveloppe du modèle  $\mathcal{P}$ , ils décrivent  $\mathcal{P}$  partiellement et se révèlent insuffisants pour interpoler la déformation en chaque  $p_i$ . Pour traiter ce problème on peut mentionner la technique du *skinning* [LCF00], dont l’idée consiste à introduire quelques points de contraintes  $\{v_i\}$  sur  $\mathcal{P}$  de manière à obtenir une description suffisante de  $\mathcal{P}$ . Des contraintes internes permettent alors de déplacer ces points conformément aux différentes déformations du squelette. Pour obtenir la déformation finale, il ne reste plus alors qu’à utiliser une méthode d’interpolation parmi celles présentées dans la section 2.2.3.

## 2.2.2 Calcul du déplacement à tous les points de contraintes

Étant donné un ensemble de points de contraintes  $\{v_i\}$ , nous souhaitons maintenant calculer pour chacun d’eux le vecteur déplacement  $\vec{d}_i$  qui indique la déformation. Cette étape consiste à déterminer la fonction  $f_{D_v}$  telle que :

$$\begin{aligned}
f_{D_v} : \quad v_i &\rightarrow v'_i \\
&v_i \rightarrow v_i + \vec{d}_i
\end{aligned}
\tag{2.1}$$

Nous avons introduit précédemment un ensemble de contraintes (externes et internes) qui s'appliquent précisément en ces points  $\{v_i\}$ . L'objectif est à présent de traduire ces contraintes en équations, pour y calculer la déformation.

Si une résolution directe de ces équations n'est pas possible car le système n'est pas linéaire, on décrit la déformation par un modèle énergétique initialement à l'équilibre que les contraintes externes viennent perturber. Cela se traduit par une fonction énergie que l'on souhaite minimiser pour établir le nouvel équilibre du système.

Les contraintes externes et internes décrivent la déformation par un ensemble de lois faisant intervenir des grandeurs associées à chaque point de contrôle. Il est fréquent que plusieurs applications, à-priori sans rapport, décrivent des déformations similaires. On peut citer par exemple la simulation chirurgicale et l'industrie du textile, qui font toutes deux usage de déformations de nature élastique. Il est donc intéressant d'étudier les différentes natures de déformation.

Suivant les grandeurs utilisées par les contraintes internes et externes, on retient deux types de déformations : les déformations géométriques (uniquement basées sur des grandeurs géométriques, telles les longueurs, angles...) et les déformations physiques (faisant intervenir des grandeurs physiques, telles la masse, la vitesse...). Nous détaillons à présent ces deux types.

### 2.2.2.1 Déformation sous contraintes géométriques

Les déformations géométriques se retrouvent dans l'ensemble des applications ne faisant pas intervenir de grandeur physique. Par exemple, lorsque l'on effectue un recalage d'image, on travaille avec un ensemble de points à déplacer, et une géométrie de l'image à déformer. Aussi, la déformation ne peut se décrire que par une loi uniquement géométrique, telle la conservation des proportions par exemple.

Malgré cette restriction, il est cependant à noter que ce type de déformation s'utilise pour approximer de manière crédible et à moindre coût certaines simulations physiques. On peut par exemple citer des déformations dans le domaine de l'animation. On utilise alors des contraintes géométriques comme heuristiques d'une loi physique. Par exemple, on peut simuler un critère mécanique de rigidité en l'approximant par un critère géométrique de conservation des aires.

Les déformations géométriques n'impliquent que des critères basés sur les positions d'un ensemble de points (la distance entre deux points par exemple). On peut toutefois citer quelques autres critères intéressants. Des travaux se sont intéressés à définir et préserver un critère de forme. [SCOL<sup>+</sup>04] et [LSCO<sup>+</sup>04] utilisent à ces fins l'opérateur du *Laplacien*, définissant la position de chaque point comme une combinaison de ses voisins. Cet opérateur est obtenu par des formulations proches telles les *mean value coordinates*, qui généralisent les coordonnées barycentriques [Flo03], ou encore les



*pyramid coordinates* [SK04], qui rendent l'opérateur insensible aux rotations. Ces travaux ont ensuite été enrichis par de nouveaux critères de conservation tels que la préservation des volumes [HSL<sup>+</sup>06], des aires [WXW<sup>+</sup>06], ou encore la simulation de la rigidité en minimisant les rotations locales.

### 2.2.2.2 Déformation sous contraintes physiques

Par opposition aux méthodes géométriques, une déformation physique fait intervenir des grandeurs physiques, qui sont variables selon les applications. Dans le domaine médical, les simulations chirurgicales modélisent la déformation des différents tissus du corps humain. Elles intègrent donc les propriétés élastiques ou viscoélastiques de ces tissus. Dans l'aéronautique, les simulations d'aéroélasticité étudient l'interaction entre forces aérodynamiques, élastiques et inertielles. On introduit en particulier des grandeurs de masse, vitesse ou densité. Enfin dans les géosciences, les méthodes physiques de restauration font appel à des grandeurs géomécaniques, telles la porosité, ou la compression de la roche. On peut obtenir un bon aperçu de l'ensemble des méthodes de déformation physique par exemple dans [GM97] et [NMK<sup>+</sup>06].

Les simulations physiques proposent donc de décrire des déformations réalistes (conformes à des observations dans la réalité) par des modèles plus précis que les méthodes géométriques. Cette précision s'accompagne toutefois d'une augmentation du coût en raison de l'obligation d'utiliser un jeu de données supplémentaires : des grandeurs physiques associées aux points de contraintes  $\{v_i\}$ .

### 2.2.3 Interpolation de la déformation pour l'ensemble du domaine

Une fois les vecteurs déplacement  $\{\vec{d}_i\}$  calculés pour l'ensemble des points de contraintes  $\{v_i\}$ , la déformation de l'espace  $\mathcal{S}$  ne sera complète que si nous pouvons estimer un vecteur déplacement  $\vec{d}$  pour tout point  $p$  de  $\mathcal{S}$ .

Cette opération consiste alors à définir une fonction  $D$  définie sur  $\mathcal{S}$ , qui à chaque point  $p$  de  $\mathcal{S}$  associe un vecteur déplacement  $\vec{d}$ , et honore les vecteurs  $\{\vec{d}_i\}$  préalablement estimés aux points  $\{v_i\}$ . Ainsi définie, la fonction  $\mathcal{D}$  est une fonction interpolante vérifiant :

$$D(v_i) = \vec{d}_i. \quad (2.2)$$

L'interpolation est un sujet largement traité dans la littérature, on peut par exemple se référer à [Fra82, Pie91, LH11, Bax92] pour avoir une vue d'ensemble des méthodes existantes. Dans ce paragraphe nous présenterons les méthodes les plus couramment utilisées dans le cadre de méthodes de déformation.

Nous pouvons séparer ces approches en deux grandes familles : les interpolants globaux et les interpolants locaux. Un interpolant est dit "local" lorsque l'ajout ou la suppression d'un couple { point ; valeur } ne modifie la valeur interpolée que sur une zone localisée du domaine de définition (à proximité du point inséré ou supprimé). Au contraire, avec un interpolant dit "global", la



valeur interpolée sera altérée sur l'ensemble du domaine de définition.

### 2.2.3.1 Interpolants locaux

Pour estimer  $D(p)$ , on définit un sous-ensemble de points  $\{v_i\}$  au voisinage  $\mathcal{N}(p)$  de  $p$ .  $D(p)$  est alors défini comme une fonction des  $\{\vec{d}_i\}_{i \in \mathcal{N}(p)}$  et tel que :

$$D(p) = f(\{\vec{d}_i\}_{i \in \mathcal{N}(p)}). \quad (2.3)$$

Ce voisinage  $\mathcal{N}(p)$  peut être fonction de la distance, comme illustré sur la figure 2.18.a, (où l'on ne garde que les points  $\{v_i\}$  contenus dans le cercle), ou bien fonction de la topologie d'un maillage. Sur la figure 2.18.a, on ne garde que les points  $\{v_i\}$  formant la maille contenant  $p$ .

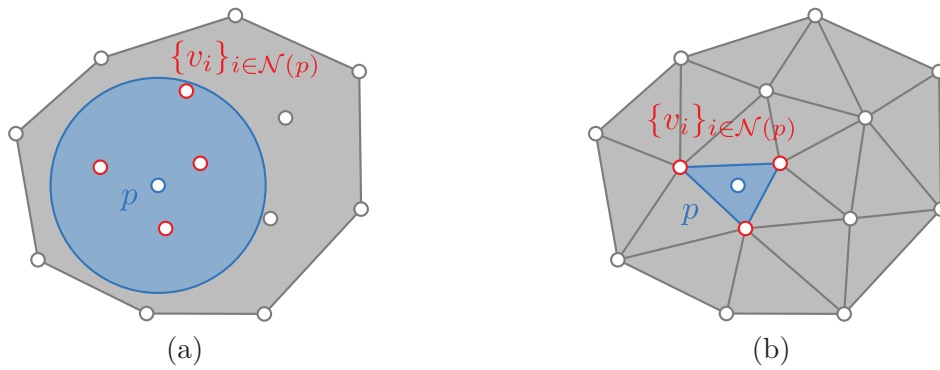


FIGURE 2.18 – Définition du voisinage  $\mathcal{N}(p)$  d'un point  $p$  : (a) en fonction de la distance, (b) en fonction de la topologie d'un maillage.

#### a. Interpolants linéaires

Parmi les interpolants locaux, la sous-classe la plus simple regroupe l'ensemble des interpolants linéaires. Soit un ensemble  $\{v_i\}$  de positions auxquelles sont associées les valeurs  $\{d_i\}$ , un interpolant linéaire définit la valeur  $D(p)$  en un point  $p$  quelconque comme une combinaison linéaire des valeurs  $d_i$  pondérées par les coefficients  $\omega_i$  de telle façon que :

$$D(p) = \sum_{i \in \mathcal{N}(p)} \omega_i \cdot d_i. \quad (2.4)$$

Parmi les différentes méthodes classiquement utilisées pour déterminer ces coefficients  $\omega_i$ , on peut mentionner en premier lieu les méthodes de pondération par la distance. L'idée principale consiste à déterminer la valeur  $D(p)$  au point  $p$  de manière à ce que seuls les points  $v_i$  proches aient un impact. On définit alors  $\omega_i$  comme étant une fonction décroissante de la distance à  $p$  :

$$\omega_i = f(\|\mathbf{p} - \mathbf{v}_i\|). \quad (2.5)$$

Les fonctions les plus classiques sont la pondération inverse à la distance [She68], ou bien la fonction gaussienne :

$$\omega_i = \frac{1}{1 + \|\mathbf{p} - \mathbf{v}_i\|} \quad \text{ou} \quad \omega_i = e^{-\frac{\|\mathbf{p} - \mathbf{v}_i\|^2}{2\sigma^2}} \quad (2.6)$$

, ou  $\sigma$  est une constante choisie qui dénote l'écart type.

Parmi les interpolants linéaires, on peut également citer l'interpolation polygonale (ou coordonnées barycentriques [MBLD02]), typiquement utilisés dans le cas d'un maillage. On détermine alors les coefficients  $\omega_i$  en fonction de la position du point  $p$  contenu dans la maille formée des points  $\{v_i\}, \forall i \in \mathcal{N}(p)$ . Sur la figure 2.19, plusieurs types de mailles sont présentés. À chaque fois, la valeur du coefficient  $\omega_i$  associé à un point  $v_i$  est déterminée en calculant l'aire  $\mathcal{A}_i$  de même couleur que le point  $v_i$  considéré. Avec certains polygones particuliers tels les triangles ou rectangles, il existe des manières directes pour déterminer ces aires  $\mathcal{A}_i$  (on choisit l'aire opposée sur les figures 2.19.a et 2.19.b). Pour des polygones quelconques, on peut utiliser la méthode des voisins naturels (figure 2.19.c).

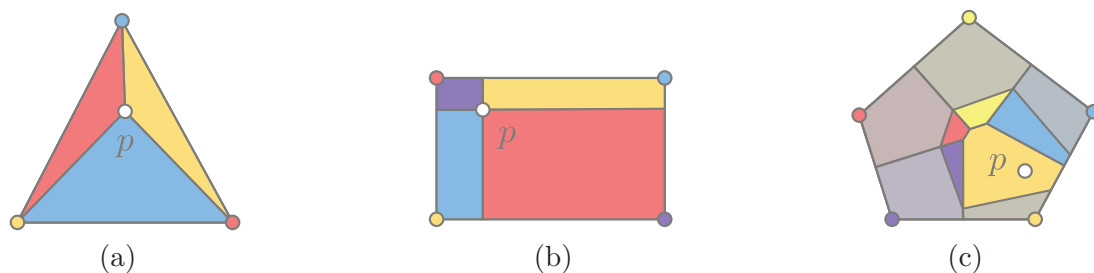


FIGURE 2.19 – **Interpolation polygonale.** (a) Interpolation barycentrique dans un triangle. (b) Interpolation bilinéaire dans un rectangle. (c) Interpolation barycentrique dans un polygone quelconque par la méthode des voisins naturels.

Pour ces approches, ayant défini l'aire  $\mathcal{A}_i$  associée au point  $v_i$ , le poids  $\omega_i$  est défini par :

$$\omega_i = \frac{\mathcal{A}_i}{\sum_{j \in \mathcal{N}(p)} \mathcal{A}_j} \quad (2.7)$$

Plus récemment, des optimisations ont été proposées afin de fournir des interpolants ayant de bonnes propriétés de continuité et de dérivabilité. Par exemple, les *mean value coordinates* [Flo03, JSW05] définissent au point  $p$  le coefficient  $\omega_i$  tel que :

$$\omega_i = \frac{\lambda_i}{\sum_{j \in \mathcal{N}(p)} \lambda_j}, \quad \lambda_i = \frac{\tan(\alpha_{i-1}/2) + \tan(\alpha_i/2)}{|p - v_i|} \quad (2.8)$$

où  $\alpha_i$  est l'angle  $\widehat{v_{i+1}v_i p}$ , ainsi qu'illustré dans la figure 2.20.

Dans la méthode des *harmonic coordinates* [DM06, JMD<sup>+</sup>07], les auteurs proposent de déterminer chaque coefficient  $\omega_i$  par une fonction harmonique  $h_i(p) = \omega_i$  garantissant  $\nabla^2 h_i(p) = 0$  (équation de

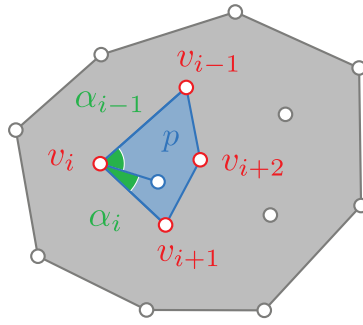


FIGURE 2.20 – Interpolation avec les *mean value coordinates*.

*Laplace*). L'intérêt de cet interpolant est de respecter au mieux la forme décrite par les points de contraintes, ce qui le rend intéressant pour les applications effectuant des déformations par cage.

On peut enfin citer les *green coordinates* [LLCO08], où les auteurs développent l'approche de déformation par cage respectueuse de la forme, introduite par les *harmonic coordinates*. Ils proposent alors un opérateur approximant qui privilégie le respect de la forme du modèle à déformer par rapport à la cage qui le déforme. La figure 2.21.c illustre le résultat d'une telle application, et l'on remarque que le modèle déformé ne respecte plus la cage. Sur la figure 2.21.b, on compare ce résultat avec la méthode des *harmonic coordinates*, et l'on peut remarquer que malgré le respect de la cage, la déformation obtenue est visuellement moins plaisante.

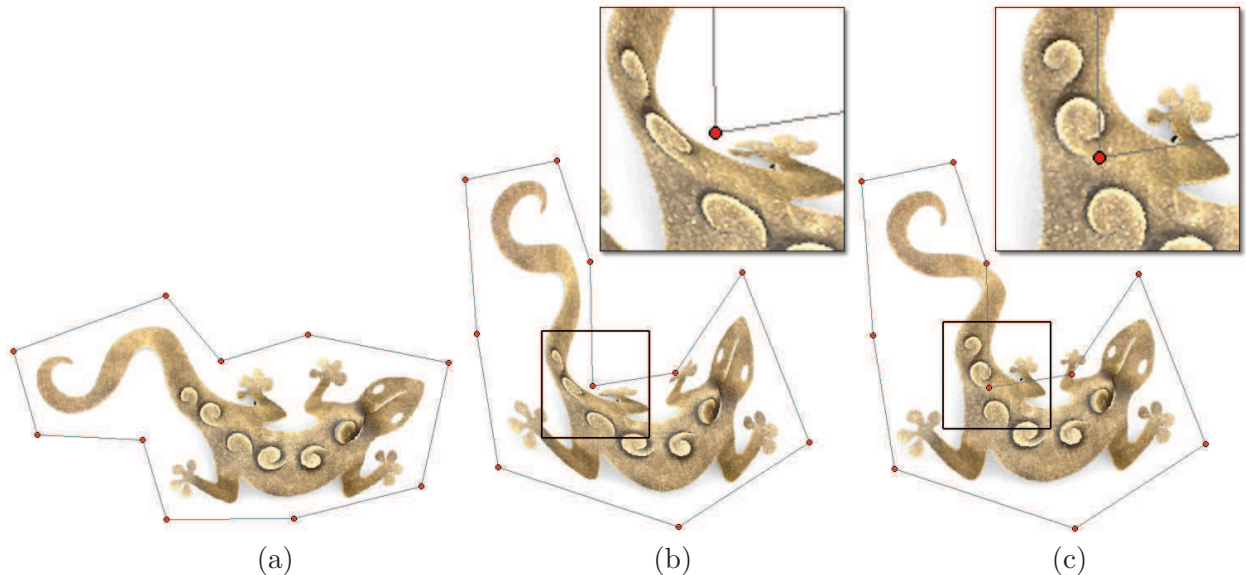


FIGURE 2.21 – Comparaison entre les *harmonic coordinates* et les *green coordinates* lors d'une déformation par cage. (a) Image initiale (b) Image déformée illustrant les *harmonic coordinates* (c) Image déformée illustrant les *green coordinates*. On remarque que la forme de l'image est mieux respectée dans ce dernier cas qu'avec les *harmonic coordinates*. Toutefois, l'image ignore désormais les frontières de la cage.

## b. Interpolants polynomiaux (splines et NURBS)

Les *splines* sont un autre exemple d'interpolant local. Ils permettent de définir une trajectoire ou une surface par la fonction qui passe par l'ensemble fini de points de contraintes  $\{v_i\}$  (figure 2.22.a) par l'écriture d'un polynôme. Elles sont typiquement utilisées dans le domaine de la modélisation [SP86, Coq90, MJ96].

Afin de simplifier leur définition ainsi que leur manipulation, on les généralise par les “courbes de Bézier” qui possèdent la particularité d'être définies par un ensemble supplémentaire de points, dits de contrôle, positionnés à l'extérieur de la trajectoire interpolée. Ces points de contrôle sont aisés à manipuler par un utilisateur et permettent ainsi de définir facilement un arc de la forme souhaitée. Nous montrons un exemple de courbe de Bézier sur la figure 2.22.b, où les points  $\{v_i\}$  sont en gris, et les points de contrôle en rouge.

Enfin, les *B-Splines* ou *NURBS* (*Non Uniform Rational B-Splines*), généralisent ces courbes de Bézier. Elles permettent de les composer grâce à une définition par parties et peuvent ainsi définir des trajectoires riches et complexes (figure 2.22.c).

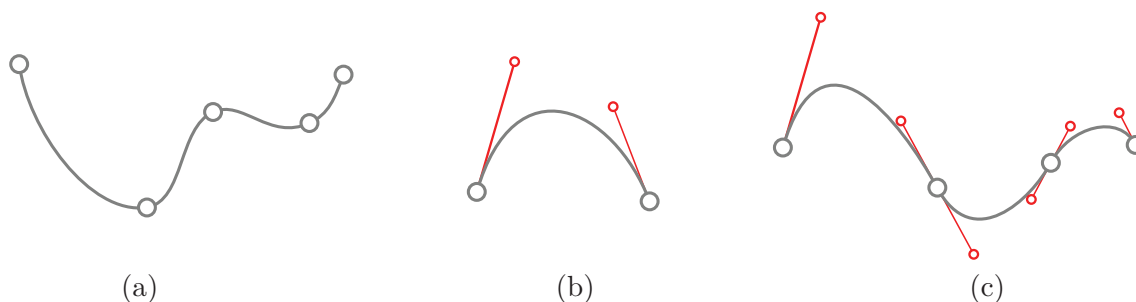


FIGURE 2.22 – **Interpolation polynomiale.** (a) Spline : fonction polynomiale passant par les points  $\{v_i\}$ . (b) Courbe de Bézier : fonction polynomiale définie par les points  $\{v_i\}$  (en gris) et les points de contrôle (en rouge). (c) NURBS (B-Splines) : composition de courbes de Bézier.

### 2.2.3.2 Interpolants globaux

#### a. RBF (Radial Basis Functions)

Les *Radial Basis Functions* (RBF) sont un interpolant global. Initialement utilisées dans des applications de reconstruction de surface [TO99, CBC<sup>+</sup>01], on les trouve également dans le domaine de la déformation [KSSH02, BK05]. Elles visent à définir la fonction la plus lisse possible passant par l'ensemble de points  $\{v_i\}$  auxquels sont valuées les valeurs  $d_i$ . On définit une interpolation au point  $p$  des valeurs  $d_i$  aux points  $v_i$  par :

$$D(\mathbf{p}) = \sum_{i=0}^n \omega_i \cdot \phi(\|\mathbf{p} - \mathbf{v}_i\|) + \Pi(\mathbf{p}), \quad (2.9)$$

où  $\{\omega_i\}$  correspond à l'ensemble de pondérations à déterminer qui dépendent de  $d_i$ ,  $\Pi(p)$  est un polynôme du premier degré (dont les coefficients sont également à déterminer), et  $\phi(r)$  est une fonction

radiale centrée en chaque  $v_i$ , qui dépend de la distance à  $p$ .

Il existe plusieurs types de fonctions radiales utilisées classiquement. Parmi les plus fréquentes, on peut citer la gaussienne :

$$\phi(r) = e^{-(cr)^2}, \text{ avec } : c \in \mathbb{R}, \quad (2.10)$$

la bi-harmonique (*thin plate*) :

$$\phi(r) = r^2 \cdot \log(|r|), \text{ avec } : r \neq 0, \quad (2.11)$$

ou encore la multi-quadratique :

$$\phi(r) = \sqrt{r^2 + c^2}, \text{ avec } : c \in \mathbb{R}. \quad (2.12)$$

L'obtention des poids  $\{\omega_i\}$  s'obtient alors par le respect de la contrainte :

$$\begin{aligned} D(v_i) &= d_i \\ &= \sum_{i=0}^n \phi(\mathbf{p} - \mathbf{v}_i) + \Pi(\mathbf{p}). \end{aligned} \quad (2.13)$$

Et les  $\{\omega_i\}$  sont solution d'un système linéaire dont le coût est directement proportionnel au nombre de points. Si l'obtention des poids peut être coûteuse pour un grand nombre de points, son évaluation est en revanche très rapide.

## 2.3 Choix de nos solutions

Dans la section précédente, nous avons résumé la façon dont est classiquement décomposé un problème de déformation en trois étapes qui sont : le choix d'un ensemble de points de contraintes  $\{v_i\}$ , l'estimation de déplacements  $\{\vec{d}_i\}$  en ces points de contraintes, et l'interpolation de ces déplacements sur l'ensemble de l'espace à déformer  $\mathcal{S}$ . Nous allons à présent introduire et justifier nos choix pour nos propres travaux concernant chacune de ces étapes, c'est-à-dire dans le cas de la déformation d'une image sismique à restaurer.

### 2.3.1 Choix d'un ensemble de points de contraintes

La restauration d'une image sismique comme simplifiée sur la figure 2.23.a impose des contraintes externes qui s'appliquent aux failles (en rouge) et horizons (en bleu) interprétés. La restauration de la faille consiste en effet à faire glisser une partie de l'image le long de la faille de manière à fusionner les points des extrémités de l'horizon. Puis la mise à plat de l'horizon modifie la géométrie de l'horizon considéré afin d'obtenir une droite dont la position verticale est choisie. Aussi, les points décrivant failles et horizons sont sujets à des contraintes externes et doivent être des points de contraintes (points rouges sur la figure 2.23.b).

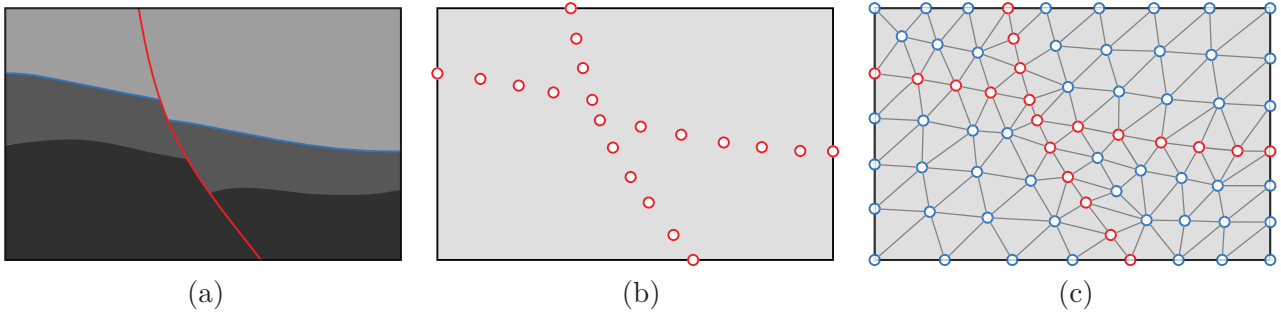


FIGURE 2.23 – **Choix des points de contraintes selon les contraintes externes et internes.** (a) Image sismique à restaurer contenant une faille et un horizon. (b) Choix de points de contraintes (en rouge) sur la faille et sur l’horizon pour évaluer les contraintes externes. (c) Choix de points de contraintes supplémentaires (en bleu) sur le reste de l’image pour évaluer les contraintes internes.

En plus des contraintes externes, nous imposons des contraintes internes permettant une préservation de la rigidité du modèle. Pour cela, nous introduisons un nouvel ensemble de points de contraintes. Ces points doivent échantillonner l’image de la façon la plus homogène.

Une manière habituelle d’obtenir une distribution homogène de points sur un espace consiste à générer un maillage isotrope. De plus, si celui-ci doit être conforme à une géométrie quelconque (dans notre cas, les failles et horizons), la triangulation se révèle un bon choix, puisque dans le domaine du maillage le triangle est la primitive permettant de s’adapter au mieux à tout type de géométrie. Aussi, nous générons sur  $\mathcal{S}$  une triangulation conforme aux géométries des failles, des horizons, ainsi qu’à la géométrie de l’image à restaurer. Nous obtenons alors sur la figure 2.23.c un nouvel ensemble de points (en bleu). Finalement, tous les noeuds de cette triangulation sont des points de contraintes.

### 2.3.2 Estimation des déplacements

Ayant un ensemble  $\{v_i\}$  de points de contrôle décrivant l’image sismique à restaurer, ses failles et ses horizons, nous définissons à présent la manière dont cette image sera déformée. En d’autres termes, nous choisissons les contraintes s’appliquant aux  $\{v_i\}$  qui permettent de restaurer l’image.

Dans la section 1, nous avons introduit le processus de restauration et précisé que nous cherchons à mettre en oeuvre une méthode qui produira une déformation rigide. Ce processus de restauration vise à être un outil d’aide à l’interprétation, et intervient ainsi dans une phase où la connaissance du sous-sol étudié est faible. En particulier, on ne dispose dans ce contexte que d’une interprétation préliminaire et d’aucune donnée physique. Pour cette raison, nous estimerons une déformation rigide en nous basant uniquement sur des propriétés géométriques.

Le chapitre 3 précisera en détail la manière dont nous traduirons une déformation rigide géométrique et les contraintes de restauration par des équations. Finalement, en résolvant ces équations nous obtiendrons un ensemble  $\{\vec{d}_i\}$  de déplacements, associés aux points  $\{p_i\}$ .

### 2.3.3 Interpolation

Ayant choisi un ensemble de points de contraintes  $\{v_i\}$ , puis calculé en ces points le déplacement  $\vec{d}_i$  qui correspond à la déformation locale, nous chercherons alors un interpolant permettant de calculer le déplacement en n'importe quel pixel de l'image sismique à restaurer. Étant donné que les points de contraintes choisis décrivent un maillage, l'interpolant retenu doit être adapté à une telle structure. En particulier, nous souhaitons que l'altération d'une maille donnée, ne modifie pas l'ensemble de l'interpolant. Le calcul doit également être suffisamment rapide. Enfin, nous n'avons pas besoin d'un interpolant possédant des propriétés particulières supplémentaires telles que la dérivabilité sur l'ensemble de l'image.

Nous pensons donc que l'interpolant le mieux adapté à ces prérequis est une interpolation barycentrique (interpolation polygonale). En effet, cette solution est bien adaptée au cas d'un maillage, et elle est de plus locale, simple et rapide.

## Chapitre 3

# Présentation de notre algorithme de restauration

*Ce chapitre a été publié sous la référence [GGJK13] : Mathieu Gilardet, Sébastien Guillon, Bruno Jobard and Dimitri Komatitsch, Seismic image restoration using nonlinear least squares shape optimization, Procedia Computer Science, vol. 18, p. 732–741 (2013).*

Ayant rappelé dans le chapitre 1 que la restauration d’une image sismique consiste à la déformer, nous avons présenté dans le chapitre 2 un aperçu du domaine de la déformation puis proposé un formalisme que nous allons utiliser pour effectuer cette opération.

Ce formalisme nous a permis d’introduire une à une les différentes étapes qui interviennent dans la déformation d’une image sismique. Nous avons pour chaque étape effectué le choix d’une approche qui conviendra pour nos travaux puis justifié ce choix. Pour rappel, nous déterminons tout d’abord des points de contraintes  $\{v_i\}$  en générant sur l’image sismique une triangulation adaptée aux failles et horizons interprétés. Nous calculons ensuite en ces points la déformation par optimisation des contraintes externes de restauration de la faille ou de l’horizon ainsi que des contraintes internes qui produisent une déformation rigide. Pour cela, nous imposons à notre méthode de respecter un certain nombre de grandeurs géométriques telles que les aires et les formes. Dans une dernière étape nous utilisons une méthode d’interpolation barycentrique pour transmettre la déformation du maillage à l’image sismique et ainsi achever de la restaurer.

Détaillons à présent le processus d’optimisation, c’est-à-dire la manière dont nous traduisons les contraintes en équations puis dont nous les résolvons pour calculer la déformation.

La section 3.1 ouvrira ce chapitre en décrivant le processus général restaurant une image sismique. Elle introduira une méthode calculant le maillage déformé qui minimise localement ses variations de forme et d’aire et qui honore les contraintes externes liées à la restauration. La section 3.2 poursuivra en détaillant l’écriture des différentes contraintes (internes et externes) intervenant dans notre méthode. Dans la section 3.3, nous proposerons deux méthodes d’optimisation de ces contraintes puis discuterons de leurs avantages et inconvénients en section 3.4. Pour finir dans les sections suivantes,



nous aborderons de manière plus précise certains aspects mathématiques, informatiques et techniques liés à l’ensemble du processus qui sont très importants en pratique et qui donc mériteront que nous rentrions dans les détails de manière plus poussée. Précisément, la section 3.5 traitera d’un raffinement du processus d’optimisation qui permet de pré-conditionner certaines étapes afin de les accélérer et d’éviter certains résultats faux. Puis, nous étudierons dans la section 3.6, la convergence de l’algorithme itératif utilisé et proposerons des critères d’arrêt automatique. Enfin, dans la section 3.7, nous détaillerons l’impact du maillage dans ce processus de déformation et proposerons des pistes pour déterminer un maillage optimisé à chaque scénario de déformation.

### 3.1 Processus de restauration

Dans cette section, nous décrivons le principe général de fonctionnement d’une méthode de déformation quasi-rigide de maillage appliquée à la restauration d’une image sismique. Une telle méthode repose essentiellement sur des contraintes internes qui assurent la préservation de grandeurs géométriques telles que les aires, mais également les formes. On peut trouver dans la littérature des méthodes proposant des déformations respectant ces grandeurs, appliquées dans d’autres domaines. Par exemple les travaux de [WXW<sup>+</sup>06] issus du domaine de l’animation permettent de déformer une image en respectant formes et proportions. Nous préciserons lors de l’introduction des contraintes internes, en quoi cette approche revient à conserver aires et longueurs. Pour restaurer une image, la méthode de déformation repose également sur des contraintes externes qui introduisent l’objectif à atteindre. Nous décrirons dans cette section la manière dont ces différentes contraintes interviennent et les résultats qu’elles produisent.

La méthode de déformation rigide d’un maillage calcule la déformation en un ensemble de points de contraintes  $\{v_i\}$  appartenant au maillage considéré  $\mathcal{M}$ . Nous rappelons que nous avons choisi dans la section 2.3 ces points  $\{v_i\}$  de manière à ce que leur distribution spatiale soit homogène et qu’ils constituent ainsi une représentation échantillonnée de l’image sismique (c’est-à-dire dans notre formalisme notre modèle numérique  $\mathcal{P}$  constitué des pixels  $\{p_i\}$ ). Calculer la déformation va donc consister à optimiser les contraintes externes et internes (en honorant les premières et respectant au mieux les secondes) pour obtenir en tout point  $v_i$  un vecteur déplacement à appliquer ensuite à l’image.

Nous rappelons également que la section 2.2 a défini le fait que les contraintes externes et internes vont s’appliquer aux points  $\{v_i\}$  du maillage. Les contraintes externes ont pour but de remplir les objectifs d’un scénario de restauration : selon le cas, dans notre contexte de restauration d’image sismique, il s’agit soit de la restauration d’une faille soit de la mise à plat d’un horizon. Nous utiliserons dans la suite l’expression “contraintes de restauration” pour les désigner. Ces contraintes de restauration permettent d’obtenir la déformation sur un ensemble des points  $\{c_i\}$  inclus dans  $\mathcal{V}$ . Afin de propager cette déformation à l’ensemble des points  $\{v_i\}$  tout en conservant une déformation rigide, dans la section 2.2 nous avons introduit les contraintes internes. Étant donné que ces contraintes internes travaillent avec des grandeurs géométriques, nous les appellerons “contraintes géométriques”

dans la suite.

Introduisons à présent de façon plus détaillée le processus de restauration. Afin de résumer l'ensemble du processus, nous utiliserons l'exemple synthétique de la figure 3.1.a dans lequel une faille (en rouge) et un horizon (en bleu) ont été interprétés. L'objectif est dans un premier temps d'inverser le glissement produit le long de la faille, puis de mettre à plat l'horizon.

### 3.1.1 Restauration d'une faille

Pour restaurer le glissement le long d'une faille, un bloc mobile est défini d'un coté de celle-ci tandis que le deuxième bloc situé de l'autre coté de la faille demeure statique. Nous commençons par générer un maillage de la partie mobile par une triangulation s'adaptant à son contour (figure 3.1.b.). Avant cette opération, nous nous assurons d'introduire le point d'intersection  $v_I^0$  de l'horizon bleu sur la partie mobile avec la faille. En effet, l'opération de restauration de faille doit assurer que  $v_I^0$  se déplace au point correspondant  $v_I$  sur la partie statique.

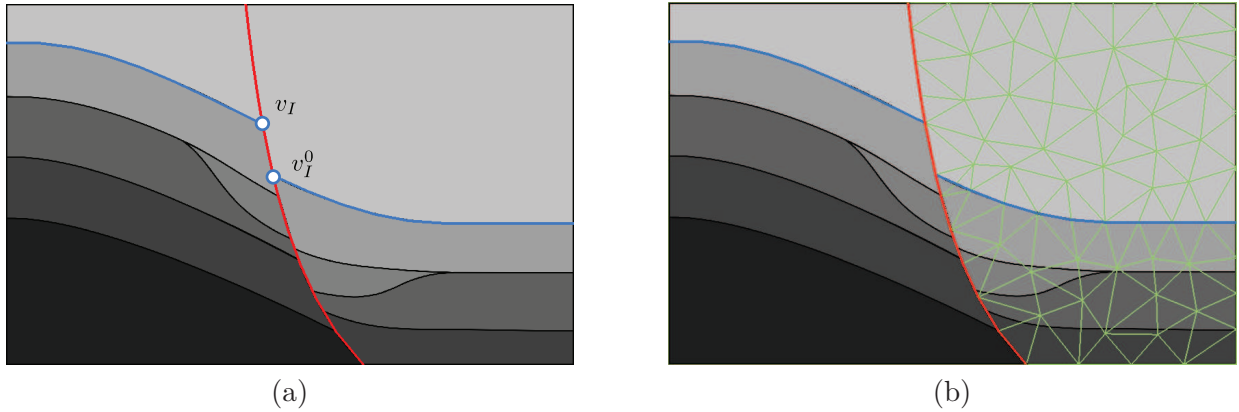


FIGURE 3.1 – **Exemple de processus de restauration appliqué à une image sismique.** (a) Modèle initial, (b) modèle triangulé.

Nous avons alors besoin de définir deux contraintes de restauration contrôlant le déplacement du bloc mobile. La première, appliquée au point d'intersection  $v_I^0$  de la faille et de l'horizon sur le bloc mobile, vise à introduire un déplacement tel que la continuité de l'horizon bleu soit retrouvée au niveau de la faille (vecteur bleu sur la figure 3.2.a). La seconde contrainte, appliquée à tous les points de la faille sur le bloc mobile, aura pour but et pour effet de maintenir le contact entre la partie mobile et la faille au cours de son déplacement (vecteurs rouges sur la figure 3.2.b).

Pour contrôler la déformation de l'image et assurer qu'elle soit "réaliste", nous devons également introduire des contraintes géométriques permettant de préserver l'aspect du maillage. Dans notre application, nous souhaitons en particulier préserver les aires et les longueurs de ce maillage. Cet objectif est atteint en définissant deux contraintes géométriques. La première, appliquée en tout point  $\{v_i\}$  du maillage, préserve la forme de chaque triangle. Puisque la taille d'un triangle peut varier tout en conservant sa forme, elle est complétée par la seconde contrainte qui préserve les longueurs de chaque arête. La combinaison de ces deux contraintes permet de conserver aires et longueurs, tel qu'escompté.

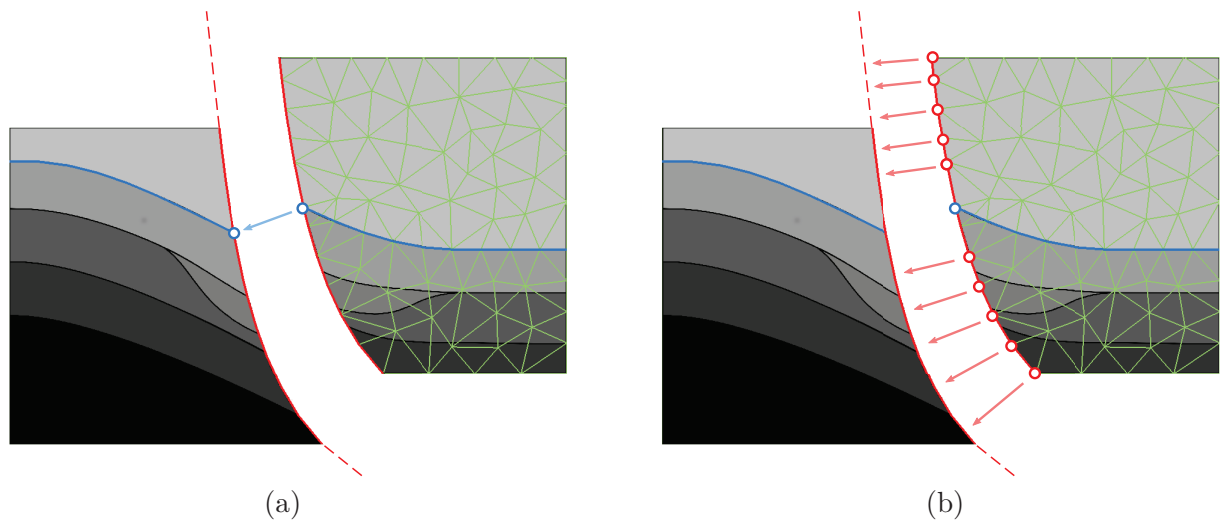


FIGURE 3.2 – **Illustration des contraintes de restauration que l'on impose pour la restauration de la faille sur la figure 3.1.** (a) Contrainte de continuité de l'horizon, (b) contrainte de maintien du contact le long de la faille.

L'écriture et la manière dont nous définissons ces différentes contraintes puis les appliquons aux ensembles de points concernées sera détaillée dans la section 3.2. Nous illustrons alors en figure 3.3.a le résultat obtenu à la fin du processus de restauration de la faille, produisant une déformation du maillage conforme aux différentes contraintes énumérées. Comme espéré et imposé, le contact à la faille de la partie mobile est conservé, la continuité de l'horizon bleu est à présent observable à l'intersection de la faille et l'aspect général du maillage après déformation à été conservé.

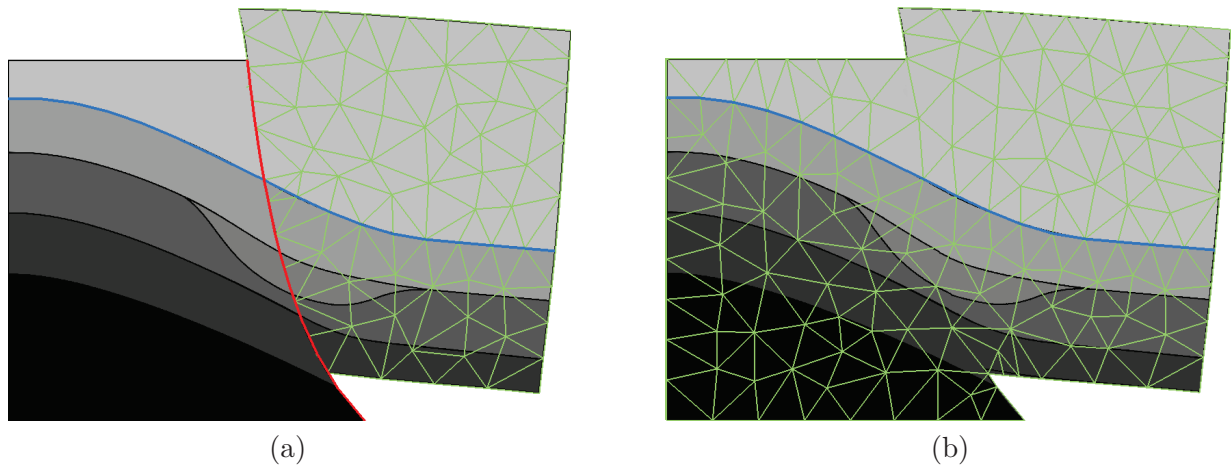


FIGURE 3.3 – **Restauration de la faille.** (a) Après restauration de la faille, (b) domaine remaillé qui sera utilisé dans le processus de mise à plat de l'horizon.

### 3.1.2 Mise à plat de l'horizon

Nous sommes maintenant parvenus à une étape où le glissement le long de la faille a été restauré et où l'horizon choisi est continu sur l'intégralité du domaine. Avant d'aborder la nouvelle étape de

déformation, nous allons abandonner le maillage courant qui ne couvrait qu'une partie de l'image. Nous créons donc un nouveau maillage par une triangulation contrainte par le contour de l'ensemble du domaine et par la géométrie de l'horizon. Il est à noter que la faille restaurée à l'étape précédente est désormais considérée comme passive et écartée du processus de restauration. Le nouveau maillage n'est donc pas contraint à sa géométrie. La figure 3.3.b illustre un exemple de situation obtenue à l'issue de cette opération.

La seconde étape dans le processus de restauration est la mise à plat de l'horizon. Le but est de fournir une représentation dépliée du système de dépôts (couches) sédimentaires. Pour ce faire, un horizon est choisi et une contrainte de restauration permettant de le mettre à plat en une position verticale est introduite, comme cela est illustré sur la figure 3.4. Après mise à plat, la longueur de l'horizon sera préservée au mieux par l'action des contraintes géométriques de préservation des formes et des longueurs d'arêtes qui a été discutée plus haut.

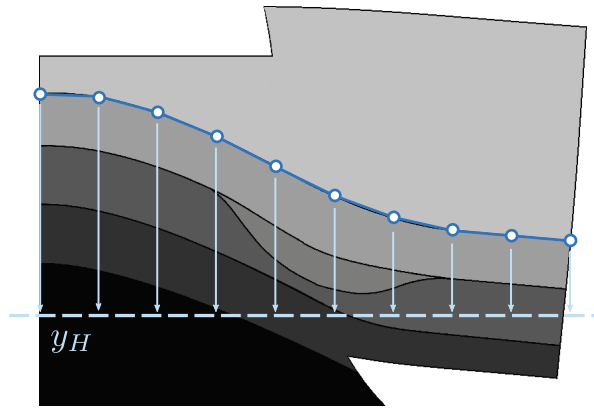


FIGURE 3.4 – Illustration de la contrainte de mise à plat de l'horizon.

La figure 3.5 présente l'image obtenue à l'issue de cette étape pour le cas du modèle de la figure 3.3.b. L'horizon obtenu est déplié comme attendu et le maillage conserve son aspect général. Le processus complet permet de mettre en lumière la restauration de l'évènement géologique coupé par la faille sur la figure 3.1.a. Son identification visuelle devient donc plus aisée.

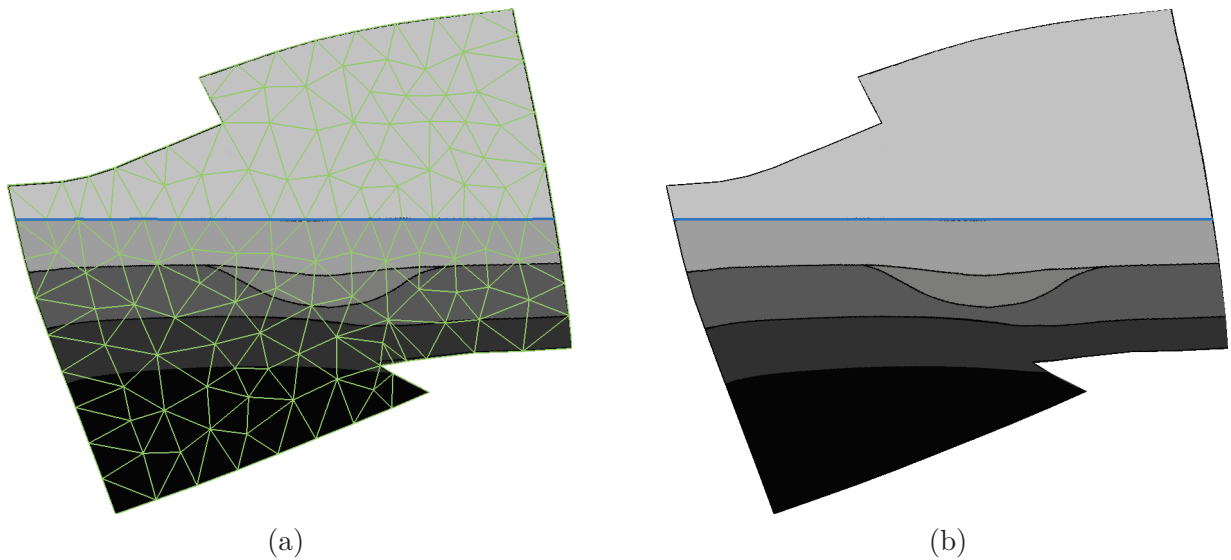


FIGURE 3.5 – Modèle restauré après remise à plat de l’horizon. (a) Triangulé, (b) non-triangulé.

## 3.2 Écriture des contraintes de déformation

Ayant décrit le processus général de restauration (en section 3.1) et introduit les différentes contraintes intervenant pour restaurer une faille puis mettre à plat un horizon, tout en respectant les aires et formes du maillage, nous allons à présent formaliser ces contraintes et les traduire par des équations.

Dans la suite nous noterons  $\mathcal{V} = \{v_i\}$  l’ensemble des points de  $\mathcal{M}$  et  $\mathcal{E}$  l’ensemble de ses arêtes (figure 3.6).  $\mathcal{V}$  contiendra  $n$  points, nous définissons donc :  $n = \text{card}(\mathcal{V})$ .

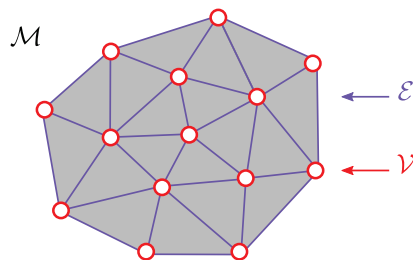


FIGURE 3.6 – Notations utilisées pour notre maillage  $\mathcal{M}$ , constitué d’un ensemble  $\mathcal{V}$  de  $n$  points reliés par un ensemble  $\mathcal{E}$  d’arêtes.

### 3.2.1 Introduction des contraintes de restauration

Une contrainte de restauration s’applique à un sous-ensemble de points de contrainte  $\mathcal{V}_c = \{c_i\}$  tel que  $\mathcal{V}_c \subset \mathcal{V}$ , pour lesquels on peut déterminer une position restaurée. Considérant une fonction  $c$  qui, appliquée à un point  $v_i \in \mathcal{V}_c$ , produit la position restaurée, on exprime l’erreur vectorielle  $\delta\mathbf{C}_i$

liée à la contrainte de restauration au point  $v_i$  par :

$$\delta \mathbf{C}_i = \mathbf{v}_i - \mathbf{c}(\mathbf{v}_i). \quad (3.1)$$

Pour respecter l'ensemble des contraintes de restauration, on souhaite donc minimiser l'ensemble des erreurs, que l'on exprime par la somme quadratique :

$$J_C = \sum_{v_i \in \mathcal{V}_c} \|\delta \mathbf{C}_i\|^2. \quad (3.2)$$

Nous pouvons réécrire ce terme sous forme matricielle :

$$J_C = \|CV - c(V)\|^2, \quad (3.3)$$

où la matrice  $V$  (de taille  $n \times 2$ ) contient l'ensemble des positions de  $\mathcal{V}$ ,  $C$  est la matrice (de taille  $\text{card}(\mathcal{V}_c) \times n$ ), qui permet d'extraire l'ensemble des  $\{c_i\}$  de  $\mathcal{V}$  : le terme  $CV$  produit en effet une matrice (de taille  $\text{card}(\mathcal{V}_c) \times 2$ ) qui contient seulement les positions des  $\{c_i\}$ , enfin, la matrice  $c(V)$  (de taille  $\text{card}(\mathcal{V}_c) \times 2$ ) contient les valeurs de  $\mathbf{c}(\mathbf{v}_i)$  pour l'ensemble des  $v_i \in \mathcal{V}_c$ . On définit également :

$$C_{r,s} = \begin{cases} 1 & \text{lorsque } v_c \text{ est le } r^{\text{ème}} \text{ élément de } \mathcal{V}_c \text{ et le } s^{\text{ème}} \text{ élément de } \mathcal{V}, \\ 0 & \text{sinon.} \end{cases} \quad (3.4)$$

$$V_i = [v_{ix}, v_{iy}], \forall i \in [1, \dots, \text{card}(\mathcal{V})], \quad (3.5)$$

$$c(V)_i = [c(v_i)_x, c(v_i)_y], \forall i \in [1, \dots, \text{card}(\mathcal{V}_c)]. \quad (3.6)$$

Dans un but d'illustration, la figure 3.7 montre un exemple dans lequel une contrainte de restauration s'applique aux points  $v_1$ ,  $v_4$  et  $v_5$  de  $\mathcal{V}$  ainsi que le contenu de l'équation (3.3) qui est obtenu dans ce cas.

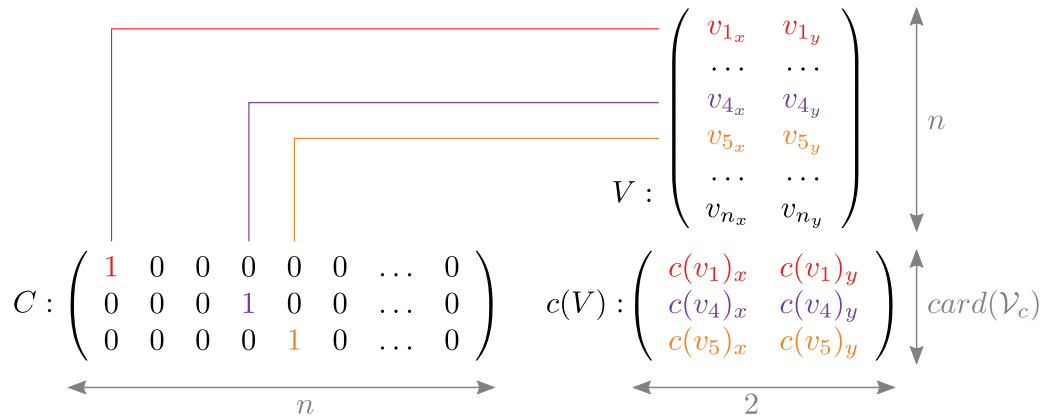


FIGURE 3.7 – Contenu des matrices  $C$ ,  $V$  et  $c(V)$  intervenant dans l'équation 3.3 dans le cas d'une contrainte de restauration s'appliquant à trois points d'un maillage.

Nous allons à présent détailler l'approche que nous utiliserons pour adapter  $J_C$  dans notre contexte de restauration d'image sismique de façon à introduire deux contraintes appliquées à deux sous-ensembles de points distincts régissant la restauration d'une faille ainsi que la mise à plat d'un horizon.

### 3.2.2 Scénario de restauration de la faille

Comme introduit dans la section 3.1, le scénario de restauration de la faille fait intervenir deux contraintes qui sont la contrainte de continuité de l'horizon et la contrainte de contact à la faille. Nous détaillons à présent l'écriture de ces contraintes.

#### 3.2.2.1 Contrainte de continuité de l'horizon

Cette contrainte vise à déplacer la partie mobile de façon à ce que le point  $v_I^0$  soit le plus proche possible du point  $v_I$  sur le contour de la partie statique, tel que l'illustre la figure 3.8 :

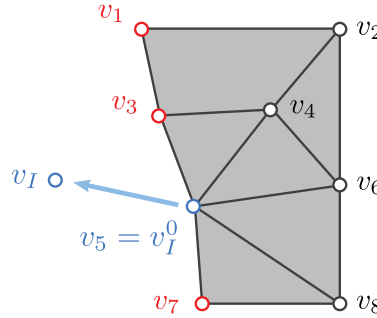


FIGURE 3.8 – Exemple de la contrainte de continuité de l'horizon.

En nous basant sur les équations (3.2) et (3.3), nous exprimons le terme à minimiser :

$$\begin{aligned} J_{F_1} &= \|\mathbf{v}_I^0 - \mathbf{v}_I\|^2 \\ &= \|C_I V - \mathbf{v}_I\|^2 \end{aligned} \quad (3.7)$$

où  $C_I$  est la matrice (de taille  $1 \times n$ ) qui choisit  $v_I^0$  parmi les points de  $\mathcal{V}$  comme présenté dans l'équation (3.3) et  $V$  la matrice des positions définie par l'équation (3.5). On définit :

$$C_{I,s} = \begin{cases} 1 & \text{lorsque } v_I^0 \text{ est le } s^{\text{ème}} \text{ élément de } \mathcal{V}, \\ 0 & \text{sinon.} \end{cases} \quad (3.8)$$

La figure 3.8 montre un exemple qui illustre l'écriture de cette contrainte de continuité de l'horizon. Pour ce jeu de données d'exemple, la matrice  $C_I$  et le vecteur  $\mathbf{v}_I$  contiennent :

$$C_I : \left( 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \right), \quad \mathbf{v}_I : [ v_{Ix} \ v_{Iy} ].$$



### 3.2.2.2 Contrainte de contact à la faille

La seconde contrainte doit garantir que la partie mobile restera en contact avec la faille durant sa déformation. Dans ce but, soit  $\mathcal{V}_f = \{v_F\}$  le sous-ensemble de  $\mathcal{V}$  contenant les points de la partie mobile appartenant à la faille, nous définissons  $f(v_F)$  leur projection sur la polyligne de faille. Nous honorons alors la contrainte de contact à la faille, en imposant le déplacement des points  $\{v_F\}$  aux positions  $\{f(v_F)\}$ , tel qu'illustré sur la figure 3.9. Toutefois, avant de détailler l'écriture de cette

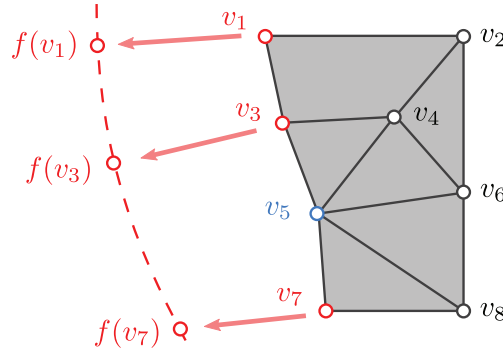


FIGURE 3.9 – Exemple de la contrainte de contact à la faille.

contrainte, nous devons définir la fonction de projection  $f(v_F)$ .

#### a. Choix d'une méthode de projection

Plusieurs stratégies de projection d'un point  $v_{F_k}$  à la faille peuvent être envisagées, parmi lesquelles la projection horizontale (voir figure 3.10.a), la projection selon le vecteur orthogonal à la droite de régression de la faille (voir figure 3.10.b), la projection suivant le vecteur normal de  $v_{F_k}$  (voir figure 3.10.c), et la projection au point le plus proche de la faille (voir figure 3.10.d). Nous proposons de détailler ces différentes méthodes et justifions celle retenue pour la restauration d'images sismiques.

La projection horizontale possède l'avantage d'être simple, robuste, et de conserver l'ordre des points  $f(v_{F_k})$  projetés par rapport aux positions initiales  $v_{F_k}$ . Elle est envisageable si l'on peut garantir que pour toute droite d'équation  $y = a$  où  $a$  est une constante, il n'existe au maximum qu'une intersection unique entre  $y$  et la faille. Les géosciences postulent que les failles respectent cette propriété. La méthode de projection horizontale peut toutefois produire des résultats éloignés de la position intuitivement voulue, ainsi que nous l'illustrons sur la figure 3.11.a. Ce scénario peut en pratique se produire avec des failles présentant une extension horizontale, mais nous considérons que l'application de nos travaux n'est pas concernée par ce cas particulier.

Si les failles sont inclinées, on pourrait considérer que projeter un point  $v_{F_k}$  sur la droite de régression  $R$  approximant les points de la faille est plus réaliste. On détermine alors la position  $f(v_{F_k})$  à l'intersection de  $R$  et de la droite orthogonale à  $R$  passant par  $v_{F_k}$ . Tout comme la projection horizontale, cette méthode est robuste car elle garantit l'ordre des points  $\{f(v_{F_k})\}$  projetés par rapport aux positions initiales.

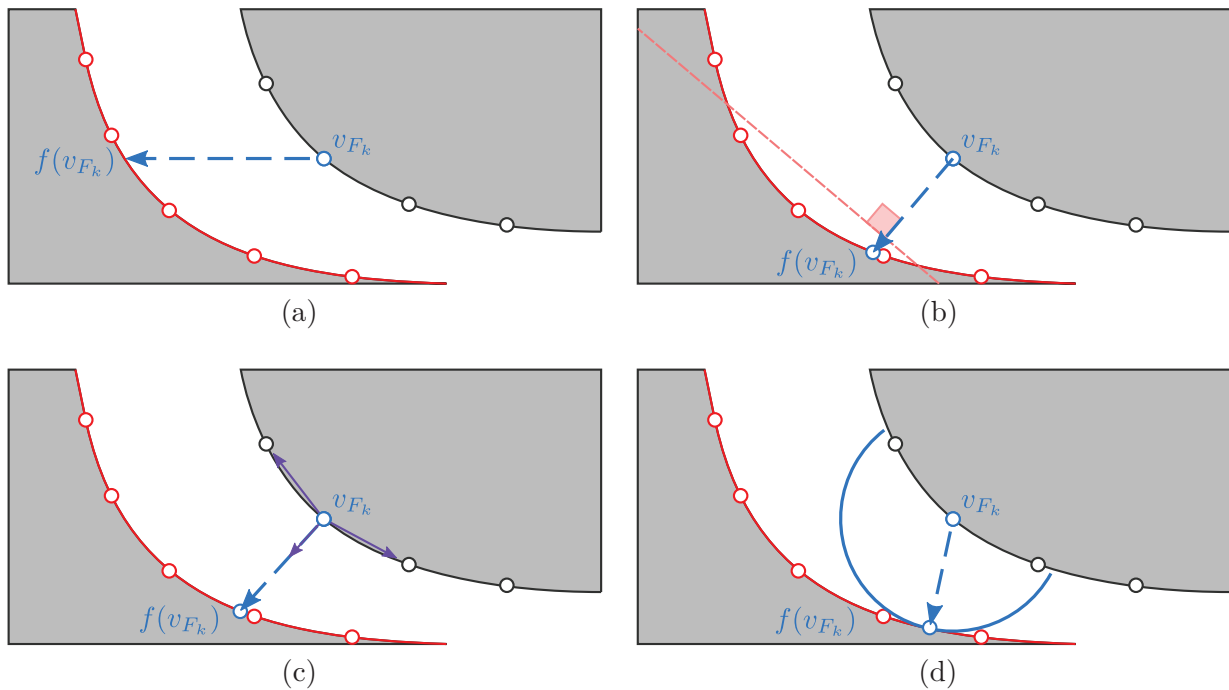


FIGURE 3.10 – **Différentes stratégies de projection d'un point  $v_{F_k}$  sur la faille** : (a) Projection horizontale, (b) projection selon le vecteur orthogonal à la droite de régression, (c) projection suivant le vecteur normal, (d) projection au point le plus proche de la faille.

La projection suivant le vecteur normal de  $v_{F_k}$  produit habituellement des résultats intuitivement plus justes qu'une projection horizontale mais nécessite de devoir déterminer ce vecteur normal pour tout point de  $v_F$ . Néanmoins, comme nous le soulignons sur l'image 3.11.b, l'ordre des points projetés  $\{f(v_{F_k})\}$  peut différer de l'ordre des points  $\{v_{F_k}\}$  et causer des instabilités (artefacts) dans notre algorithme de restauration. En pratique, bien que les failles aient une géométrie lisse, une certaine rugosité peut être observable sur des pointés classiques, en particulier avec l'emploi de triangulations denses.

Dans la dernière méthode de projection, il existe plusieurs manières de déterminer la distance la plus courte à la faille. En particulier, on peut déterminer géométriquement le cercle de centre  $v_{F_k}$  qui n'a qu'un point d'intersection  $f(v_{F_k})$  avec la faille ou connaître (en utilisant une RBF par exemple [TO99]) une fonction distance signée à la faille  $d(x, y)$  et chercher la position la plus proche de son passage par zéro au moyen d'une réduction de gradient (méthode de descente classique). Cette approche permet d'obtenir des positions en accord avec celles intuitivement espérées, mais à l'instar de la méthode de projection selon le vecteur normal, elle ne garantit pas le respect de l'ordre des points projetés  $\{f(v_{F_k})\}$  par rapport à l'ordre des points  $\{v_{F_k}\}$ . Aussi, certaines instabilités ont été remarquées lors de l'usage de maillages denses (voir 3.11.c). Enfin, cette méthode peut s'avérer coûteuse en temps de calcul.

En pratique, si la distance  $\|v_{F_k} - f(v_{F_k})\|$  est très faible, les résultats produits par ces différentes méthodes sont très proches. Or c'est précisément le cas dans notre application de restauration d'image

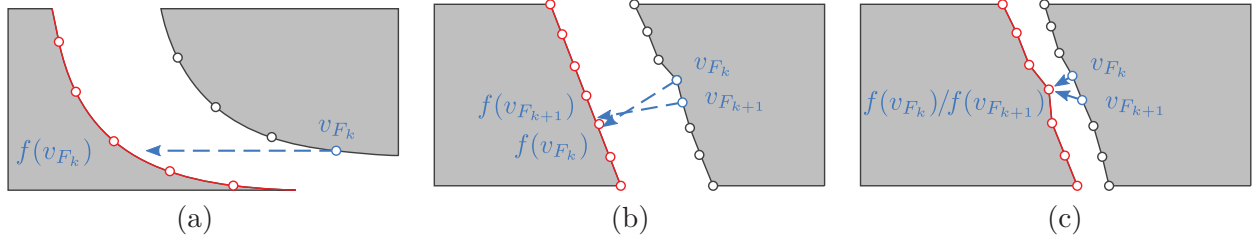


FIGURE 3.11 – **Inconvénients des différentes méthodes de projection** : (a) Projection horizontale dans le cas d’une faille ayant une extension horizontale, la position déterminée peut être éloignée de celle intuitivement escomptée. (b) Projection suivant le vecteur normal : l’ordre des points peut s’inverser après projection. (c) Projection à la distance la plus proche : plusieurs points peuvent être projetés au même endroit.

sismique, à l’exception des premières itérations comme cela est détaillé dans les sections 3.5 et 3.6. Pour cette raison, une solution robuste et peu coûteuse nous semble la plus adaptée et nous retenons la méthode de projection horizontale.

## b. Écriture de la contrainte de contact à la faille

Ayant défini une méthode de projection, on exprime pour un point  $v_i$  de  $\mathcal{V}_f$  l’erreur vectorielle  $\delta\mathbf{F}_i$  à sa position escomptée  $f(v_i)$  par :

$$\delta\mathbf{F}_i = \mathbf{v}_i - \mathbf{f}(\mathbf{v}_i). \quad (3.9)$$

La contrainte de contact à la faille s’exprime alors en minimisant la somme quadratique  $J_{F_2}$  des erreurs vectorielles  $\delta\mathbf{F}_i$  :

$$\begin{aligned} J_{F_2} &= \sum_{v_i \in \mathcal{V}_f} \|\mathbf{v}_i - \mathbf{f}(\mathbf{v}_i)\|^2 \\ &= \|C_F V - c_F(V)\|^2 \end{aligned} \quad (3.10)$$

où  $C_F$  est la matrice (de taille  $\text{card}(\mathcal{V}_f) \times n$ ) permettant d’extraire l’ensemble des  $\{v_F\}$  de  $\mathcal{V}$  tel que présenté en équation (3.2),  $V$  la matrice (de taille  $n \times 2$ ) des positions définie dans l’équation (3.5), et  $c_F(V)$  la matrice (de taille  $\text{card}(\mathcal{V}_f) \times 2$ ) contenant les valeurs de  $\mathbf{f}(\mathbf{v}_i)$  pour l’ensemble des  $v_i \in \mathcal{V}_f$ . On rappelle que  $n = \text{card}(\mathcal{V})$  et définit :

$$C_{F_{r,s}} = \begin{cases} 1 \text{ lorsque } v_F \text{ est le } r^{\text{ème}} \text{ élément de } \mathcal{V}_f \text{ et le } s^{\text{ème}} \text{ élément de } \mathcal{V}, \\ 0 \text{ sinon.} \end{cases} \quad (3.11)$$

$$c_F(V)_i = [f(v_i)_x, f(v_i)_y], \forall v_i \in \mathcal{V}_f. \quad (3.12)$$

Si nous reprenons l’exemple illustré en 3.9, lors de l’écriture de la contrainte de contact à la faille, les matrices  $C_F$  et  $c_F(V)$  contiennent :

$$C_F : \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, c_F(V) : \begin{pmatrix} f(v_1)_x & f(v_1)_y \\ f(v_3)_x & f(v_3)_y \\ f(v_7)_x & f(v_7)_y \end{pmatrix}.$$

Précisons que le point  $v_I$ , à l'intersection de la faille et de l'horizon est exclu de  $\mathcal{V}_f$  pour ce cas précis. Nous rappelons en effet que la contrainte de continuité de l'horizon y impose un déplacement spécifique à une position sur la faille. L'ajout d'une contrainte de contact à la faille supplémentaire serait alors redondante.

### 3.2.3 Scénario de mise à plat de l'horizon

Comme introduit dans la section 3.1, ce scénario ne fait intervenir qu'une contrainte unique de mise à plat de l'horizon. Nous détaillons à présent l'écriture de cette contrainte.

Soit  $\mathcal{V}_h$  l'ensemble des points de  $\mathcal{V}$  appartenant à l'horizon choisi : la contrainte de mise à plat de l'horizon consiste à déplacer chaque point  $\{v_i : (x_i, y_i)\}$  de  $\mathcal{V}_h$  sur la droite horizontale d'équation  $y = y_H$ , tel qu'illustré en figure 3.12.

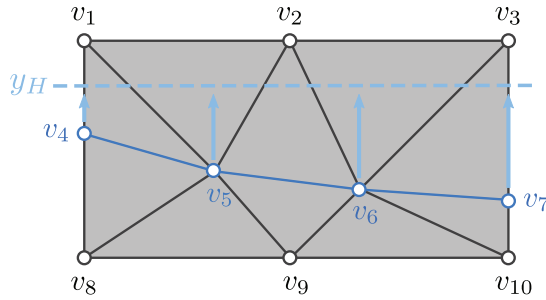


FIGURE 3.12 – Exemple de la contrainte de mise à plat de l'horizon.

On exprime alors pour tout point  $v_{H_i}$  l'erreur verticale  $\delta H_i$  à cette droite par :

$$\delta H_i = y_H - y_i. \quad (3.13)$$

La contrainte de mise à plat de l'horizon consiste alors à minimiser la somme quadratique  $J_H$  des erreurs verticales :

$$J_H = \sum_{v_i \in \mathcal{V}_h} (y_H - y_i)^2, \quad (3.14)$$

que l'on peut réécrire sous la forme matricielle :

$$J_H = \|C_H V - c_H(V)\|^2, \quad (3.15)$$

où  $C_H$  est la matrice (de taille  $\text{card}(\mathcal{V}_h) \times n$ ) qui permet d'extraire l'ensemble des  $\{v_H\}$  de  $\mathcal{V}$  comme présenté en équation (3.2),  $V$  la matrice des positions définie dans l'équation (3.5), et  $c_H(V)$  la matrice (de taille  $\text{card}(\mathcal{V}_h) \times 2$ ) contenant les positions mises à plat souhaitées. On définit :

$$C_{Hr,s} = \begin{cases} 1 & \text{lorsque } v_H \text{ est le } r^{\text{ème}} \text{ élément de } \mathcal{V}_h \text{ et le } s^{\text{ème}} \text{ élément de } \mathcal{V}, \\ 0 & \text{sinon.} \end{cases} \quad (3.16)$$

$$c_H(V)_i = [x_i, y_H], \forall v_i \in \mathcal{V}_h. \quad (3.17)$$

La figure 3.12 montre un exemple illustrant l'écriture de la contrainte de mise à plat de l'horizon. Pour ce jeu de données d'exemple, les matrices  $C_H$  et  $c_H(V)$  contiennent :

$$C_H : \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, c_H(V) : \begin{pmatrix} v_{4x} & y_H \\ v_{5x} & y_H \\ v_{6x} & y_H \\ v_{7x} & y_H \end{pmatrix}.$$

### 3.2.4 Introduction des contraintes géométriques

Après avoir décrit la manière que nous utilisons pour introduire une déformation en imposant des contraintes à des sous-ensembles spécifiques de points, nous allons à présent introduire le mécanisme que nous mettrons en œuvre pour propager une déformation rigide à l'ensemble du maillage. Il est en effet très important que cette déformation rigide soit distribuée au mieux afin d'éviter des distorsions locales fortes et non réalistes, qui constitueraient donc un artefact. Ce rôle est dévolu aux contraintes géométriques, qui visent à préserver au mieux la forme ainsi que l'aire de chaque triangle.

Pour atteindre cet objectif de conservation des formes et des aires des triangles, l'approche classique consiste à essayer de préserver le *Laplacien* du maillage ainsi que les longueurs d'arêtes des triangles. Le critère préservant le *Laplacien* du maillage conserve la forme de  $\mathcal{M}$  en préservant les angles et les rapports entre les longueurs des arêtes (voir figure 3.13.b, les angles étant représentés en vert et les arêtes en bleu). Le second critère conserve les longueurs d'arêtes et permet d'assurer la prise en compte de la sensibilité aux homothéties de  $\mathcal{M}$  (voir figure 3.13.c, les longueurs préservées étant représentées en violet). Au final, en préservant les formes et les longueurs, on préserve également les aires. Dans les sections qui suivent nous allons décrire plus en détail chacune de ces deux contraintes.

#### 3.2.4.1 Préservation des formes

Afin de préserver la forme du maillage, un critère classique consiste à utiliser le *Laplacien* (voir par exemple [Sor06]). Localement, en chaque point  $v_i$  du maillage, on exprime par la mesure vectorielle du *Laplacien*  $\mathcal{L}_i(v_i)$ , la position de ce point relativement à ses points adjacents. Pour cela, on définit un ensemble  $\mathcal{N}(v_i)$  qui regroupe tous les points  $v_j \in \mathcal{M}$  adjacents à  $v_i$  (voir figure 3.14) :

$$\forall v_j \in \mathcal{M}, \mathcal{N}(v_i) = \{v_j, j \in \mathbb{N} \text{ tel que } v_j \text{ adjacent à } v_i\}. \quad (3.18)$$

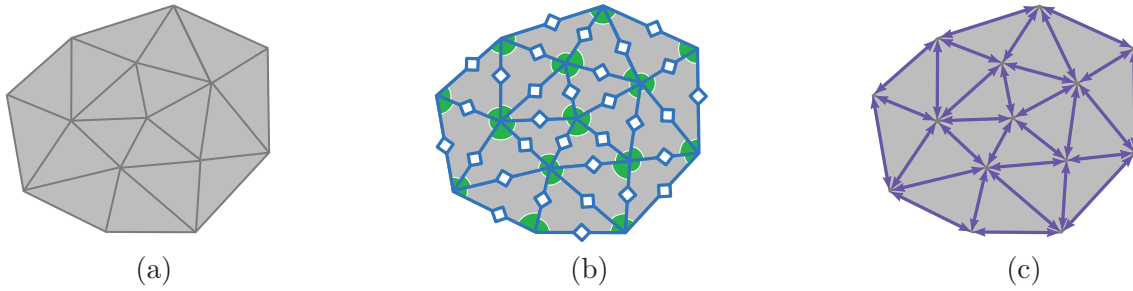


FIGURE 3.13 – **Action des contraintes géométriques sur un maillage.** (a) Maillage initial  $\mathcal{M}$ , (b) utilisation du *Laplacien* pour préserver les angles et les rapports des arêtes dans  $\mathcal{M}$ , (c) préservation des longueurs d'arête.

Le *Laplacien* s'exprime au point  $v_i$  par :

$$\mathcal{L}_i(\mathbf{v}_i) = \mathbf{v}_i - \sum_{j \in \mathcal{N}(v_i)} \omega_j \cdot \mathbf{v}_j, \quad (3.19)$$

où  $\omega_j$  est un ensemble de coefficients de pondération associés aux points  $\{v_j\}$ .

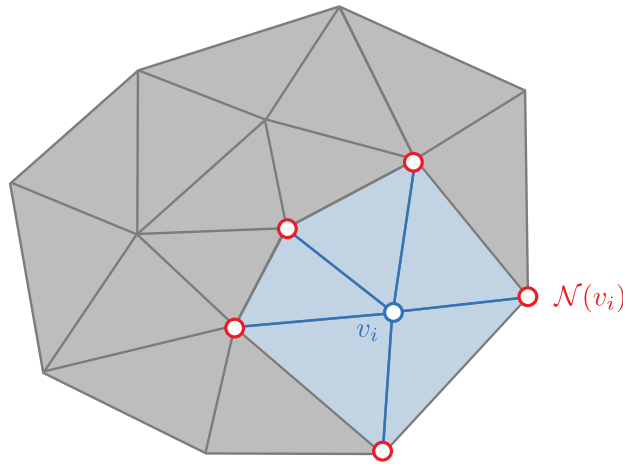


FIGURE 3.14 – **Définition de l'ensemble  $\mathcal{N}(v_i)$  des points adjacents à  $v_i$  intervenant dans le calcul du *Laplacien*  $\mathcal{L}_i(v_i)$ .**

Selon les maillages considérés, la littérature propose plusieurs manières classiques possibles de choisir ces coefficients de pondération  $\omega_j$ . Si la longueur des arêtes du maillage est partout égale, c'est-à-dire si le maillage est parfaitement régulier, il est classique [Sor06] de choisir comme coefficients

$$\omega_j = \frac{1}{d}, \quad (3.20)$$

où  $d$  est le nombre de points adjacents  $v_j$ , c'est-à-dire une simple pondération par le nombre de voisins. Une telle définition de *Laplacien* est alors connue sous le nom d'*umbrella operator*.

En revanche cette formulation n'est pas applicable aux maillages irréguliers car elle n'est pas

barycentrique. En effet, notre approche consiste à déterminer pour chaque point  $v_i$  un *Laplacien* (et ses coefficients  $\omega_j$ ) qui définit la position de  $v_i$  par rapport à ces voisins  $\{v_j\}$ . En d'autres termes, on souhaite exprimer l'équilibre des distances de  $v_i$  à ses points adjacents. Si ces derniers se déplacent (c'est-à-dire si le maillage se déforme), on espère alors, grâce au *Laplacien*, pouvoir estimer une nouvelle position de  $v_i$  qui conserve au mieux l'équilibre initial. Typiquement, cela consiste à considérer que  $\mathcal{L}_i(\mathbf{v}_i)$  est le barycentre des points  $\{v_j\}$  pondéré des coefficients  $\{\omega_j\}$ . En conséquence, une formulation de *Laplacien* conforme à la définition du barycentre respecte :

$$\mathcal{L}_i(\mathbf{v}_i) = 0. \quad (3.21)$$

Enfin, comme on souhaite exprimer l'équilibre des distances de  $v_i$  à ses points adjacents, les pondérations  $\{\omega_j\}$  doivent être positives.

Pour les maillages irréguliers, un choix de coefficients classique possible pour tenir compte de la longueur des arêtes dans la formule de pondération et respecter la définition barycentrique de l'équation (3.21) va consister à calculer des coefficients dits "cotangeants" [MDS<sup>+</sup>02], qui sont définis en un point  $v_i$  par :

$$\omega_j = \frac{\lambda_j}{\sum_{k \in \mathcal{N}(v_i)} \lambda_k}, \quad \lambda_j = \cot(\alpha) + \cot(\beta), \quad (3.22)$$

où  $\alpha$  est l'angle  $\widehat{v_i v_{j-1} v_j}$  et  $\beta$  l'angle  $\widehat{v_j v_{j+1} v_i}$ ,  $\forall j \in \mathcal{N}(v_i)$ , comme cela est illustré sur la figure 3.15.

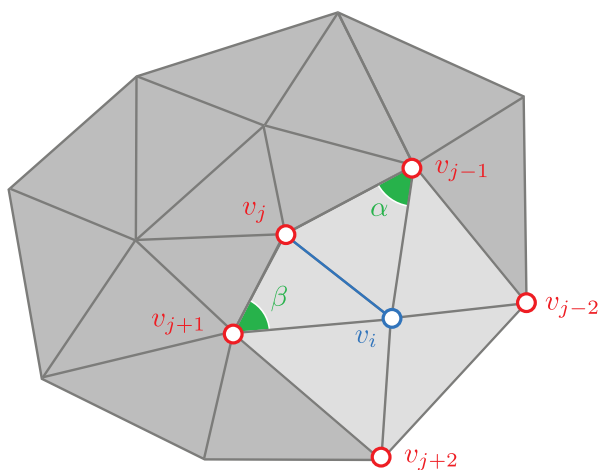


FIGURE 3.15 – Définition du *Laplacien* pondéré par des coefficients dits "cotangeant" au point  $v_i$  d'un maillage non régulier.

Notre application requiert toutefois que ces coefficients soient toujours positifs et les coefficients cotangeants n'assurent  $w_j \geq 0$ , qu'à condition de respecter pour tout couple de triangles adjacents la condition  $\alpha + \beta < \pi$  (voir [BS07]), où  $\alpha$  et  $\beta$  sont les angles opposés à l'arête d'adjacence, comme on le voit sur la figure 3.15. Pour se convaincre que :  $\omega_i \geq 0$  si  $\alpha + \beta < \pi$ , nous proposons la reformulation



suivante et rappelons que le *sinus* d'un angle inférieur à  $\pi$  est par définition positif :

$$\omega_j = \cot(\alpha) + \cot(\beta) = \frac{\cos(\alpha)}{\sin(\alpha)} + \frac{\cos(\beta)}{\sin(\beta)} = \frac{\sin(\alpha + \beta)}{\sin(\alpha) \cdot \sin(\beta)}. \quad (3.23)$$

Le terme reformulé ne sera positif que si son numérateur l'est également, c'est-à-dire si  $\alpha + \beta < \pi$ .

Lorsqu'on ne peut assurer la condition  $\alpha + \beta < \pi$ , ce qui en pratique est généralement le cas avec les triangulations qui ne sont pas dites de *Delaunay*, un choix classique de coefficients est celui dit des coordonnées moyennées (*mean value coordinates*) introduits par [Flo03], qui s'écrit :

$$\omega_j = \frac{\lambda_j}{\sum_{k \in \mathcal{N}(v_i)} \lambda_k}, \quad \lambda_j = \frac{\tan(\alpha_{j-1}/2) + \tan(\alpha_j/2)}{|v_i - v_j|}, \quad (3.24)$$

où  $\alpha_j$  est l'angle  $\widehat{v_{j+1}v_jv_i}$ ,  $\forall j \in \mathcal{N}(v_i)$ , comme cela est illustré sur la figure 3.16. Ces coefficients qui sont une généralisation des coefficients barycentriques, garantissent par propriété :  $\omega_j \geq 0$ .

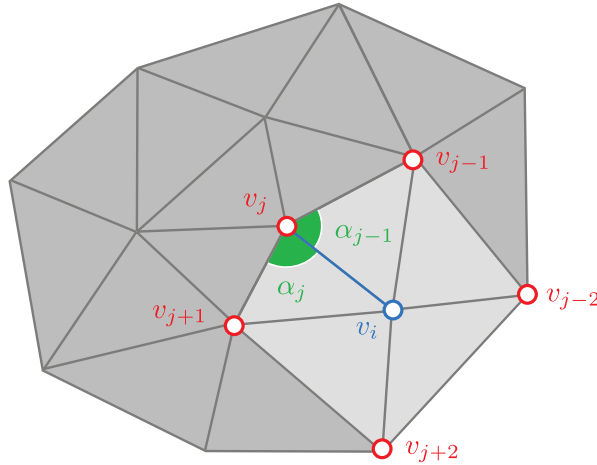


FIGURE 3.16 – Définition des des coordonnées moyennées (*mean value coordinates*) au point  $v_i$  d'un maillage en étoile centré en  $v_i$ .

Les *mean value coordinates* ont toutefois une limitation importante (en pratique) de ne pouvoir être définies aux points formant le contour (bord externe) du maillage car la formule qui est utilisée nécessite d'avoir des points de maillage tout autour du point de calcul et pas seulement d'un seul côté. Cette remarque concerne également les coordonnées cotangents, qui se définissent grâce à deux triangles adjacents, ce qui est à-priori impossible pour ces mêmes points formant le contour, où le triangle adjacent est absent. Nous verrons toutefois à la fin de cette section qu'il existe une méthode applicable aux coordonnées cotangents pour s'affranchir de cette limitation. À notre connaissance toutefois, aucune méthode similaire n'existe pour traiter la limitation des *mean value coordinates*.

Dans le cas de notre travail sur la restauration d'images sismiques, nous utilisons toujours une triangulation conforme de *Delaunay*, dont une des propriétés garantit que tout triangle  $ABD$  et  $BCD$  (voir figure 3.17.b) respecte  $\widehat{ABD} + \widehat{BCD} < \pi$ . En conséquence nous pouvons toujours garantir que

les angles  $\alpha, \beta$  utilisés dans la formule de définition des coefficients cotangents (3.22) vérifient la condition  $\forall \alpha, \beta : \alpha + \beta < \pi$ . De ce fait, et pour éviter de rencontrer la limitation importante du critère des *mean value coordinates* qui est qu'il n'est pas possible de définir les coefficients  $\omega_j$  sur le contour du maillage, nous choisissons donc d'utiliser les coefficients cotangents.

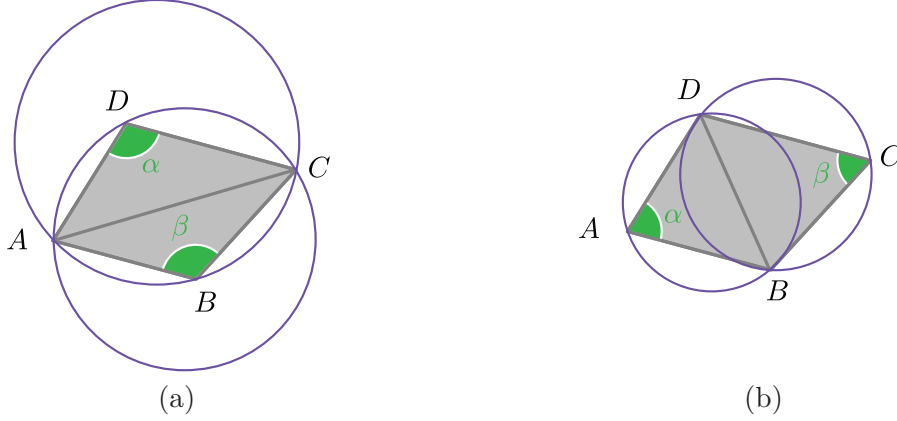


FIGURE 3.17 – **Construction d'une triangulation de Delaunay.** (a) La propriété de *Delaunay* n'est pas respectée, car les cercles inscrits aux triangles contiennent d'autres points que les sommets des triangles. Dans ce cas,  $\alpha + \beta > \pi$  et ce maillage n'est pas conforme de *Delaunay*. (b) On supprime l'arête  $AC$ , et on crée l'arête  $BD$ . Les cercles inscrits aux triangles ne contiennent désormais plus que les sommets. La triangulation à la base de ce maillage est alors une triangulation de *Delaunay* et garantit par propriété que  $\alpha + \beta < \pi$ .

Ayant maintenant défini une mesure de forme (le *Laplacien*) en chaque point d'un maillage, nous cherchons à présent à mesurer la variation de forme totale d'un maillage initial  $\mathcal{M}^0$  que l'on déforme pour obtenir un maillage  $\mathcal{M}$ . Dans un premier temps, nous nous intéressons donc à définir la variation de forme locale  $\delta\mathcal{L}_i$ , qui mesure l'impact de la déformation de  $\mathcal{M}^0$  en  $\mathcal{M}$ , sur le positionnement d'un point  $v_i^0 \in \mathcal{M}^0$  par rapport à ses voisins, que l'on transforme en un point  $v_i \in \mathcal{M}$ .

Pour déterminer la variation de forme locale  $\delta\mathcal{L}_i$ , on commence par définir le *Laplacien* de référence  $\mathcal{L}_i^0$  en chaque point  $v_i^0$  de  $\mathcal{M}^0$ , c'est-à-dire par calculer :

$$\mathcal{L}_i^0(\mathbf{v}_i^0) = \mathbf{v}_i^0 - \sum_{j \in \mathcal{N}(v_i)} \omega_j^0 \cdot \mathbf{v}_j^0, \quad (3.25)$$

où  $\omega_j^0$  sont les coefficients déterminés par l'équation (3.22).

Cette mesure de référence  $\mathcal{L}_i^0(\mathbf{v}_i^0)$  doit à présent être comparée à une mesure de forme des points  $\{v_i\}$  après déformation. Nous choisissons  $\mathcal{L}_i^0(\mathbf{v}_i)$ . Cette mesure est comparable à  $\mathcal{L}_i^0(\mathbf{v}_i^0)$  puisque l'on exprime les points déplacés comme une combinaison pondérée par les coefficients initiaux  $\omega_j^0$ .

Finalement, nous cherchons à exprimer une mesure de variation de forme locale  $\delta\mathcal{L}_i$ , exprimée en fonction de  $\mathcal{L}_i^0(\mathbf{v}_i^0)$  et  $\mathcal{L}_i^0(\mathbf{v}_i)$ , qui soit exactement nulle si les maillages  $\mathcal{M}^0$  et  $\mathcal{M}$  sont similaires. Dans le contexte de notre application nous entendons par "similaires" deux maillages qui sont identiques à

une translation, rotation et/ou homothétie près. On doit ainsi respecter :

$$T\mathcal{L}_i^0(\mathbf{v}_i^0) = \mathcal{L}_i^0(\mathbf{v}_i), \text{ si } \mathcal{M}^0 \text{ similaire à } \mathcal{M}. \quad (3.26)$$

où  $T$  est une matrice de transformation décrivant les éventuelles translations, homothéties et/ou rotations pouvant distinguer deux maillages similaires  $\mathcal{M}$  et  $\mathcal{M}_0$ . Afin de mesurer l'impact des transformations que nous avons énumérées, nous allons comparer les mesures  $\mathcal{L}_i^0(\mathbf{v}_i^0)$  et  $\mathcal{L}_i^0(\mathbf{v}_i)$  pour chacun des cas de la translation, de la rotation et de l'homothétie.

Le *Laplacien* est une grandeur vectorielle invariante à la translation (car basée sur des dérivées spatiales) mais impactée par les rotations et les homothéties, comme cela est illustré visuellement sur les figures 3.18, 3.19 et 3.20, où l'on compare les grandeurs  $\mathcal{L}_i^0(\mathbf{v}_i^0)$  et  $\mathcal{L}_i^0(\mathbf{v}_i)$  en fixant  $\omega_j^0 = 1$ . Pour la simple translation présentée sur la figure 3.18, ces *Laplaciens* sont identiques comme attendu, mais pour la rotation de la figure 3.19 et l'homothétie de la figure 3.20 ces *Laplaciens* sont différents. On en déduit que la matrice  $T$  ne doit décrire que la rotation et/ou homothétie distinguant  $\mathcal{M}$  de  $\mathcal{M}^0$  et que la description de la translation s'avère superflue.

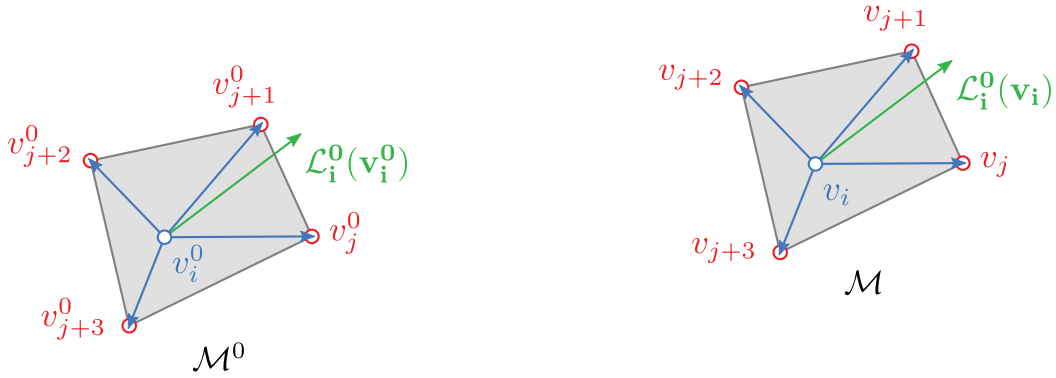


FIGURE 3.18 – Comparaison des *Laplaciens*  $\mathcal{L}_i^0(\mathbf{v}_i^0)$  et  $\mathcal{L}_i^0(\mathbf{v}_i)$  dans le cas d'une translation, montrant que le *Laplacien* est insensible aux translations.

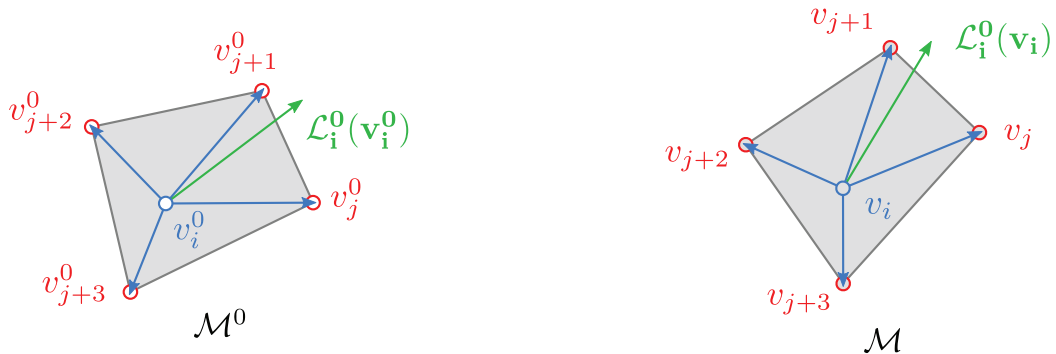


FIGURE 3.19 – Comparaison des *Laplaciens*  $\mathcal{L}_i^0(\mathbf{v}_i^0)$  et  $\mathcal{L}_i^0(\mathbf{v}_i)$  dans le cas d'une rotation, montrant que le *Laplacien* est sensible aux rotations.

Cependant, l'équation (3.26) ne peut directement être utilisée dans notre démarche visant à déter-

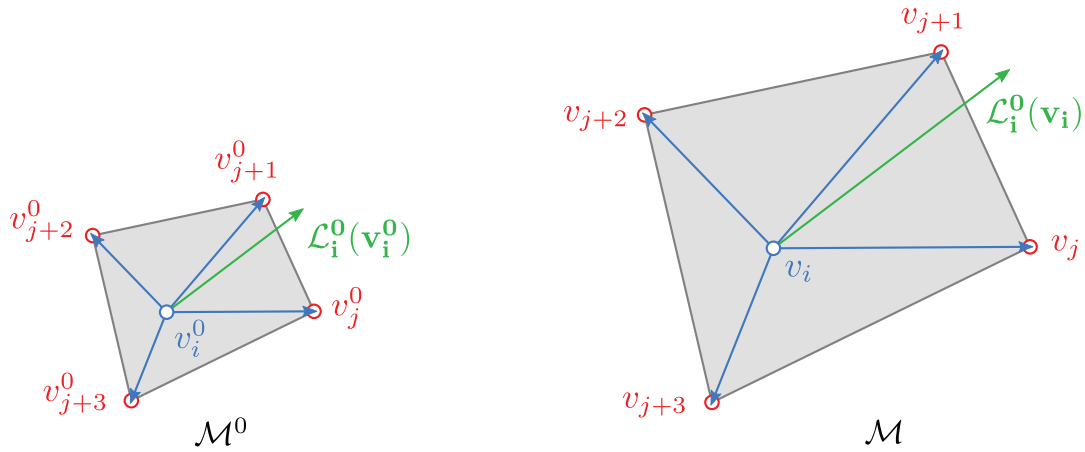


FIGURE 3.20 – Comparaison des *Laplaciens*  $\mathcal{L}_i^0(\mathbf{v}_i^0)$  et  $\mathcal{L}_i^0(\mathbf{v}_i)$  dans le cas d’une homothétie, montrant que le *Laplacien* est sensible aux homothéties.

miner la variation de forme locale  $\delta\mathcal{L}_i$  : en effet,  $T$  décrit la transformation globale (rotation et/ou homothétie) distinguant deux maillages  $\mathcal{M}^0$  et  $\mathcal{M}$  similaires. Cette équation indique donc la similarité totale entre deux maillages, mais pas la similarité locale. Cet aspect implique que l’usage de la transformation  $T$  ne permettrait pas avec deux maillages différents, mais localement similaires, d’égaliser les termes de l’équation (3.26) dans ces zones précisément similaires. Afin d’illustrer notre propos, nous présentons sur la figure 3.21 un exemple où deux maillages différents possèdent une zone strictement similaire et nous évaluons sur chacune d’entre elles les *Laplaciens* (nous choisissons dans cet exemple pour coefficients  $\omega_i = 1$ ). Sans surprise, les mesures vectorielles obtenues sont identiques (puisque la zone d’évaluation est rigoureusement similaire sur les deux maillages). Toutefois, étant donné que les maillages sont différents, la transformation  $T$  ne devrait en aucun cas être unitaire. (En toute rigueur,  $T$  est indéfini si les maillages sont différents, puisque il s’agit d’une transformation affine qui préserve donc les formes. Toutefois, on pourrait choisir de déterminer  $T$  par l’usage d’une méthode qui calcule la rotation et/ou homothétie moyenne entre deux maillages différents. Nous présenterons par ailleurs cette méthode dans le paragraphe suivant). Si l’on assume que  $T$  est défini et non-unitaire, alors les termes de l’équation (3.26) seront inégaux après substitution des *Laplaciens* obtenus et de  $T$ .

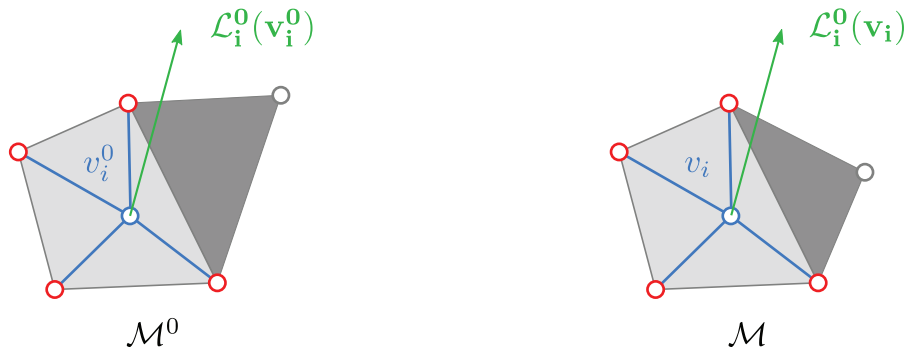


FIGURE 3.21 – Exemple de deux maillages différents  $\mathcal{M}^0$  et  $\mathcal{M}$ , mais localement similaires sur la zone définie par les triangles gris clairs. Les *Laplaciens*  $\mathcal{L}_i^0(\mathbf{v}_i^0)$  et  $\mathcal{L}_i^0(\mathbf{v}_i)$  évalués dans cette zone sont identiques sur les deux maillages.

Aussi, on peut exprimer la similarité locale en remplaçant dans (3.26), le terme  $T$  par la transformation  $T_i$  qui décrit les éventuelles rotation et/ou homothétie se produisant localement lors de la déformation de  $\mathcal{M}^0$  en  $\mathcal{M}$ . De cette manière, si les points  $v_i^0$  et  $v_i$  appartiennent respectivement à deux zones similaires des maillages  $\mathcal{M}^0$  et  $\mathcal{M}$ , on doit alors respecter :

$$T_i \mathcal{L}_i^0(\mathbf{v}_i^0) = \mathcal{L}_i^0(\mathbf{v}_i). \quad (3.27)$$

Considérant un point  $v_i^0$  et les points adjacents  $\{v_j^0\}, \forall j \in \mathcal{N}(v_i)$ , qui se déplacent respectivement aux positions  $v_i$  et  $\{v_j\}$ , nous cherchons une transformation  $T_i$  respectueuse de forme (c'est-à-dire affine) qui transforme les vecteurs  $\mathbf{v}_j^0 - \mathbf{v}_i^0$  en  $\mathbf{v}_j - \mathbf{v}_i$ . Comme on le voit sur la figure 3.22, ce critère de forme peut être impossible à respecter parfaitement, ce qui est toujours le cas en pratique pour des images sismiques réelles à restaurer. On cherche alors la transformation qui minimise l'erreur quadratique, qui s'écrit :

$$J_{T_i} = \left\| \sum_{j \in \mathcal{N}(v_i)} (\mathbf{v}_j^0 - \mathbf{v}_i^0) - (\mathbf{v}_j - \mathbf{v}_i) \right\|^2. \quad (3.28)$$

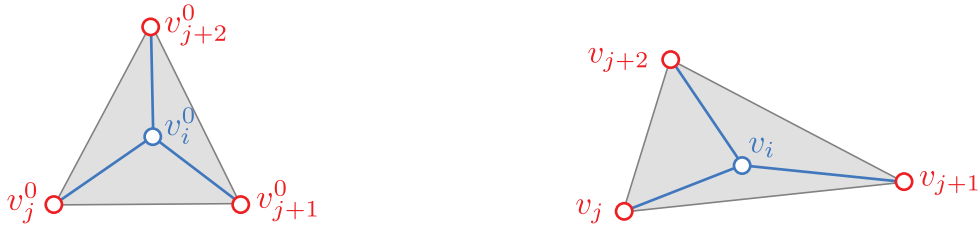


FIGURE 3.22 – Exemple d'une déformation ne pouvant être modélisée exactement par une transformation affine.

Dans la section 3.3 de [MHTG05], les auteurs détaillent la manière de calculer une telle matrice de transformation  $T_i$ . En suivant leurs résultats, la transformation au point  $v_i$  se définit par :

$$T_i = D_i(D_i^0)^{-1}, \quad (3.29)$$

avec :

$$D_i = \sum_{j \in \mathcal{N}(v_i)} (\mathbf{v}_j - \mathbf{v}_i)(\mathbf{v}_j^0 - \mathbf{v}_i^0)^T. \quad (3.30)$$

$$D_i^0 = \sum_{j \in \mathcal{N}(v_i)} (\mathbf{v}_j^0 - \mathbf{v}_i^0)(\mathbf{v}_j^0 - \mathbf{v}_i^0)^T. \quad (3.31)$$

Finalement, l'équation (3.27) permet de quantifier la variation de forme  $\delta \mathcal{L}_i$  en chaque  $v_i$  après la déformation du maillage  $\mathcal{M}$  par :

$$\delta \mathcal{L}_i = \mathcal{L}_i^0(\mathbf{v}_i) - T_i \mathcal{L}_i^0(\mathbf{v}_i^0). \quad (3.32)$$

Pour préserver la forme globale d'un maillage, nous souhaitons minimiser la somme quadratique des variations de forme locales. On écrit donc le critère  $J_S$  à minimiser :

$$J_S = \sum_{i \in [1, \dots, n]} \|\delta \mathcal{L}_i\|^2. \quad (3.33)$$

Enfin, nous réécrivons l'équation (3.33) sous forme matricielle :

$$\begin{aligned} J_S &= \sum_{i \in [1, \dots, n]} \|\mathcal{L}_i^0(\mathbf{v}_i) - T_i \mathcal{L}_i^0(\mathbf{v}_i^0)\|^2 \\ &= \|SV - s(V)\|^2, \end{aligned} \quad (3.34)$$

où la matrice  $S$  (de taille  $n \times n$ ) contient les poids  $w_j^0$  déterminés par (3.25), la matrice  $V$  (de taille  $n \times 2$ ) est définie dans l'équation (3.5) et la fonction  $s(V)$  renvoie une matrice (de taille  $n \times 2$ ) contenant les *Laplaciens* de référence auxquels s'appliquent les transformations locales respectueuses de formes définies dans l'équation (3.29). On écrit,

$$S_{r,s} = \begin{cases} \omega_j^0 & \text{lorsque } v_i \text{ est le } r^{\text{ème}} \text{ élément de } \mathcal{V}, \text{ et } v_j \text{ est le } s^{\text{ème}} \text{ élément de } \mathcal{V}, \\ - \sum_{j \in \mathcal{N}(v_i)} \omega_j^0 & \text{lorsque } v_i \text{ est le } r^{\text{ème}} \text{ élément de } \mathcal{V}, \text{ et le } s^{\text{ème}} \text{ élément de } \mathcal{V}, \\ 0 & \text{sinon.} \end{cases} \quad (3.35)$$

et

$$s(V)_i = T_i \mathcal{L}_i^0(\mathbf{v}_i^0). \quad (3.36)$$

Ayant présenté le critère général de la conservation de forme d'un maillage, nous allons à présent considérer quelques subtilités, selon la position des points appartenant à ce maillage.

### a. Points situés à l'intérieur de la triangulation

Considérant un maillage  $\mathcal{M}$ , notons  $\mathcal{V}$  l'ensemble de ses points,  $\mathcal{V}_p$  le sous-ensemble de  $\mathcal{V}$  contenant les points du contour  $\mathcal{M}$ .  $\mathcal{V}_g$  est le sous-ensemble de  $\mathcal{V}$  contenant les points situés à l'intérieur du maillage, c'est-à-dire n'appartenant pas à  $\mathcal{V}_p$ . La figure 3.23 illustre ces notations.

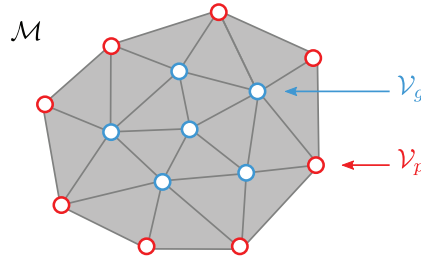


FIGURE 3.23 – Distinction entre les points intérieurs  $\mathcal{V}_g$  d'un maillage  $\mathcal{M}$  et les points  $\mathcal{V}_p$  de son contour.

Considérant à présent le sous-ensemble  $\{v_i\} \in \mathcal{V}_g$ , lorsque nous calculons en ces points une mesure de forme de référence  $\mathcal{L}_i^0$  par l'équation (3.25), les coefficients  $\omega_j^0$  déterminés satisfont [Flo03] :

$$\mathbf{v}_i - \sum_{j \in \mathcal{N}(v_i)} \omega_j \mathbf{v}_j^0 = 0. \quad (3.37)$$

Cela implique, d'après (3.25), que pour tout  $v_i \in \mathcal{V}_g$  nous avons  $\|\mathcal{L}_i(\mathbf{v}_i)\| = 0$ . Cette propriété est intéressante car lors d'une transformation respectueuse de forme qui déforme  $\mathcal{V}^0$  en  $\mathcal{V}$  elle permet de garantir :

$$\forall v_i \in \mathcal{V}_g, \|\mathcal{L}_i^0(\mathbf{v}_i^0)\| = \|\mathcal{L}_i^0(\mathbf{v}_i)\| = 0. \quad (3.38)$$

Si l'on reconsidère à présent l'équation (3.32), on constate que cette égalité rend inutile pour tout  $\{v_i\} \in \mathcal{V}_g$  l'introduction d'une matrice de transformation  $T_i$ . Nous pouvons en conséquence simplifier cette équation en :

$$\delta \mathcal{L}_i = \mathcal{L}_i^0(\mathbf{v}_i), \quad \forall v_i \in \mathcal{V}_g \quad (3.39)$$

et nous pouvons alors enfin écrire le critère  $J_{S_g}$  que nous aurons à minimiser pour les points de l'intérieur du maillage :

$$\begin{aligned} J_{S_g} &= \sum_{v_i \in \mathcal{V}_g} \|\mathcal{L}_i^0(\mathbf{v}_i)\|^2 \\ &= \|S_g V\|^2, \quad \forall v_i \in \mathcal{V}_g, \end{aligned} \quad (3.40)$$

où la matrice  $S_g$  ( $card(\mathcal{V}_g) \times n$ ) contient les poids  $w_j$  déterminés par l'équation (3.25) pour tous les  $v_i \in \mathcal{V}_g$ . On écrit :

$$S_{g,r,s} = \begin{cases} \omega_j^0 \text{ lorsque } v_i \text{ est le } r^{\text{ème}} \text{ élément de } \mathcal{V}_g, \text{ et } v_j \text{ est le } s^{\text{ème}} \text{ élément de } \mathcal{V}, \\ - \sum_{j \in \mathcal{N}(v_i)} \omega_j^0 \text{ lorsque } v_i \text{ est le } r^{\text{ème}} \text{ élément de } \mathcal{V}_g, \text{ et le } s^{\text{ème}} \text{ élément de } \mathcal{V}, \\ 0 \text{ sinon.} \end{cases} \quad (3.41)$$

## b. Points situés sur le contour de la triangulation

Dans l'équation (3.22) nous calculons les coefficients cotangents  $w_j$ . Dans le cas général, ces coefficients dépendent des valeurs d'angle  $\alpha$  et  $\beta$  définis par  $\alpha = \widehat{v_i v_{j-1} v_j}$  et  $\beta = \widehat{v_j v_{j+1} v_i}$ . Or, comme le montre la figure 3.24, ces angles peuvent être indéterminés si  $v_i \in \mathcal{V}_p$  c'est-à-dire lorsque le point de calcul est situé sur le contour du maillage. Plus précisément,  $\alpha$  est indéterminé si  $v_{j-1}$  est indéfini (figure 3.24.b), et  $\beta$  est indéterminé si  $v_{j+1}$  est indéfini (figure 3.24.c).

Lorsque ce cas se produit, il est admis dans la littérature [MDS<sup>+</sup>02] de simplement ignorer le terme



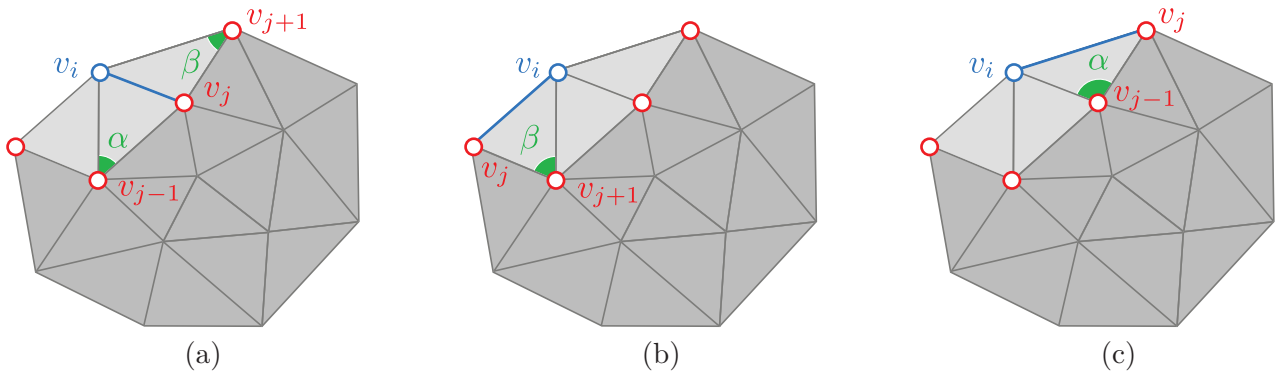


FIGURE 3.24 – Difficultés pour calculer des coefficients cotangents pour  $v_i \in \mathcal{V}_p$ , c'est-à-dire lorsque le point de calcul est situé sur le contour du maillage. (a)  $\alpha$  et  $\beta$  sont déterminés, (b) seul  $\beta$  est déterminé, (c) seul  $\alpha$  est déterminé.

ne pouvant être calculé. En conséquence, nous réécrivons l'équation (3.22) en ignorant ces termes :

$$\omega_j = \frac{\lambda_j}{\sum_{k \in \mathcal{N}(v_i)} \lambda_k}, \quad \lambda_j = a + b$$

$$a = \begin{cases} \cot(\alpha) & \text{si } v_{j-1} \text{ est défini,} \\ 0 & \text{sinon.} \end{cases} \quad (3.42)$$

$$b = \begin{cases} \cot(\beta) & \text{si } v_{j+1} \text{ est défini,} \\ 0 & \text{sinon.} \end{cases}$$

On remarque que comme conséquence du fait d'avoir ignoré les points du contour, les normes locales des mesures de *Laplacien* aux points  $\{v_i\} \in \mathcal{V}_p$  de ce contour ne sont pas systématiquement nulles, contrairement à celles mesurées aux points  $\{v_i\} \in \mathcal{V}_g$  intérieurs au domaine. L'utilisation d'une matrice de transformation  $T_i$  est donc requise en pratique en tous les points du contour lors de l'expression de la variation de forme, étant donné que les normes des *Laplaciens* ne sont pas nulles. En revanche cette matrice est superflue pour les points intérieurs, puisqu'elle n'aurait alors aucun effet en transformant des vecteurs de norme nulle.

### 3.2.4.2 Préservation des longueurs d'arêtes

La différence des *Laplaciens* permet de mesurer les variations de forme des triangles, mais comme nous l'avons expliqué dans la section 3.2.4.1 et formalisé dans l'équation (3.27), la mesure utilisée est insensible aux homothéties. Elle ne permet donc pas de conserver les aires des triangles. De ce fait, afin de respecter le critère de conservation des aires, nous allons compléter la contrainte de forme par une contrainte de préservation des longueurs d'arête.

Pour cela, considérons une arête  $\overrightarrow{v_i^0 v_j^0}$  transformée par déformation en une nouvelle arête  $\overrightarrow{v_i v_j}$  et intéressons-nous à une mesure d'erreur entre les normes euclidiennes des deux vecteurs. D'un point de

vue scalaire, si l'on définit  $l_{ij}^0 = \|\overrightarrow{v_i^0 v_j^0}\|$  et  $l_{ij} = \|\overrightarrow{v_i v_j}\|$ , on peut noter l'erreur scalaire  $e_{Lij}$  à déduire de la longueur de  $\overrightarrow{v_i v_j}$ , pour obtenir la longueur de  $\overrightarrow{v_i^0 v_j^0}$  et écrire :

$$l_{ij} - e_{Lij} = l_{ij}^0. \quad (3.43)$$

La même démarche d'un point de vue vectoriel consiste à définir une erreur vectorielle  $\overrightarrow{\delta_{Lij}}$  telle que si l'on déduit  $\overrightarrow{\delta_{Lij}}$  de  $\overrightarrow{v_i v_j}$ , alors le vecteur obtenu a même norme que  $\overrightarrow{v_i^0 v_j^0}$ . On note ainsi :

$$\|\overrightarrow{v_i v_j} - \overrightarrow{\delta_{Lij}}\| = \|\overrightarrow{v_i^0 v_j^0}\| \quad (3.44)$$

ou

$$\|\mathbf{v}_j - \mathbf{v}_i - \delta_{Lij}\| = \|\mathbf{v}_j^0 - \mathbf{v}_i^0\|, \quad (3.45)$$

que l'on peut également écrire

$$\|\mathbf{v}_j - \mathbf{v}_i - \delta_{Lij}\| = \left\| \frac{l_{ij}^0}{l_{ij}} (\mathbf{v}_j - \mathbf{v}_i) \right\|. \quad (3.46)$$

À présent, pour déterminer  $\delta_{Lij}$ , nous devons nous affranchir des écritures de normes euclidiennes dans l'équation (3.46). Il est possible de les supprimer en assurant que toutes les écritures vectorielles ont même orientation, c'est-à-dire assumer que le vecteur  $\overrightarrow{\delta_{Lij}}$  est colinéaire à  $\overrightarrow{v_i v_j}$ . Cette assumption est possible, puisque nous cherchons à déterminer  $\overrightarrow{\delta_{Lij}}$  et pouvons en conséquence lui imposer cette propriété. On écrit donc :

$$(3.46) \Leftrightarrow \mathbf{v}_j - \mathbf{v}_i - \delta_{Lij} = \frac{l_{ij}^0}{l_{ij}} (\mathbf{v}_j - \mathbf{v}_i), \quad \text{avec } \mathbf{v}_j - \mathbf{v}_i = k \delta_{Lij} \text{ et } k \in \mathbb{R}, \quad (3.47)$$

et l'erreur vectorielle  $\delta_{Lij}$  s'écrit :

$$\delta_{Lij} = (\mathbf{v}_j - \mathbf{v}_i) - \frac{l_{ij}^0}{l_{ij}} (\mathbf{v}_j - \mathbf{v}_i). \quad (3.48)$$

Essayer de conserver l'ensemble des longueurs des arêtes du maillage consiste alors à minimiser la somme quadratique des erreurs vectorielles aux longueurs euclidiennes. On définit en conséquence le critère  $J_L$  à minimiser :

$$J_L = \sum_{(i,j) \in \mathcal{E}} \|\delta_{Lij}\|^2 \quad (3.49)$$

et nous réécrivons par commodité le critère  $J_L$  sous la forme matricielle :

$$\begin{aligned} J_L &= \sum_{(i,j) \in \mathcal{E}} \left\| (\mathbf{v}_j - \mathbf{v}_i) - \frac{l_{ij}^0}{l_{ij}} (\mathbf{v}_j - \mathbf{v}_i) \right\|^2 \\ &= \|LV - l(V)\|^2, \end{aligned} \quad (3.50)$$

où  $L$  est la matrice (de taille  $\text{card}(\mathcal{E}) \times n$ ) d'adjacence orientée (c'est-à-dire une matrice décrivant

un graphe où chaque ligne définit une connexion orientée et indique les indices des nœuds où débute et finit la connexion),  $V$  la matrice (de taille  $n \times 2$ ) des positions définie dans l'équation (3.5) et  $l(V)$  la fonction qui renvoie la matrice (de taille  $\text{card}(\mathcal{E}) \times 2$ ) contenant les quantités vectorielles  $\left\{ \frac{l_{ij}^0}{l_{ij}}(\mathbf{v}_j - \mathbf{v}_i) \right\}$ . On écrit :

$$L_{r,s} = \begin{cases} 1 & \text{lorsque } (i,j) \text{ est le } r^{\text{ème}} \text{ élément de } \mathcal{E} \text{ et } v_i \text{ le } s^{\text{ème}} \text{ élément de } \mathcal{V}, \\ -1 & \text{lorsque } (i,j) \text{ est le } r^{\text{ème}} \text{ élément de } \mathcal{E} \text{ et } v_j \text{ le } s^{\text{ème}} \text{ élément de } \mathcal{V}, \\ 0 & \text{sinon.} \end{cases} \quad (3.51)$$

$$V_i = [x_i, y_i] \quad (3.52)$$

$$l(V)_r = \left[ \frac{l_{ij}^0}{l_{ij}}(\mathbf{v}_j - \mathbf{v}_i)_x, \frac{l_{ij}^0}{l_{ij}}(\mathbf{v}_j - \mathbf{v}_i)_y \right], \text{ lorsque } (i,j) \text{ est le } r^{\text{ème}} \text{ élément de } \mathcal{E}. \quad (3.53)$$

La figure 3.25 montre un exemple de maillage ainsi que l'indexage de ses points. Dans cet exemple la première ligne des matrices  $L$  et  $l(V)$  contient :

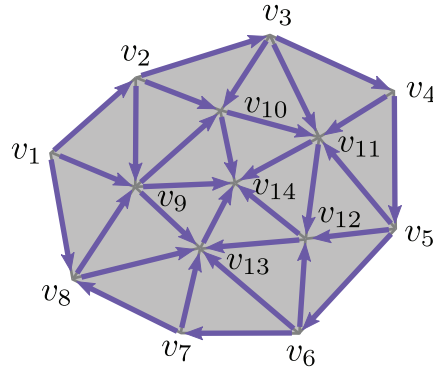


FIGURE 3.25 – Préservation globale des longueurs d'arêtes dans un maillage.

$$L_1 = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$l(V)_1 = \begin{bmatrix} \frac{l_{12}^0}{l_{12}}(v_{2x} - v_{1x}) & \frac{l_{12}^0}{l_{12}}(v_{2y} - v_{1y}) \end{bmatrix}.$$

### 3.3 Optimisation des contraintes de déformation

Dans les sections 3.1 et 3.2, nous avons présenté le processus de restauration et décrit l'ensemble des critères permettant d'assurer chaque étape de la restauration (restauration de la faille puis mise à plat de l'horizon) tout en conservant l'aspect général du maillage. Cependant, dans la majeure partie des cas, on ne peut pas respecter strictement toutes ces contraintes comme on l'illustre sur la figure 3.26.a, où il est impossible de déplier la ligne bleue sans altérer la forme des triangles adjacents. Par exemple, la ligne bleue dépliée en figure 3.26.b cause la déformation des deux triangles rosés.



FIGURE 3.26 – **Illustration de l'impossibilité de respecter strictement toutes les contraintes** : (a) On ne peut déplier la ligne bleue sans altérer la forme des triangles, (b) une possible mise à plat de la ligne bleue cause la déformation des triangles rosés.

Aussi, nous allons optimiser ces critères en déterminant un consensus satisfaisant par une minimisation conjointe, qui respecte au mieux les contraintes géométriques et honore les contraintes de restauration imposées. Nous allons ci-dessous étudier différentes options possibles pour atteindre ce but et proposons en particulier deux méthodes : une minimisation pondérée de contraintes, utilisant l'algorithme itératif de *Gauss-Newton* et une minimisation avec respect strict, basée sur la méthode des multiplicateurs de *Lagrange*.

### 3.3.1 Optimisation par minimisation pondérée des contraintes

Une première manière pour optimiser les différentes contraintes qui minimise l'erreur des contraintes géométriques et honore les contraintes de restauration consiste à effectuer une minimisation conjointe pondérée. Nous proposons en ce sens une formulation aux moindres carrés. Celle-ci n'étant pas linéaire et le système d'équation ne pouvant être résolu par une méthode numérique directe, nous utilisons l'algorithme de *Gauss-Newton* pour déterminer les positions optimales. Enfin, l'application de pondérations aux différentes contraintes permet d'assurer un respect strict des contraintes de restauration imposées.

#### 3.3.1.1 Formulation du système d'équations

Dans les paragraphes précédents, nous avons décrit chaque contrainte de restauration et chaque contrainte géométrique, par un terme quadratique quantifiant l'erreur mesurée à l'état escompté. À présent, afin de calculer la déformation globale qui minimise tous ces termes, nous les regroupons dans une formulation aux moindres carrés générique que l'on doit minimiser :

$$J = \alpha J_{S_g} + \beta J_{S_p} + \gamma J_L + \zeta J_C, \quad (3.54)$$

où  $\alpha$ ,  $\beta$ ,  $\gamma$  et  $\zeta$  sont des constantes pouvant pondérer chaque terme afin de les privilégier, ainsi qu'il est d'usage lors de formulations aux moindres carrés (par exemple dans [LTGM97]). Nous discuterons à la fin de cette section l'impact de ces pondérations et détaillerons notre choix de poids  $\alpha$ ,  $\beta$ ,  $\gamma$  et  $\zeta$ .

Pour le cas de la faille, le terme  $J_F$  à minimiser est :

$$J_F = \alpha_F J_{S_g} + \beta_F J_{S_p} + \gamma_F J_L + \zeta_F J_{F_1} + \lambda_F J_{F_2}, \quad (3.55)$$

et pour le cas de l'horizon, le terme  $J_H$  s'écrit :

$$J_H = \alpha_H J_{S_g} + \beta_H J_{S_p} + \gamma_H J_L + \zeta_H J_H, \quad (3.56)$$

où  $\alpha_F, \beta_F, \gamma_F, \zeta_F, \lambda_F, \alpha_H, \beta_H, \gamma_H, \zeta_H$  sont des constantes.

D'après les équations (3.40), (3.34), (3.50) et (3.3), on développe le terme générique  $J$  de la manière suivante :

$$J = \alpha \cdot \|S_g V\|^2 + \beta \cdot \|S_p V - s(V)\|^2 + \gamma \cdot \|LV - l(V)\|^2 + \zeta \cdot \|CV - c(V)\|^2, \quad (3.57)$$

et comme les termes de cette équation (3.57) ont une forme factorisable, on réécrit  $J$  sous la forme matricielle classique :

$$J = \|AV - b(V)\|^2, \text{ avec } A = \begin{bmatrix} \alpha \cdot S_g \\ \beta \cdot S_p \\ \gamma \cdot L \\ \zeta \cdot C \end{bmatrix}, \quad b(V) = \begin{bmatrix} 0 \\ \beta \cdot s(V) \\ \gamma \cdot l(V) \\ \zeta \cdot c(V) \end{bmatrix} \quad (3.58)$$

### 3.3.1.2 Résolution du système non-linéaire par la méthode itérative de *Gauss-Newton*

Dans l'équation (3.58), le terme  $b(V)$  n'est pas linéaire (à cause des termes  $s(V)$  et  $l(V)$ ), ce qui rend impossible toute résolution directe. Aussi, il est d'usage de résoudre un tel problème par une approche itérative. Nous avons de ce fait choisi d'utiliser, comme dans [WXW+06], une méthode classique itérative de *Gauss-Newton* [Bjo96]. L'idée de cette approche standard consiste à linéariser  $b(V)$  en le calculant pour un ensemble de positions données  $V^t$  au temps  $t$ . Le terme  $b(V^t)$  devient alors localement constant, ce qui permet de déterminer des positions  $V^{t+1}$  à l'itération  $t + 1$ , minimisant l'ensemble des contraintes énoncées par résolution directe de :

$$AV^{t+1} = b(V^t). \quad (3.59)$$

On illustre le fonctionnement de l'algorithme de *Gauss-Newton* sur la figure 3.27, où l'on présente une fonction coût  $f(x)$  non linéaire, pour laquelle on recherche le minimum  $f(x_n) = \min_x f(x)$ . Assumant qu'une résolution directe est impossible, on fixe une condition initiale  $x = x_0$  afin de linéariser  $f$  en ce point. La fonction ainsi linéarisée permet de déterminer par résolution directe une position  $x_1$  qui est plus proche que  $x_0$  du minimum recherché. On linéarise alors une nouvelle fois  $f$  en  $x_1$ , permettant de déterminer  $x_2$ . On procède ainsi par raffinements successifs jusqu'à s'approcher suffisamment de la valeur de  $x_n$  qui correspond au minimum de la fonction  $f(x)$ .

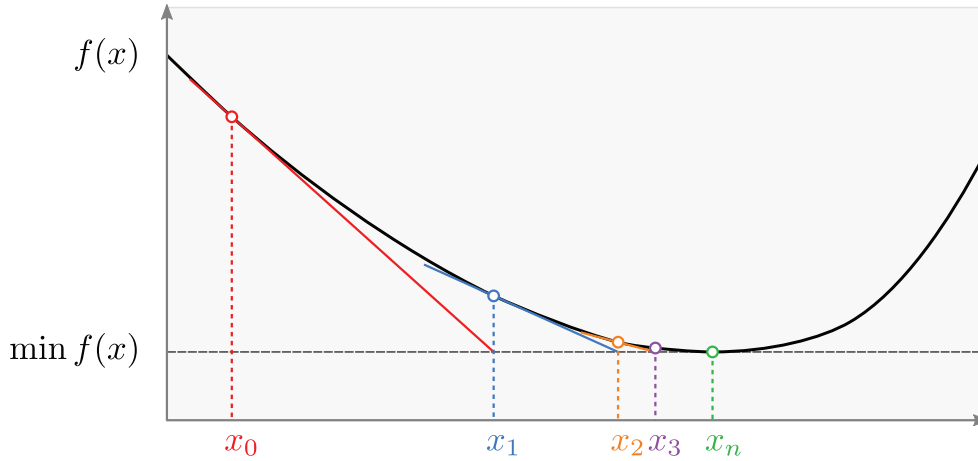


FIGURE 3.27 – Illustration du fonctionnement de l’algorithme de *Gauss-Newton* permettant de déterminer itérativement le minimum  $f(x_n)$  d’une fonction coût  $f(x)$ , en fixant comme condition initiale  $x = x_0$ .

Dans l’équation (3.59), la matrice  $A$  n’étant pas inversible, on multiplie l’égalité par la matrice  $G$ , dite “pseudo-inverse de *Moore-Penrose*” que l’on définit par :

$$G = (A^T A)^{-1} A^T, \quad (3.60)$$

ce qui nous permet alors d’obtenir  $V^{t+1}$  :

$$\begin{aligned} GAV^{t+1} &= Gb(V) \\ (A^T A)^{-1}(A^T A)V^{t+1} &= (A^T A)^{-1}A^T b(V^t) \\ V^{t+1} &= (A^T A)^{-1}A^T b(V^t). \end{aligned} \quad (3.61)$$

Les positions obtenues  $V^{t+1}$  correspondent à la solution optimale de l’équation (3.59). En ce sens, elles minimisent mieux le résultat de (3.58) que les positions  $V^t$ , c’est-à-dire qu’elles font décroître la fonction coût définie. Ainsi, en substituant  $V^{t+1}$  à  $V^t$  dans l’équation (3.59), on obtient un nouvel ensemble de positions  $V^{t+2}$  réduisant (3.58). De manière résumée, l’algorithme itératif consiste donc à résoudre :

$$\min_{t+1} \|AV^{t+1} - b(V^t)\|^2, \quad (3.62)$$

où  $V_t$  est le vecteur des positions courantes au temps  $t$  et  $V^{t+1}$  le vecteur des positions mises à jour à l’itération suivante, c’est-à-dire au temps  $t + 1$ .

Enfin, dans l’équation (3.61),  $A$  dépend du maillage avant déformation. De ce fait, la pseudo-inverse  $G = (A^T A)^{-1}A^T$  peut être pré-calculée avant la résolution itérative, ce qui simplifie significativement le calcul de chaque itération en le réduisant à une multiplication de matrices.

Comme toutes les méthodes d’optimisation itératives, notre algorithme converge en quelques itérations vers une bonne approximation initiale. Cependant, déterminer le nombre adéquat d’itérations nécessaire à la restauration d’une image sismique est important pour garantir performance et précision de la déformation obtenue. Aussi, dans la section 3.6 nous mènerons une étude plus détaillée de la convergence et nous montrerons comment choisir ce nombre d’itérations.

### 3.3.1.3 Pondération des différentes contraintes

Nous avons rencontré dans l’équation (3.54), un système composé de la somme de plusieurs termes et affectés de différents coefficients de pondération que nous pouvons utiliser pour leur donner des importances (poids) relatives différentes. Nous allons à présent décrire notre choix de coefficients de pondération, en expliquant tout d’abord que les différents termes du système expriment une grandeur comparable puis en privilégiant les contraintes imposées par une surpondération.

L’affectation de pondérations a pour but de modifier l’équilibre du système, faisant intervenir différentes grandeurs (longueurs, formes. . .). Dans un premier temps, nous devons choisir ces coefficients de manière à trouver un “bon équilibre” du système.

Il est intuitivement très difficile de déterminer le “bon équilibre” d’un système, si celui-ci fait intervenir des grandeurs de natures différentes. Par exemple, dans le contexte de la restauration, nous aurions pu choisir une approche hybride introduisant deux termes : l’un exprimant une grandeur géométrique et l’autre exprimant une grandeur géomécanique. Déterminer quelle proportion de chacun de ces deux termes conduit au “meilleur équilibre”, aurait alors consisté à privilégier l’aspect géomécanique ou géométrique d’une restauration, ce qui finalement est un choix relativement subjectif.

Dans notre application, toutes les contraintes que nous avons décrites expriment la même grandeur, c’est-à-dire une erreur quadratique de distance euclidienne. Elles sont donc comparables entre elles. Dans le cas du critère de forme, nous exprimons l’erreur de distance au barycentre d’un ensemble de points (pondérée par la distance à ces points). Dans le cas du critère de longueur d’arête, nous exprimons l’erreur de distance entre une arête initiale et une arête déformée. Enfin, pour toutes les contraintes de restauration, nous exprimons l’erreur de distance d’un point à une position restaurée. En conséquence, nous considérons dans notre application, qu’un “bon équilibre” du système consiste à accorder autant d’importance aux termes conservant l’énergie interne (contraintes de forme et de longueur) et nous privilégierons de manière significative les termes imposés (contraintes de restauration).

Ainsi, soient  $\omega_G$  et  $\omega_R$  deux scalaires positifs appliqués respectivement aux contraintes géométriques et aux contraintes de restauration, on définit pour l’équation (3.54) :

$$\left\{ \begin{array}{l} \alpha = \omega_G \\ \beta = \omega_G \\ \gamma = \omega_G \\ \zeta = \omega_R \end{array} \right. \quad (3.63)$$



et l'on définit pour les équations (3.55) et (3.56) :

$$\left\{ \begin{array}{l} \alpha_H = \omega_G \\ \beta_H = \omega_G \\ \gamma_H = \omega_G \\ \zeta_H = \omega_R \\ \alpha_F = \omega_G \\ \beta_F = \omega_G \\ \gamma_F = \omega_G \\ \zeta_F = \omega_R \\ \lambda_F = \omega_R \end{array} \right. . \quad (3.64)$$

Ensuite, afin d'établir une surpondération des contraintes de restauration, nous introduisons un scalaire positif  $k$  tel que  $k > 1$  qui décrit l'amplitude de cette surpondération et écrivons :

$$\omega_R = k\omega_G. \quad (3.65)$$

La valeur de  $\omega_G$  peut alors à ce stade être fixée arbitrairement et n'aura aucune influence.

Enfin, bien qu'il soit direct que plus  $k$  est élevé, plus le respect aux contraintes imposées est bon (puisque on augmente alors l'amplitude de la surpondération) déterminer la valeur de  $k$  reste expérimental, comme généralement avec les minimisations au moindres carrés (par exemple dans [WXW<sup>+</sup>06], [HSL<sup>+</sup>06] ou [LTGM97]). Dans notre application, nous considérons que  $k = 10$  est un minimum pour obtenir un respect acceptable des contraintes de restauration et que  $k = 100$  est une bonne valeur. Il est de plus possible de laisser l'utilisateur contrôler ce paramètre afin de pouvoir garantir un résultat satisfaisant.

### 3.3.2 Optimisation avec respect strict de contraintes

La méthode de minimisation pondérée que nous venons de présenter produit en pratique des résultats honorant les contraintes imposées (avec des coefficients correctement choisis). Toutefois, d'aucun pourrait objecter que le respect strict de ces contraintes n'a pas été défini au sens mathématique, aussi nous proposons une méthode alternative insistant sur ce point.

Dans [WXW<sup>+</sup>06] et [HSL<sup>+</sup>06] les auteurs proposent une autre approche permettant de respecter strictement un jeu de contraintes, en reformulant le problème d'optimisation par l'introduction et la détermination de multiplicateurs de *Lagrange* pour des contraintes imposées. Puis des contraintes restantes sont optimisées au mieux par un processus de minimisation similaire à celui détaillé plus haut. Nous allons donc présenter ci-dessous l'adaptation que nous avons faite de leur méthode au contexte de notre application de restauration, en déterminant les multiplicateurs de *Lagrange* pour les contraintes de restauration puis minimisant les contraintes géométriques. Bien que la même méthode de résolution s'applique aux cas de la restauration d'une faille et de la mise à plat d'un horizon, nous

préciserons dans un premier temps la différence entre ces deux applications qui se distinguent lors de la définition de la contrainte imposée.

### 3.3.2.1 Formulation des contraintes imposées

Les contraintes de mise à plat de l'horizon et de contact à la faille, telles que définies dans les équations (3.14) et (3.10) ne sont pas applicables avec une méthode d'optimisation avec respect strict des contraintes. En effet, dans chacun de ces deux cas, on détermine pour les points  $\{v_{H_i}\}$  (appartenant à l'horizon) ou  $\{v_{F_i}\}$  (appartenant à la faille) une position projetée, qui est une position proche de celle que l'on souhaite obtenir après restauration, mais qui à-priori n'est pas la position restaurée. Ce critère fonctionne très bien avec la méthode de minimisation pondérée, car les positions projetées sont recalculées à chaque itération de l'algorithme et on converge alors pour chacun des points dans les deux cas vers une position projetée optimale. Ce comportement est rendu possible par la formulation mathématique n'imposant pas de respect strict sur les contraintes imposées. Ainsi, à chaque itération, l'optimisation détermine pour les points  $\{v_{H_i}\}$  ou  $\{v_{F_i}\}$ , une position respectivement proche de l'horizon ou la faille, mais qui autorise également les déplacements latéraux. Au fur et à mesure de la convergence, l'écart à la faille ou l'horizon se réduit afin de d'honorer la contrainte de restauration qui est privilégiée. Ceci nous amène à considérer que de tels déplacement latéraux sont interdits si l'on impose un respect strict, et que nous imposerions alors le déplacement des points  $\{v_{H_i}\}$  ou  $\{v_{F_i}\}$  sur des positions projetées à-priori fausses. Aussi nous devons décrire différemment les contraintes de mise à plat de l'horizon et de contact à la faille pour le cas d'une minimisation avec respect strict de contraintes.

#### a. Reformulation du critère de contact à l'horizon

Pour éviter d'imposer le déplacement des points de l'horizon à des positions fausses, on reformule la contrainte de l'horizon par une approche différente. Soit un point  $v_{H_i} \in \mathcal{V}_H$ , la mise à plat de l'horizon consiste à projeter ce point par l'application d'une fonction de projection  $f(v_{H_i})$  sur une droite  $D_H$  d'équation  $y = y_H$ . Comme on l'illustre sur la figure 3.28.a, il existe alors plusieurs projetés possibles  $f_1(v_{H_i})$ ,  $f_2(v_{H_i})$  et  $f_3(v_{H_i})$  qui satisfont la contrainte de mise à plat.

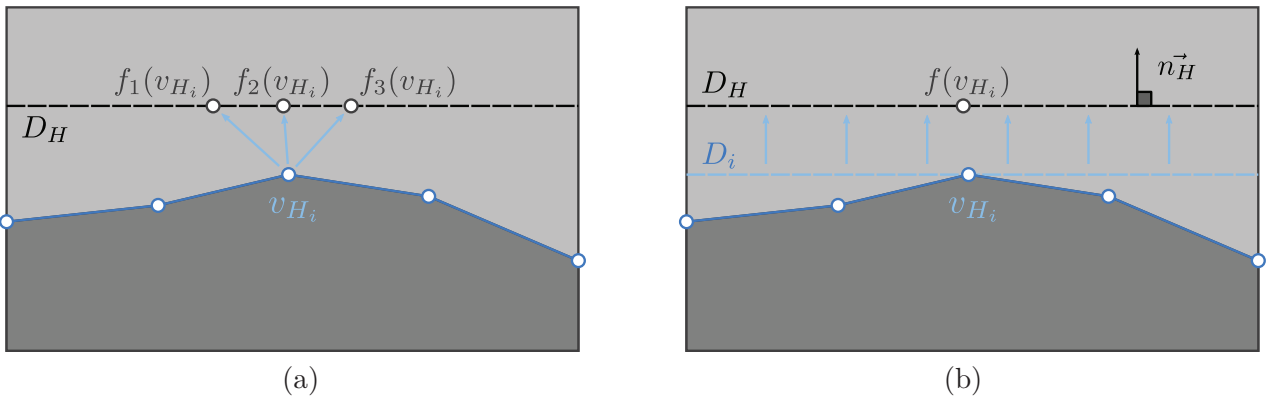


FIGURE 3.28 – Exemple de la contrainte de mise à plat de l'horizon.

Ainsi, si l'on définit un vecteur  $\overrightarrow{n_{H_i}}$ , normal à  $D_H$  et la droite  $D_i$  passant par  $v_{H_i}$  normale à  $\overrightarrow{n_{H_i}}$ , satisfaire la contrainte de mise à plat au point  $v_{H_i}$  consiste à assurer (voir figure 3.28.b) :

$$D_i = D_H. \quad (3.66)$$

Cette contrainte est respectée si  $f(v_{H_i}) \in D_i$ , ou d'un point de vue vectoriel, si le vecteur  $\overrightarrow{v_{H_i} f(v_{H_i})}$  est normal au vecteur  $\overrightarrow{n_{H_i}}$  (c'est-à-dire si le produit scalaire de ces deux vecteurs est nul). On peut donc écrire :

$$\begin{aligned} (3.66) \quad &\Leftrightarrow (\mathbf{v}_{H_i} - \mathbf{f}(\mathbf{v}_{H_i})) \cdot \mathbf{n}_H = 0 \\ &\Leftrightarrow \mathbf{v}_{H_i} \cdot \mathbf{n}_H - \mathbf{f}(\mathbf{v}_{H_i}) \cdot \mathbf{n}_H = 0. \end{aligned} \quad (3.67)$$

En utilisant les matrices  $C_H$  et  $V$  introduites respectivement dans les équations (3.17) et (3.5) puis introduisant deux nouvelles matrices  $N_H$  et  $P_H$ , on réécrit (3.67) sous la forme matricielle suivante :

$$C_{H_i} V \cdot N_{H_i} - P_{H_i} = 0 \quad (3.68)$$

où  $C_H$  est la matrice (de taille  $\text{card}(\mathcal{V}_h) \times n$ ) définie en équation (3.17),  $V$  est la matrice des positions (de taille  $n \times 2$ ) définie en équation (3.5),  $N_H$  une matrice (de taille  $2 \times \text{card}(\mathcal{V}_h)$ ) dont chaque colonne contient le vecteur normal  $\overrightarrow{n_{H_i}}$  et  $P_H$  est une matrice (de taille  $\text{card}(\mathcal{V}_h) \times 1$ ) contenant les produits scalaires  $\mathbf{f}(\mathbf{v}_{H_i}) \cdot \mathbf{n}_H$ . On définit alors l'écriture des matrices  $N_H$  et  $P_H$  :

$$(N_{H_i})^T = [(n_H)_x, (n_H)_y]_i, \forall v_i \in \mathcal{V}_h, \quad (3.69)$$

$$P_{H_i} = [\mathbf{f}(\mathbf{v}_{H_i}) \cdot \mathbf{n}_H]_i, \forall v_i \in \mathcal{V}_h. \quad (3.70)$$

où  $M_i$  indique la  $i^{\text{ème}}$  ligne d'une matrice  $M$ .

## b. Reformulation du critère de contact à la faille

Reformulons à présent la contrainte de contact à la faille en suivant la même approche. Soit un point  $v_{F_i} \in \mathcal{V}_F$ , la restauration de la faille consiste à projeter ce point par l'application d'une fonction de projection  $f(v_{F_i})$  sur la faille. On définit la droite  $D_{F_i}$  passant par le segment de la faille, sur lequel a été déterminé la position  $f(v_{F_i})$ . On considère alors que toutes les positions appartenant à  $D_{F_i}$  sont une solution possible (voir figure 3.29). On remarque immédiatement que cette méthode ne contraint pas le déplacement au segment de la faille, mais l'autorise sur l'ensemble de la droite  $D_{F_i}$ . En d'autres termes, on autorise qu'une position extérieure à la faille soit solution. En pratique, cela ne pose aucun problème car la contrainte de préservation des longueurs évite les déplacements latéraux trop importants et car l'optimisation est itérative, impliquant qu'une nouvelle projection  $f(v_{F_i})$  sera déterminée à la prochaine itération et repositionnera le point sur la faille. Si quelques erreurs peuvent être remarquables lors des premières itérations, elles disparaissent au fur et à mesure que le système converge et se stabilise.

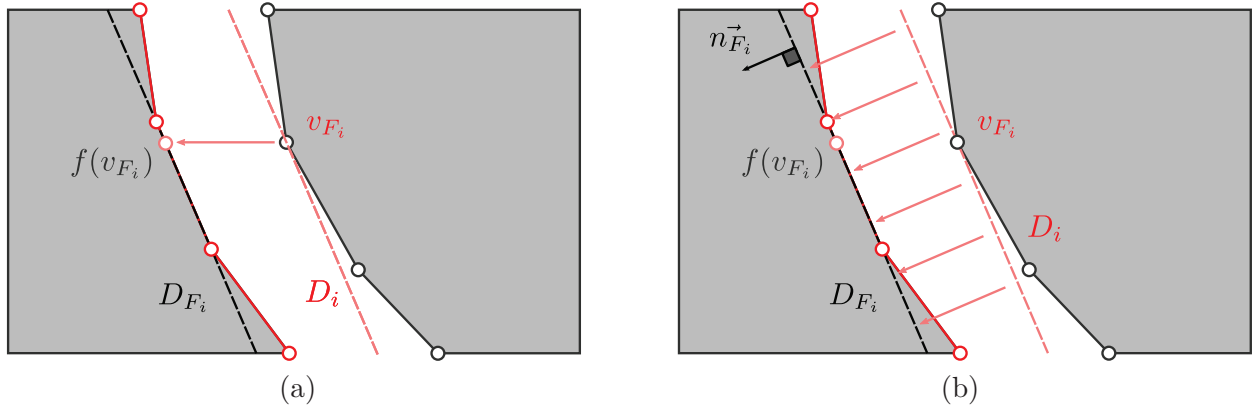


FIGURE 3.29 – Exemple de la contrainte de restauration de la faille.

Ainsi, si l'on définit un vecteur  $\vec{n}_{F_i}$ , normal à  $D_{F_i}$  et la droite  $D_i$  passant par  $v_{F_i}$  normale à  $\vec{n}_{F_i}$ , satisfaire la contrainte de mise à plat au point  $v_{F_i}$  consiste à assurer (voir figure 3.28.b) :

$$D_i = D_{F_i}. \quad (3.71)$$

Comme précédemment, nous respectons cette contrainte en assurant que le produit scalaire des vecteurs  $\vec{v}_{F_i} f(v_{F_i})$  et  $\vec{n}_{F_i}$  soit nul :

$$\begin{aligned} (3.71) &\Leftrightarrow (\mathbf{v}_{F_i} - \mathbf{f}(\mathbf{v}_{F_i})) \cdot \mathbf{n}_{F_i} = 0 \\ &\Leftrightarrow \mathbf{v}_{F_i} \cdot \mathbf{n}_{F_i} - \mathbf{f}(\mathbf{v}_{F_i}) \cdot \mathbf{n}_{F_i} = 0. \end{aligned} \quad (3.72)$$

En utilisant les matrices  $C_F$  et  $V$  introduites respectivement dans les équations (3.11) et (3.5) puis introduisant deux nouvelles matrices  $N_F$  et  $P_F$ , on réécrit (3.72) sous la forme matricielle suivante :

$$C_{F_i} V \cdot N_{F_i} - P_{F_i} = 0 \quad (3.73)$$

où  $C_F$  est la matrice (de taille  $\text{card}(\mathcal{V}_f) \times n$ ) définie en équation (3.11),  $V$  est la matrice des positions (de taille  $n \times 2$ ) définie en équation (3.5),  $N_F$  une matrice (de taille  $2 \times \text{card}(\mathcal{V}_f)$ ) dont chaque colonne contient le vecteur normal  $\vec{n}_{F_i}$  et  $P_F$  est une matrice (de taille  $\text{card}(\mathcal{V}_f) \times 1$ ) contenant les produits vectoriels  $\mathbf{f}(\mathbf{v}_{F_i}) \cdot \mathbf{n}_{F_i}$ . On définit alors l'écriture des matrices  $N_F$  et  $P_F$  :

$$(N_{F_i})^T = [(n_{F_i})_x, (n_{F_i})_y]_i, \forall v_i \in \mathcal{V}_f, \quad (3.74)$$

$$P_{F_i} = [\mathbf{f}(\mathbf{v}_{F_i}) \cdot \mathbf{n}_{F_i}]_i, \forall v_i \in \mathcal{V}_f. \quad (3.75)$$

où  $M_i$  indique la  $i^{\text{ème}}$  ligne d'une matrice  $M$ .

### 3.3.2.2 Généralisation des notations

Comme on le voit dans les équations (3.68) et (3.73), les critères obtenus ont une forme similaire conduisant à une méthode d'optimisation identique pour les deux opérations de restauration. Dans

un but de simplification, nous proposons d'introduire quelques notations communes, qui permettront de présenter une résolution générique du système de contraintes.

Les deux premiers termes que nous devons redéfinir sont  $A$  et  $b(V)$  issus de l'équation (3.58) de la minimisation conjointe. Ayant redéfini les contraintes imposées, nous devons en effet extraire les contraintes précédentes de ces matrices. Nous réécrivons donc les critères  $J_F$  pour le cas de la faille et  $J_H$  pour le cas de l'horizon :

$$J_F = \|AV - b(V)\|^2, \text{ avec } A = \begin{bmatrix} \alpha_F \cdot S_g \\ \beta_F \cdot S_p \\ \gamma_F \cdot L \\ \zeta_F \cdot C_I \end{bmatrix}, \quad b(V) = \begin{bmatrix} 0 \\ \beta_F \cdot s(V) \\ \gamma_F \cdot l(V) \\ \zeta_F \cdot \mathbf{v}_I \end{bmatrix} \quad (3.76)$$

$$J_H = \|AV - b(V)\|^2, \text{ avec } A = \begin{bmatrix} \alpha_H \cdot S_g \\ \beta_H \cdot S_p \\ \gamma_H \cdot L \end{bmatrix}, \quad b(V) = \begin{bmatrix} 0 \\ \beta_H \cdot s(V) \\ \gamma_H \cdot l(V) \end{bmatrix} \quad (3.77)$$

Il est remarquable que nous laissons la contrainte de continuité de l'horizon dans la 4<sup>ème</sup> ligne de l'équation (3.76) de minimisation conjointe pour le cas de la faille. En effet, satisfaire cette contrainte consiste à effectuer une simple translation du maillage et aucune autre contrainte n'impose telle translation. Ainsi, il existe donc une solution unique et la contrainte de continuité de l'horizon ne requiert pas d'être définie strictement.

À présent, nous généralisons les équations (3.68) et (3.73). Soit  $C$  la matrice (de taille  $n_c \times n$ ) permettant d'extraire les positions de l'ensemble de points auxquels s'applique la contrainte de restauration imposée. On définit :

$$C = \begin{cases} C_F \text{ lors de l'opération de restauration de la faille,} \\ C_H \text{ lors de l'opération de mise à plat de l'horizon} \end{cases} \quad (3.78)$$

et :

$$n_c = \begin{cases} \text{card}(\mathcal{V}_f) \text{ lors de l'opération de restauration de la faille,} \\ \text{card}(\mathcal{V}_h) \text{ lors de l'opération de mise à plat de l'horizon.} \end{cases} \quad (3.79)$$

Soit  $N$  la matrice contenant les vecteurs normaux à la droite  $D$ . On définit :

$$N = \begin{cases} N_f \text{ lors de l'opération de restauration de la faille,} \\ N_h \text{ lors de l'opération de mise à plat de l'horizon.} \end{cases} \quad (3.80)$$

Enfin, soit  $P$  la matrice contenant les points passant par les différentes droites. On définit :

$$P = \begin{cases} P_f \text{ lors de l'opération de restauration de la faille,} \\ P_h \text{ lors de l'opération de mise à plat de l'horizon.} \end{cases} \quad (3.81)$$

### 3.3.2.3 Formulation du système d'équations

Ayant défini les termes nécessaires, on reprend le système d'équations à minimiser donné par (3.58) auquel nous ajoutons les critères à respecter pour assurer un respect strict des contraintes. Nous cherchons donc :

$$\begin{cases} \min \|AV - b(V)\|^2 \\ \text{satisfaisant : } C_i V N_i - P_i = 0, \forall i \in [1, \dots, n_c]. \end{cases} \quad (3.82)$$

Nous allons en réalité chercher à résoudre (3.82) sous la forme suivante, mieux adaptée à la procédure de minimisation que l'on souhaite mettre en œuvre :

$$\begin{cases} \min \frac{1}{2} \|AV - b(V)\|^2 \\ \text{satisfaisant : } C_i V N_i - P_i = 0, \forall i \in [1, \dots, n_c]. \end{cases} \quad (3.83)$$

Nous formulons à présent la minimisation sous contraintes (3.83) en introduisant des multiplicateurs de *Lagrange*  $\{\lambda_i\}$  (décrits en section 7.2 de [Cia88]) que nous devons déterminer pour assurer un respect strict des contraintes :

$$\begin{cases} \min \left( \frac{1}{2} \|AV - b(V)\|^2 + \sum_{i=1}^{n_c} \lambda_i (C_i V N_i - P_i) \right) \\ \text{satisfaisant : } C_i V N_i - P_i = 0, \forall i \in [1, \dots, n_c]. \end{cases} \quad (3.84)$$

De même que dans l'équation (3.62), le caractère non linéaire du terme  $b(V)$  rend impossible une résolution directe de ce système et nous utilisons donc à nouveau une résolution itérative de *Gauss-Newton*. Dans ce but nous reformulons donc l'équation précédente sous la forme :

$$\begin{cases} \min \left( \frac{1}{2} \|AV^{t+1} - b(V^t)\|^2 + \sum_{i=1}^{n_c} \lambda_i (C_i V^{t+1} N_i - P_i) \right) \\ \text{satisfaisant : } C_i V^{t+1} N_i - P_i = 0, \forall i \in [1, \dots, n_c]. \end{cases} \quad (3.85)$$

Nous noterons  $\mathcal{L}(V^{t+1}) = \frac{1}{2} \|AV^{t+1} - b(V^t)\|^2 + \sum_{i=1}^{n_c} \lambda_i (C_i V^{t+1} N_i - P_i)$ , la fonction *Lagrangienne* à minimiser.

### 3.3.2.4 Résolution du système d'équation

Nous minimisons cette fonction *Lagrangienne*  $\mathcal{L}(V^{t+1})$  en cherchant les termes  $V^{t+1}$  et  $\{\lambda_i, i \in [1, \dots, n_c]\}$ , tel que son gradient par rapport à  $V^{t+1}$  soit nul :

$$\begin{aligned}
\nabla_{V^{t+1}} \mathcal{L}(V^{t+1}) = 0 &\Leftrightarrow \nabla_{V^{t+1}} \left( \frac{1}{2} \|AV^{t+1} - b(V^t)\|^2 \right) + \nabla_{V^{t+1}} \left( \sum_{i=1}^{n_c} \lambda_i (C_i V^{t+1} N_i - P_i) \right) = 0 \\
&\Leftrightarrow \frac{1}{2} \times 2 \left( A^T (AV^{t+1} - b(V^t)) \right) + \sum_{i=1}^{n_c} \lambda_i (C_i)^T (N_i)^T = 0 \\
&\Leftrightarrow A^T (AV^{t+1} - b(V^t)) + \sum_{i=1}^{n_c} \lambda_i (C_i)^T (N_i)^T = 0.
\end{aligned} \tag{3.86}$$

Et en réarrangeant l'équation précédente, nous pouvons définir  $V^{t+1}$  :

$$\begin{aligned}
(3.86) \Leftrightarrow A^T AV^{t+1} - A^T b(V^t) + \sum_{i=1}^{n_c} \lambda_i (C_i)^T (N_i)^T &= 0 \\
\Leftrightarrow V^{t+1} - (A^T A)^{-1} A^T b(V^t) + (A^T A)^{-1} \sum_{i=1}^{n_c} \lambda_i (C_i)^T (N_i)^T &= 0 \\
\Leftrightarrow V^{t+1} = (A^T A)^{-1} A^T b(V^t) - (A^T A)^{-1} \sum_{i=1}^{n_c} \lambda_i (C_i)^T (N_i)^T.
\end{aligned} \tag{3.87}$$

Pour des raisons de lisibilité, nous définissons la matrice  $G = (A^T A)^{-1} A^T$  (matrice pseudo-inverse de  $A$ ) et réécrivons l'équation (3.87) :

$$(3.87) \Leftrightarrow V^{t+1} = Gb(V^t) - (A^T A)^{-1} \sum_{i=1}^{n_c} \lambda_i (C_i)^T (N_i)^T. \tag{3.88}$$

Ayant défini  $V^{t+1}$  en fonction des  $\{\lambda_i\}$ , nous allons à présent pouvoir déterminer et calculer ces derniers. Dans ce but, on substitue le terme  $V^{t+1}$  à la deuxième ligne de l'équation (3.85). Remarquons toutefois que les équations (3.85) et (3.88) font chacune intervenir un indice  $i$ . Nous remplaçons en conséquence  $i$  par  $j$  dans l'équation (3.88) et écrivons :

$$\begin{aligned}
(3.85) \Rightarrow C_j V^{t+1} N_j - P_j = 0, \forall j \in [1, \dots, n_c] \\
\Rightarrow C_j \left( Gb(V^t) - (A^T A)^{-1} \sum_{i=1}^{n_c} \lambda_i (C_i)^T (N_i)^T \right) N_j - P_j = 0, \forall j \in [1, \dots, n_c] \\
\Rightarrow C_j Gb(V^t) N_j - C_j (A^T A)^{-1} \sum_{i=1}^{n_c} \lambda_i (C_i)^T (N_i)^T N_j - P_j = 0, \forall j \in [1, \dots, n_c] \\
\Rightarrow C_j Gb(V^t) N_j - P_j - \sum_{i=1}^{n_c} (C_j (A^T A)^{-1} (C_i)^T (N_i)^T N_j) \lambda_i = 0, \forall j \in [1, \dots, n_c].
\end{aligned} \tag{3.89}$$

Pour simplifier les écritures, on définit la matrice  $L$  dont chaque élément  $\{i, j\}$  est donné par :

$$L_{ij} = C_i (A^T A)^{-1} (C_j)^T (N_j)^T N_i, \quad \forall \{i, j\} \in [1, \dots, n_c] \times [1, \dots, n_c], \tag{3.90}$$

permettant ainsi de réécrire l'équation (3.89) :



$$\begin{aligned}
(3.89) \Leftrightarrow C_j Gb(V^t)N_j - P_j - \sum_{i=1}^{n_c} L_{ji}\lambda_i &= 0, \forall j \in [1, \dots, n_c] \\
\Leftrightarrow \sum_{i=1}^{n_c} L_{ji}\lambda_i &= C_j Gb(V^t)N_j - P_j, \forall j \in [1, \dots, n_c]
\end{aligned} \tag{3.91}$$

soit :

$$(3.91) \Leftrightarrow \begin{cases} \sum_{i=1}^{n_c} L_{1i}\lambda_i = C_1 Gb(V^t)N_1 - P_1 \\ \sum_{i=1}^{n_c} L_{2i}\lambda_i = C_2 Gb(V^t)N_2 - P_2 \\ \dots \\ \sum_{i=1}^{n_c} L_{n_c i}\lambda_i = C_{n_c} Gb(V^t)N_{n_c} - P_{n_c}. \end{cases} \tag{3.92}$$

Si l'on note  $M$  la matrice (de taille  $n_c \times 1$ ) donnée par :

$$M = \begin{pmatrix} C_1 Gb(V^t)N_1 - P_1 \\ C_2 Gb(V^t)N_2 - P_2 \\ \dots \\ C_{n_c} Gb(V^t)N_{n_c} - P_{n_c} \end{pmatrix} \tag{3.93}$$

et  $\Lambda$  le vecteur colonne donné par :

$$\Lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \dots \\ \lambda_{n_c} \end{pmatrix}, \tag{3.94}$$

alors :

$$(3.92) \Leftrightarrow L\Lambda = M. \tag{3.95}$$

On inverse finalement la matrice  $L$  (inversible) pour déterminer par résolution directe les multiplicateurs de *Lagrange*  $\lambda_i$  :

$$\begin{aligned}
(3.95) \quad &\Rightarrow \Lambda = L^{-1}M \\
&\Rightarrow \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \dots \\ \lambda_{n_c} \end{pmatrix} = \begin{pmatrix} L_{11} & L_{12} & \dots & L_{1n_c} \\ L_{21} & L_{22} & \dots & L_{2n_c} \\ \dots & \dots & & \dots \\ L_{n_c1} & L_{n_c2} & \dots & L_{n_cn_c} \end{pmatrix}^{-1} \begin{pmatrix} M_1 \\ M_2 \\ \dots \\ M_{n_c} \end{pmatrix} \\
&\Rightarrow \begin{cases} \lambda_1 = (L^{-1})_{11}M_1 + (L^{-1})_{12}M_2 + \dots + (L^{-1})_{1n_c}M_{n_c} \\ \lambda_2 = (L^{-1})_{21}M_1 + (L^{-1})_{22}M_2 + \dots + (L^{-1})_{2n_c}M_{n_c} \\ \dots \\ \lambda_{n_c} = (L^{-1})_{n_c1}M_1 + (L^{-1})_{n_c2}M_2 + \dots + (L^{-1})_{n_cn_c}M_{n_c} \end{cases},
\end{aligned} \tag{3.96}$$

où  $(L^{-1})_{ij}$  désigne l'élément  $\{i, j\}$  de la matrice  $L^{-1}$ ,  $\forall \{i, j\} \in [1, \dots, n_c] \times [1, \dots, n_c]$ .

Et donc,  $\forall i \in [1, \dots, n_c]$ ,

$$\begin{aligned}
(3.96) \quad &\Rightarrow \lambda_i = \sum_{j=1}^{n_c} (L^{-1})_{ij} M_j \\
&\Rightarrow \lambda_i = \sum_{j=1}^{n_c} \left( (L^{-1})_{ij} \left( C_j G b(V^t) N_j - P_j \right) \right).
\end{aligned} \tag{3.97}$$

Il ne reste alors qu'à substituer les  $\{\lambda_i\}$  calculés dans l'équation (3.87) définissant  $V^{t+1}$ , afin de calculer l'ensemble des positions optimisées honorant les contraintes imposées.

### 3.4 Comparaison et choix d'une méthode d'optimisation

Afin de comparer les deux méthodes améliorant le respect des contraintes de restauration (méthode des pondérations en section 3.3.1 et méthode de respect strict en section 3.3.2), nous étudions sur la figure 3.30 l'exemple de deux modèles composés chacun de 2000 points, après mise à plat de l'horizon. L'image de gauche est le résultat d'une minimisation conjointe pondérée, tandis que l'image de droite est obtenue par la minimisation avec respect strict des contraintes imposées en déterminant des multiplicateurs de *Lagrange*. Les deux résultats sont très similaires, sans différence forte visible. Toutefois, en mesurant la somme des erreurs verticales entre l'horizon et la droite "objectif", sur l'image obtenue grâce à la méthode des pondérations (qui nous le rappelons est une méthode approximant les contraintes imposées), nous obtenons une différence cumulée correspondant à 1.5% de la hauteur de la section présentée (alors que le respect est strict pour la méthode des multiplicateurs de *Lagrange*, impliquant que la somme des erreurs verticales est nulle). En pratique, pour l'application de notre outil de restauration visant à assister la compréhension d'images sismiques lors d'interprétations préliminaires, cette différence est parfaitement acceptable.

Comparons à présent la complexité, c'est-à-dire également le coût typique de ces deux algorithmes

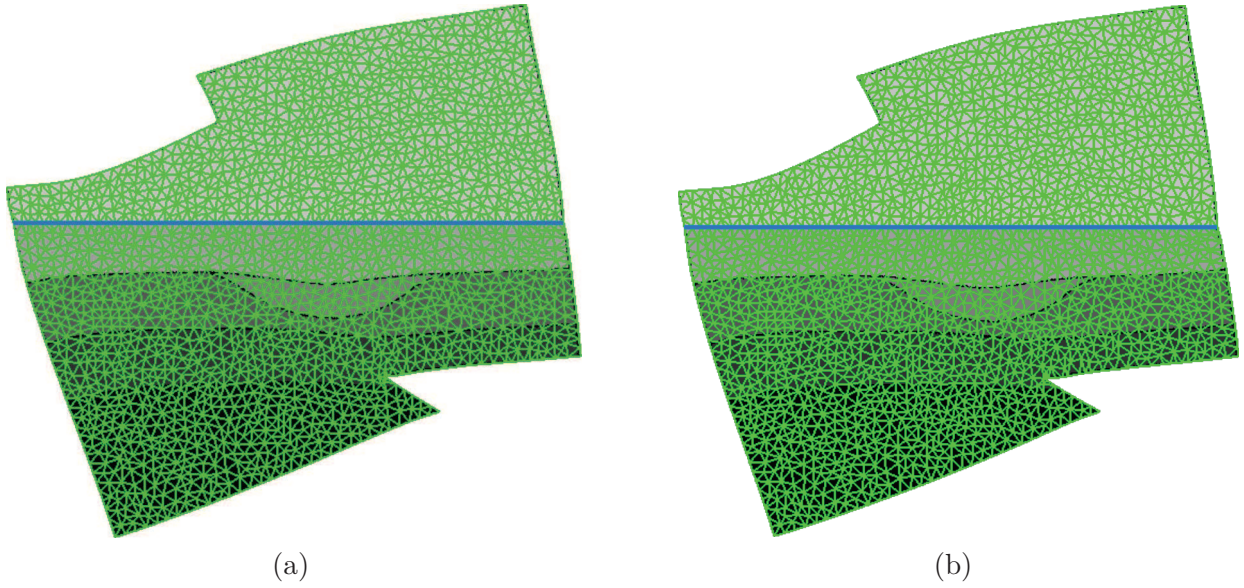


FIGURE 3.30 – **Comparaison entre deux méthodes possibles de remise à plat d’un horizon :** (a) Respect approximatif seulement par une méthode de minimisation conjointe pondérée, (b) respect strict de contraintes par détermination de multiplicateurs de *Lagrange*.

lors d’une itération. Pour l’approche faisant intervenir une minimisation conjointe pondérée, l’essentiel du coût provient de la multiplication des deux termes matriciels  $(A^T A)^{-1} A^T$  et  $b(V)$  issus de l’équation (3.61), sachant que le terme  $(A^T A)^{-1} A^T$  lui peut être pré-calculé lors d’une phase d’initialisation.

Dans l’approche respectant strictement la contrainte, le coût d’une itération est donné par l’équation (3.87) déterminant les positions optimisées. Nous rappelons cette équation pour plus de lisibilité :

$$V^{t+1} = (A^T A)^{-1} A^T b(V^t) - (A^T A)^{-1} \sum_{i=1}^{n_c} \lambda_i (C_i)^T (N_i)^T. \quad (3.98)$$

Nous retrouvons la multiplication des termes matriciels  $(A^T A)^{-1} A^T$  et  $b(V^t)$  qui est légèrement plus rapide que pour le cas de la minimisation pondérée, car les contraintes strictes sont évincées des matrices  $A$  et  $b(V)$  réduisant alors leurs tailles (en pratique sans que ce soit significatif). Puis nous avons la multiplication non négligeable des termes  $(A^T A)^{-1}$  et  $\sum_{i=1}^{n_c} \lambda_i (C_i)^T (N_i)^T$ , qui réintroduit les contraintes strictes et dont le coût correspond d’après nos mesures, au gain réalisé lors de la multiplication des deux premiers termes. Nous négligeons enfin la multiplications des vecteurs  $(C_i)^T$  et  $(N_i)^T$ , car leurs tailles sont trop petites pour être représentatives. À présent, dans l’équation (3.87), nous devons également déterminer le vecteur  $\Lambda$ , donné par l’équation (3.96) que nous rappelons :  $\Lambda = L^{-1} M$ . Si la matrice  $M$  est assez directe à déterminer (il s’agit d’un vecteur colonne, où chaque élément est défini par des multiplications de vecteurs, avec le terme  $(A^T A)^{-1} A^T b(V^t)$  à-priori déjà connu), il faut en revanche déterminer la matrice  $L$ , puis l’inverser. Cette matrice carrée contient autant de lignes que de contraintes strictes, c’est-à-dire de points définissant un horizon ou une faille et en pratique, cette matrice à généralement une taille d’ordre  $100 \times 100$ . Si l’on se base sur cette valeur, son inversion n’est pas négligeable et il faut de plus la populer, c’est-à-dire déterminer pour 10000 éléments  $\{i, j\}$  :

$L_{ij} = C_i(A^T A)^{-1}(C_j)^T(N_j)^T N_i$  et bien que les termes  $C_i$  et  $N_i$  soient vectoriels, cette opération n'est pas négligeable.

Ceci nous permet d'établir que la méthode respectant strictement les contraintes de restauration est plus complexe et donc plus coûteuse, ce à quoi on pouvait s'attendre intuitivement car un respect strict paraît plus coûteux à imposer qu'un respect approché. Afin de quantifier cela avec précision, nous avons mesuré les temps de calcul de ces deux méthodes dans le cas de la restauration des images de la figure 3.30. Le modèle initial est volontairement composé d'un nombre élevé de points (2000 points) de manière à rendre le problème compliqué et réaliste et surtout faire que les comparaisons soient fiables et peu sensibles à des problèmes d'*overhead* ou de petites initialisations complémentaires non pris en compte dans notre calcul théorique de coût ci-dessus. Dans le cas de la méthode par minimisation pondérée nous obtenons des temps d'exécution de 12 secondes pour l'initialisation puis de 7 secondes de convergence. Pour la méthode respectant strictement les contraintes de restauration, nous obtenons 11 secondes pour l'initialisation et 21 secondes pour l'étape de convergence.

On peut remarquer que le temps d'initialisation pour la méthode de respect strict est inférieur au temps d'initialisation pour la méthode approximante. En effet pour ces deux méthodes, cette étape pré-calcule  $(A^T A)^{-1}A^T$  et la matrice  $A$  est un peu plus petite pour la méthode avec respect strict, comme nous l'avons souligné plus haut. Enfin, sans surprise, le coût par itération est nettement plus élevé pour la méthode respectant strictement les contraintes imposées.

En conclusion, nous avons montré que les deux méthodes conduisent en pratique à des résultats très proches, mais que la technique de respect strict des contraintes impose un surcoût significatif. De ce fait, il semble raisonnable de conclure que la méthode de respect strict ne devrait être choisie par l'utilisateur que si celui-ci a besoin d'une précision très forte sur le résultat final obtenu. Dans le cas de notre application l'aspect temps quasi-réel est plus important, aussi nous retenons la méthode de minimisation pondérée.

### 3.5 Pré-conditionnement du problème de restauration

Les algorithmes d'optimisation d'une fonction coût non-linéaire  $f$  à  $n$ -variables, tels que la méthode de *Gauss-Newton* utilisée dans ce travail, permettent de déterminer itérativement le minimum de  $f$  en fixant pour chaque itération une condition initiale (c'est-à-dire en affectant un ensemble de valeurs aux différentes variables) linéarisant ainsi  $f$  et autorisant une résolution directe déterminant l'ensemble des variables optimisées. De tels algorithmes convergent alors vers la solution optimale la plus proche de la condition initiale et il peut arriver que l'optimisation des différentes variables ne soit pas la solution attendue, car elle correspond à un minimum local de la fonction coût  $f$  et non pas au minimum global recherché. On illustre ce point sur la figure 3.31, où si nous linéarisons la fonction coût à la position  $x = x_0$ , alors nous déterminerons après convergence la position  $x_n$  qui n'est pas optimale, car correspond à un minimum local de  $f$ .

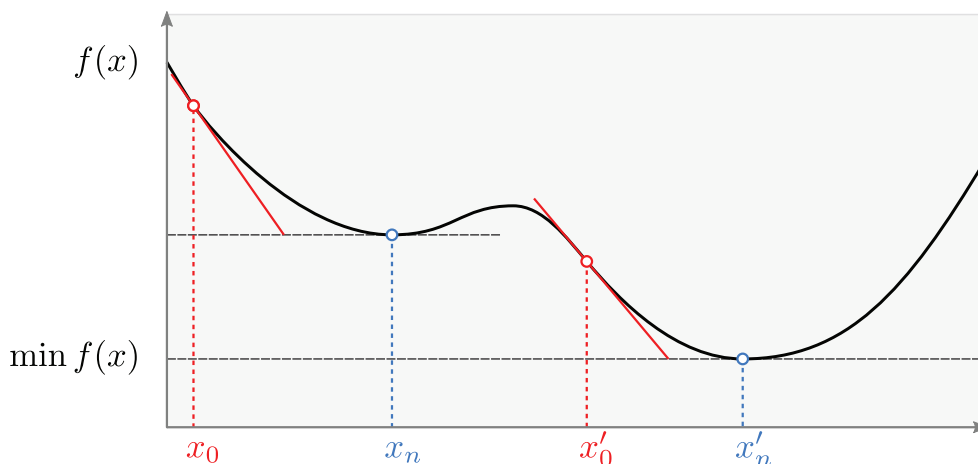


FIGURE 3.31 – Illustration du fonctionnement de l’algorithme de *Gauss-Newton* permettant de déterminer itérativement le minimum  $f(x_n)$  d’une fonction coût  $f(x)$ , en fixant comme condition initiale  $x = x_0$ .

Dans notre application, nous avons pu rencontrer ce phénomène de convergence vers un minimum local lors de l’opération de restauration d’une faille. Le résultat obtenu s’avère dans ce cas faux et inacceptable. Nous allons donc dans un premier temps décrire comment il est théoriquement possible d’éviter un minimum local par un pré-conditionnement de la minimisation, nous détaillerons ensuite le problème rencontré lors de l’opération de restauration d’une faille et expliquerons en quoi la minimisation est mal posée, enfin nous appliquerons la théorie en proposant un pré-conditionnement à l’opération de restauration d’une faille.

### 3.5.1 Pré-conditionnement d’un algorithme de minimisation

Comme nous l’avons souligné en introduction, un algorithme d’optimisation échoue car les valeurs affectées aux variables à optimiser sont trop éloignées de celles recherchées et que l’algorithme rencontre durant sa convergence un minimum local dans lequel il se piège. En d’autres termes, on peut considérer que l’estimation initiale est mauvaise, précisément car les valeurs fixées (la condition initiale) sont trop éloignées.

Pour éviter ce phénomène, une solution consiste à pré-conditionner la minimisation, c’est-à-dire à choisir une condition initiale proche du minimum escompté, telle que la convergence ne rencontre aucun minimum local. On illustre cette approche sur la figure 3.31, où fixer  $x = x'_0$  se révèle un meilleur choix que  $x = x_0$  car on converge comme espéré vers la position optimale  $x'_n$ , puisque aucun minimum local de  $f$  ne perturbe alors la convergence.

### 3.5.2 Observation de convergence vers des minima locaux lors de la restauration de failles

Tel que nous l’avons mentionné en introduction de cette section, nous avons rencontré des minima locaux avec l’opération de restauration d’une faille qui se traduit par un problème d’inversion de maille (repliement d’une maille sur elle-même conduisant à un *Jacobien* devenant localement négatif

et donc à un maillage non valide). Nous illustrons un tel cas pathologique sur le modèle de la figure 3.32.a, pour lequel le résultat obtenu par application de notre algorithme est présenté sur la figure 3.32.b. Par une simple première analyse visuelle un utilisateur notera immédiatement que le résultat obtenu paraît problématique. Bien que les contraintes de restauration que nous souhaitions imposer, c'est-à-dire la continuité de l'horizon et le contact à de faille soient en effet respectées par notre algorithme, le glissement du bloc de droite le long de la faille s'avère faible, alors qu'on pouvait penser intuitivement obtenir une amplitude de déplacement plus importante. De plus, on remarque visuellement une déformation locale forte de l'épaisseur de la deuxième couche en partant du haut, qui conduit à une discontinuité nette de la courbure de l'horizon remis en cohérence.

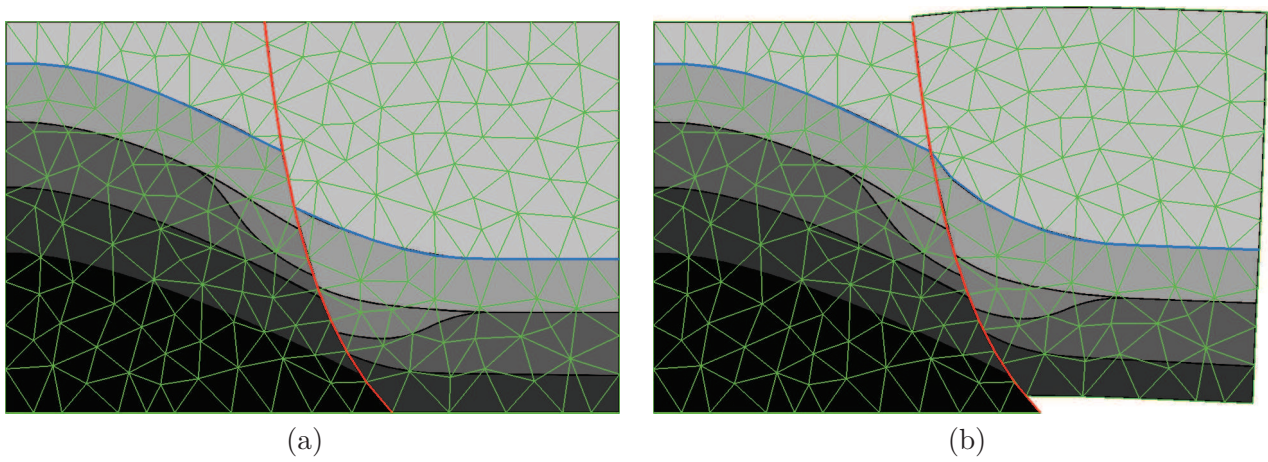


FIGURE 3.32 – **Illustration de collision, c'est-à-dire d'inversion de maille donnant un *Jacobien* localement négatif et donc un maillage invalide, pour un exemple pathologique mais possible de restauration de faille :** (a) modèle initial, (b) modèle déformé (restauré) mais donnant un maillage invalide car contenant des *Jacobiens* négatifs.

Pour analyser plus précisément ce mauvais résultat obtenu, nous montrons sur la figure 3.33 un grossissement de la partie concernée, la figure 3.33.a montrant le modèle avant restauration. Nous avons associé une couleur unique à certaines facettes intéressantes afin d'observer leur comportement plus aisément. Les points situés le long de la faille sont également mis en évidence par une alternance de couleurs franches blanches et noires. Sur la figure 3.33.b nous montrons la même zone après restauration. Premièrement, si nous regardons les points situés à droite de la faille, nous nous apercevons que l'alternance de points noirs et blancs n'est plus respectée. Deuxièmement, désormais des facettes se chevauchent ; en particulier, la facette bleu foncé s'est déplacée sous la facette rouge et la facette violette. On remarque également le chevauchement des facettes verte et violette. Ces observations illustrent plus clairement le caractère non valide du maillage obtenu.

Maintenant que nous avons analysé visuellement le caractère non valide du maillage obtenu après restauration, en détectant les chevauchements indésirables, cherchons à présent à en comprendre plus précisément la cause. Dans l'exemple de la restauration de la faille présenté sur la figure 3.32, intéressons-nous aux contraintes de restauration que nous avons exprimées. La figure 3.34 rappelle les contraintes de restauration (bleu pour la continuité de l'horizon, rouge pour le contact à la faille)



que nous avons appliquées.

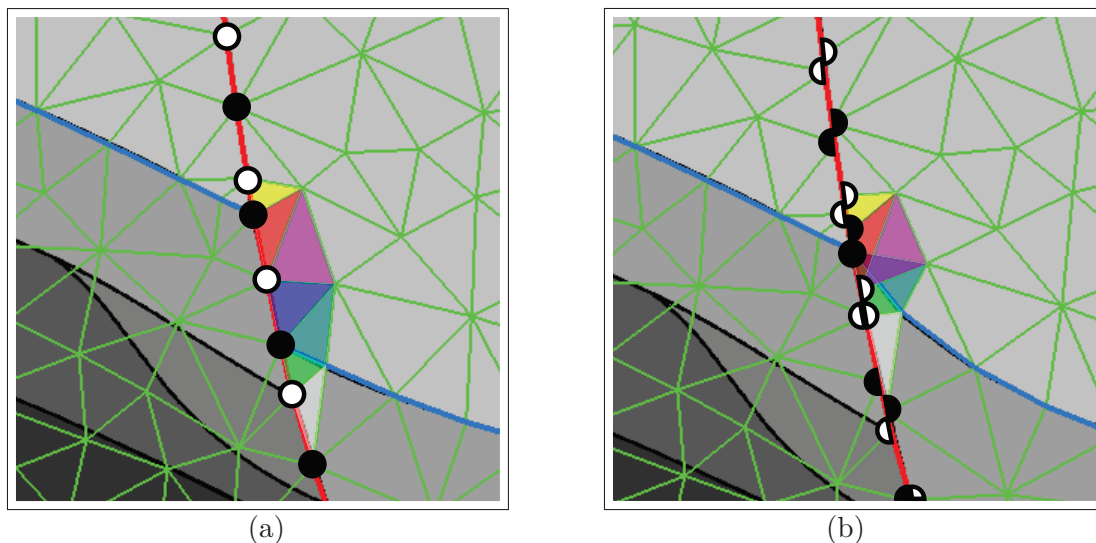


FIGURE 3.33 – Grossissement du phénomène de collision, c'est-à-dire d'inversion de maille donnant un *Jacobien* localement négatif et donc un maillage invalide : (a) modèle initial, (b) modèle déformé (restauré), on voit que le maillage est devenu non valide car des facettes se chevauchent désormais.

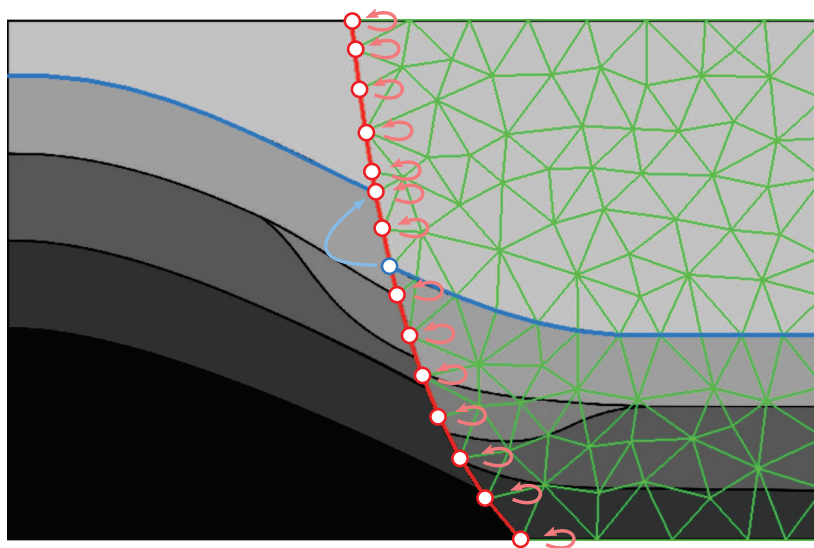


FIGURE 3.34 – Illustration des contraintes de restauration que nous avons appliquées lors de la restauration de la faille.

Comme nous l'avons exposé dans la section 3.2.2.2, la contrainte de contact à la faille calcule les projetés de tous les points présents sur cette dernière afin d'y imposer le déplacement correspondant. Or, dans l'état initial de l'image non restaurée, ces projetés sont les positions propres (initiales) des points concernés, c'est-à-dire que bien entendu le déplacement initial est nul. En d'autres termes, lors de la première itération, la contrainte de respect à la faille a pour action d'imposer l'immobilité aux points concernés. On explique ainsi le faible glissement le long de la faille observé sur la figure 3.32.b.

Le problème apparaît alors du fait que les contraintes de restauration sont dans ce cas contradictoires car un déplacement conséquent est imposé au point bleu afin de retrouver la continuité de l'horizon tandis qu'on impose l'immobilité aux points rouges proches. La solution obtenue correspond donc à un minimum local de la fonction d'optimisation, ce qui explique la distorsion autour de l'intersection entre la faille et l'horizon ainsi que les phénomènes de recouvrements de facettes obtenus.

### 3.5.3 Pré-conditionnement de l'algorithme d'optimisation pour la restauration de la faille

Ayant décrit plus haut que, lors de la restauration d'une faille, le processus de minimisation des contraintes pouvait échouer en restant piégé dans un minimum local puis identifié la cause, en décrivant le comportement problématique introduit par la contrainte de contact à la faille lors de la première itération, nous proposons une approche heuristique évitant ce problème en améliorant la condition initiale, c'est-à-dire en modifiant les positions des points  $\{v_i\}$  de manière à se rapprocher des positions optimisées intuitivement attendues.

Nous modifions donc le maillage initial en effectuant la translation du bloc mobile qui respecte la contrainte de continuité de l'horizon, comme illustré sur la figure 3.35. Cette figure présente également les nouveaux vecteurs déplacements imposés par les contraintes de restauration. On peut alors noter visuellement le fait que ces déplacements sont désormais complémentaires et non contradictoires. En respectant dès le début la contrainte de continuité de l'horizon, nous nous assurons ainsi du fait que les calculs des projetés à la faille seront de bonnes estimations. L'équilibre final sera alors apporté par l'introduction des contraintes géométriques dans une deuxième étape.

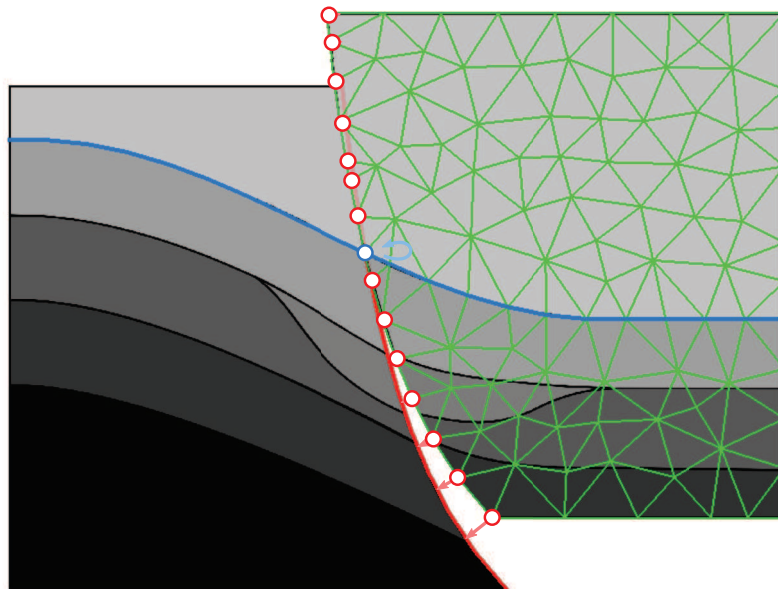


FIGURE 3.35 – Illustration du pré-conditionnement de l'opération de restauration de la faille, par la translation du bloc à déformer, qui honore la contrainte de continuité de l'horizon.



## 3.6 Convergence et critère d'arrêt du processus itératif d'optimisation

Dans la section 3.2, nous avons traduit les contraintes de restauration ainsi que les contraintes géométriques par un ensemble d'équations, qui n'étant pas linéaires n'ont pu être résolues par une approche numérique directe et ont nécessité l'emploi d'une méthode d'optimisation itérative de type *Gauss-Newton* décrite en section 3.3. Cependant, toute méthode itérative requiert un critère (ou un ensemble de critères) d'arrêt de l'algorithme. En conséquence, nous discutons et détaillons dans cette section les critères d'arrêts mis en œuvre par nos travaux.

Les contraintes intervenant dans le processus de restauration sont décrites par des quantités vectorielles que chaque itération de l'algorithme optimise (minimise). Ces quantités vectorielles qualifient individuellement le respect de chaque contrainte. En d'autres termes, une quantité vectorielle nulle (vecteur nul) indique un respect parfait de la contrainte considérée, alors qu'une quantité vectorielle importante (vecteur dont on estime la valeur de sa norme élevée) suggère que la contrainte imposée n'est pas honorée. L'étude de ces quantités vectorielles durant l'exécution de l'algorithme se révèle alors être un indicateur intéressant pour étudier puis qualifier sa convergence. En ce but, nous allons suivre l'évolution de ces quantités vectorielles dans le scénario typique de restauration d'une image sismique incluant la restauration d'une faille et la mise à plat d'un horizon. Nous montrerons alors qu'il est possible de détecter la convergence de l'algorithme et donc de définir un critère d'arrêt de l'algorithme itératif en détectant une évolution devenant quasi-nulle des critères étudiés.

### 3.6.1 Étude de la convergence pour le scénario typique de restauration d'une faille

Commençons par étudier le cas de la restauration de la faille qui est présente dans le modèle synthétique de la figure 3.1.a et suivons l'évolution de plusieurs critères pendant un nombre fixe d'itérations. Les courbes de variation obtenues sont présentées sur les figures 3.36, 3.37 et 3.38 où nous avons fixé le nombre d'itérations total à 400 et visualisons les 200 premières significatives pour suivre l'évolution des différentes courbes.

Les trois graphiques présentés permettent de comparer trois niveaux de résolution différents du maillage généré pour décrire le modèle synthétique. Le graphique de la figure 3.36 correspond à un maillage contenant 152 points et 254 triangles ; pour le graphique de la figure 3.37, le modèle contient 504 points et 808 triangles ; et enfin, pour le graphique de la figure 3.38 le maillage contient 705 points et 1293 triangles.

Un premier critère qui nous donne des indications sur la convergence de l'algorithme (courbe rouge) est celui qui mesure la quantité de déformation se produisant entre deux itérations et qui est calculé en mesurant la somme cumulée des normes des vecteurs déplacement appliqués aux points de contraintes. Sur les graphiques, l'échelle verticale utilisée pour ce premier critère est située sur l'axe de droite, de type logarithmique. Deux autres critères permettent de visualiser le respect des

contraintes géométriques : l'erreur à la contrainte de forme (en bleu) et l'erreur à la contrainte de longueur d'arête (en violet). Le critère d'erreur à la contrainte de forme mesure la somme des erreurs vectorielles  $J_S$  définie dans l'équation (3.33) et le critère d'erreur à la contrainte de longueur d'arête mesure la somme des erreurs vectorielles  $J_L$  définie dans l'équation (3.49). Pour ces critères, l'échelle verticale utilisée dans les graphiques est située sur l'axe de gauche et est logarithmique.

Les deux derniers critères qui nous permettent de visualiser le respect des contraintes de restauration sont l'erreur à la contrainte de continuité de l'horizon (en vert) et l'erreur à la contrainte de contact à la faille (en orange). Le critère d'erreur à la continuité de l'horizon mesure la somme des erreurs vectorielles  $J_{F1}$  définie dans l'équation (3.7) et le critère d'erreur à la contrainte de contact à la faille mesure la somme des erreurs vectorielles  $J_{F2}$  définie dans l'équation (3.10). Pour ces critères, l'échelle verticale utilisée dans les graphiques est située sur l'axe de gauche et est également logarithmique.

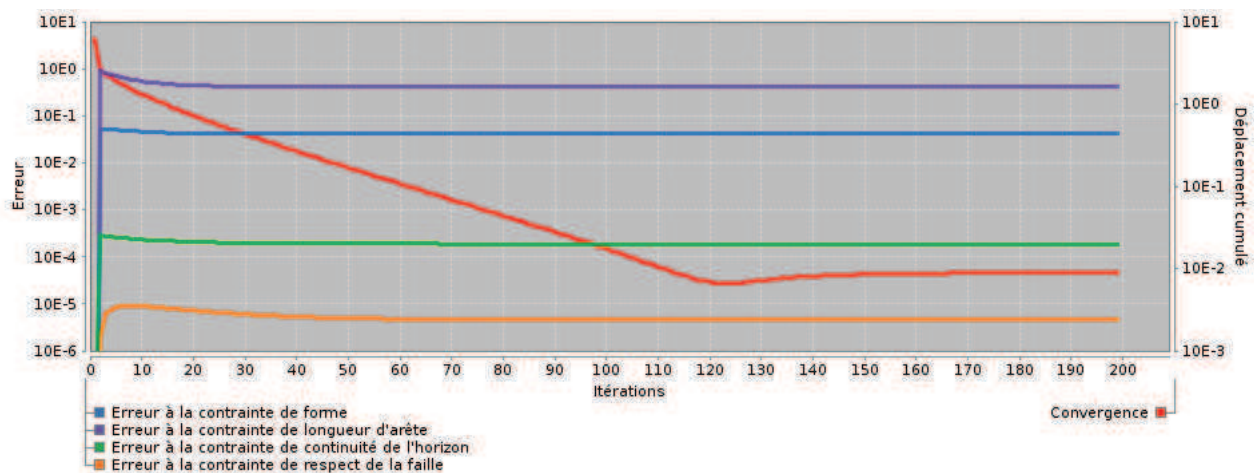


FIGURE 3.36 – Illustration de la convergence au cours des itérations pour le cas de la restauration de la faille de la figure 3.1.a dans le cas d'un maillage contenant 152 points et 254 triangles.

Sur cette première série de figures nous observons tout d'abord que les courbes mesurant le critère de convergence (en rouge) ont un profil fortement décroissant sur les toutes premières itérations et qu'ensuite la pente s'atténue mais reste non nulle. Puis, à partir de la 120<sup>ème</sup> ou 140<sup>ème</sup> itération selon les cas, on remarque que l'amplitude du déplacement cumulé devient quasiment constante et que la pente de la courbe devient quasiment nulle, ce qui donne un plateau horizontal.

Nous pouvons également remarquer que les courbes mesurant les critères d'erreurs aux contraintes géométriques (en bleu et violet) et aux contraintes de restauration (en orange et vert) ont des caractéristiques proches : elles sont fortement croissantes sur la 1<sup>ère</sup> itération, puis deviennent décroissantes, ce qui traduit une diminution des erreurs liées aux contraintes. La pente s'atténue jusqu'à devenir quasi-nulle et former un plateau horizontal, les erreurs mesurées devenant alors constantes et n'évoluant plus.

La formation du plateau horizontal sur les courbes d'erreurs (bleue, violette, orange et verte) se

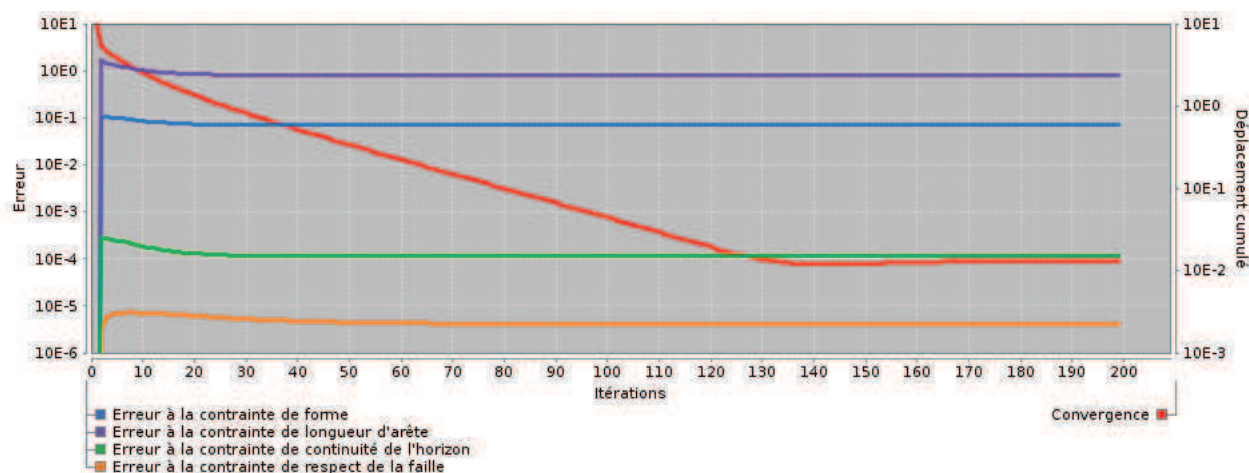


FIGURE 3.37 – Illustration de la convergence au cours des itérations pour le cas de la restauration de la faille de la figure 3.1.a dans le cas d’un maillage contenant 504 points et 808 triangles.

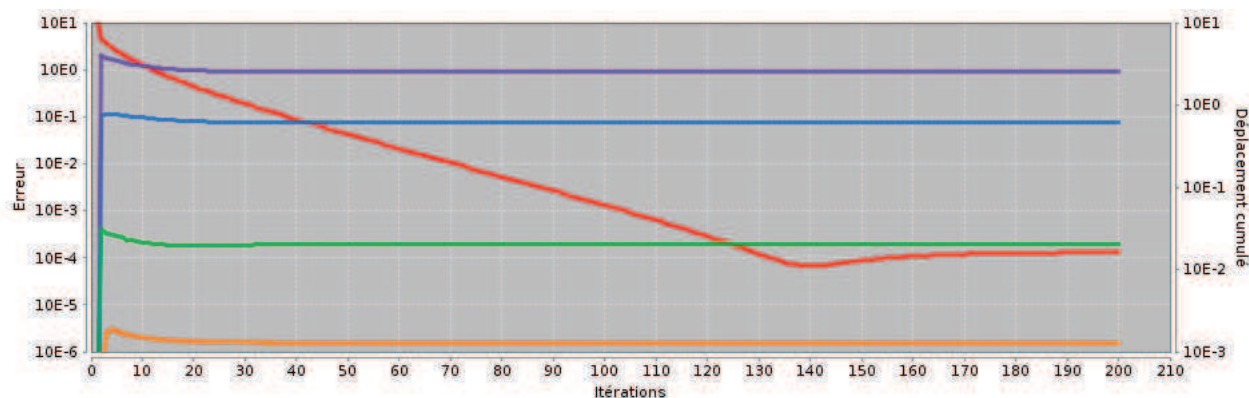


FIGURE 3.38 – Illustration de la convergence au cours des itérations pour le cas de la restauration de la faille de la figure 3.1.a dans le cas d’un maillage contenant 705 points et 1293 triangles.

produit entre la 10<sup>ème</sup> et la 60<sup>ème</sup> itération selon les cas et autour de la 130<sup>ème</sup> itération pour la courbe de convergence en rouge. On explique cette différence par le fait que la courbe de convergence ne peut se stabiliser, qu’une fois les pentes des courbes d’erreur devenues quasi-nulles. En effet, si la déformation mesurée à chaque itération reste constante, cela indique que le système a convergé vers un minimum et que toutes les contraintes sont honorées au mieux. On peut également souligner qu’étant donné les faibles pentes initiales des courbes d’erreur et leurs aspects logarithmiques, il est difficile de décrire le moment exact de la formation d’un plateau. Toutefois, au moins une de ces courbes conserve un légère pente jusqu’à stabilisation de la courbe de convergence.

En comparant les trois graphiques, on peut enfin noter que la stabilisation de la courbe de convergence (rouge) est plus tardive lorsque le maillage est plus dense : elle intervient avant la 130<sup>ème</sup> itération sur la figure 3.36 et après la 130<sup>ème</sup> sur la figure 3.38.

### 3.6.2 Étude de la convergence pour le scénario de mise à plat de l'horizon

Étudions à présent la mise à plat de l'horizon pour ce même scénario (modèle synthétique de la figure 3.4). Comme précédemment, nous suivons l'évolution de plusieurs critères pour un nombre d'itérations fixé à 400. Les graphiques obtenus sont présents sur les figures 3.39, 3.40 et 3.41.

Nous comparons également sur ces trois graphiques, trois niveaux de résolution différents du maillage. Le graphique de la figure 3.39 correspond à un maillage contenant 300 points et 522 triangles, celui de la figure 3.40 à un maillage contenant 963 points et 1789 triangles, et celui de la figure 3.41 à un maillage contenant 1361 points et 2546 triangles.

Les critères de convergence (en rouge) et de respect des contraintes géométriques, c'est-à-dire de respect des formes en bleu et de respect des longueurs d'arête en violet, restent pertinents et sont donc de nouveau présentés.

Pour analyser le respect de la contrainte de restauration, nous introduisons enfin un nouveau critère qui exprime l'erreur à la contrainte de mise à plat de l'horizon (en rose). Ce critère mesure la somme des erreurs vectorielles  $J_H$  définie dans l'équation (3.14). Comme pour les critères de respect des contraintes géométriques, son échelle verticale se lit sur l'axe de gauche dans les graphiques et est logarithmique.

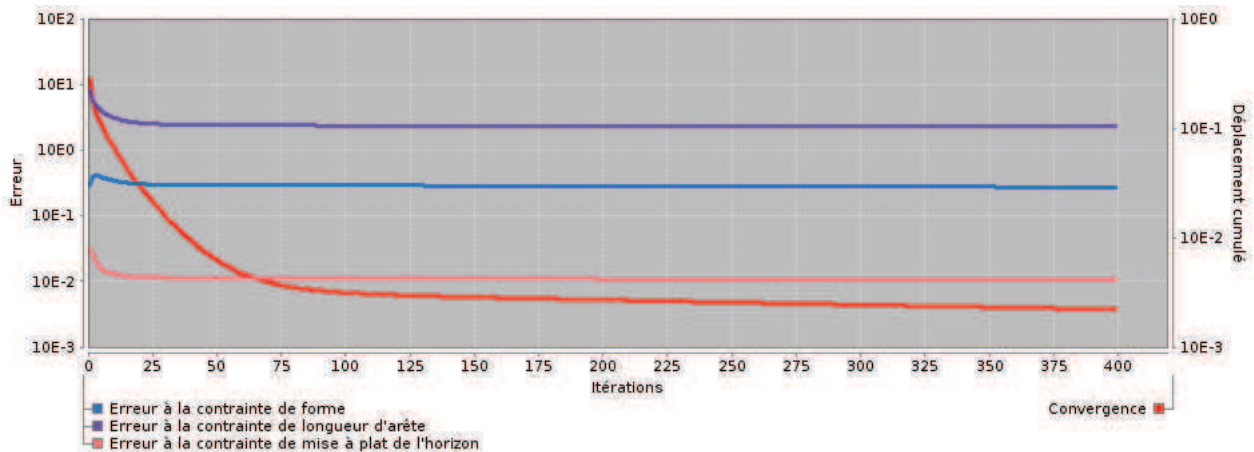


FIGURE 3.39 – Illustration de la convergence au cours des itérations pour le cas de la mise à plat de l'horizon de la figure 3.4 dans le cas d'un maillage contenant 300 points et 522 triangles.

Dans cette seconde série de graphiques, nous confirmons les observations faites ci-dessus pour le scénario de restauration de la faille. Tout d'abord, les courbes mesurant le critère de convergence (en rouge) ont un profil identique fortement décroissant aux premières itérations, puis s'atténuant au cours des itérations suivantes. On remarque par ailleurs que pour ce scénario précis, les pentes ne sont pas encore tout à fait nulles autour de 400 itérations, ce qui indique que les systèmes continuent de converger lentement.

Les courbes mesurant les critères d'erreur aux contraintes géométriques (en bleu et violet) et à la

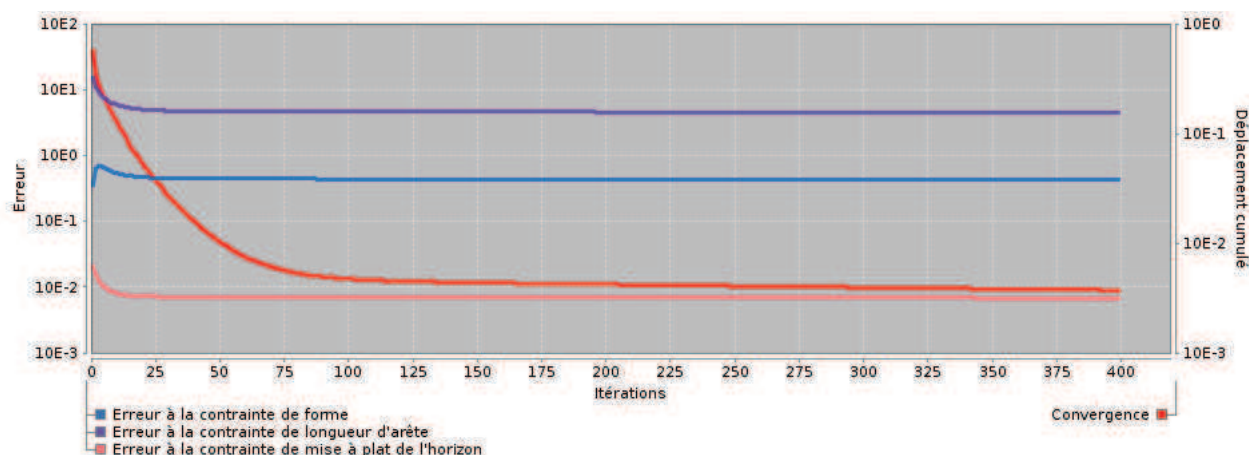


FIGURE 3.40 – Illustration de la convergence au cours des itérations pour le cas de la mise à plat de l’horizon de la figure 3.4 dans le cas d’un maillage contenant 963 points et 1789 triangles.

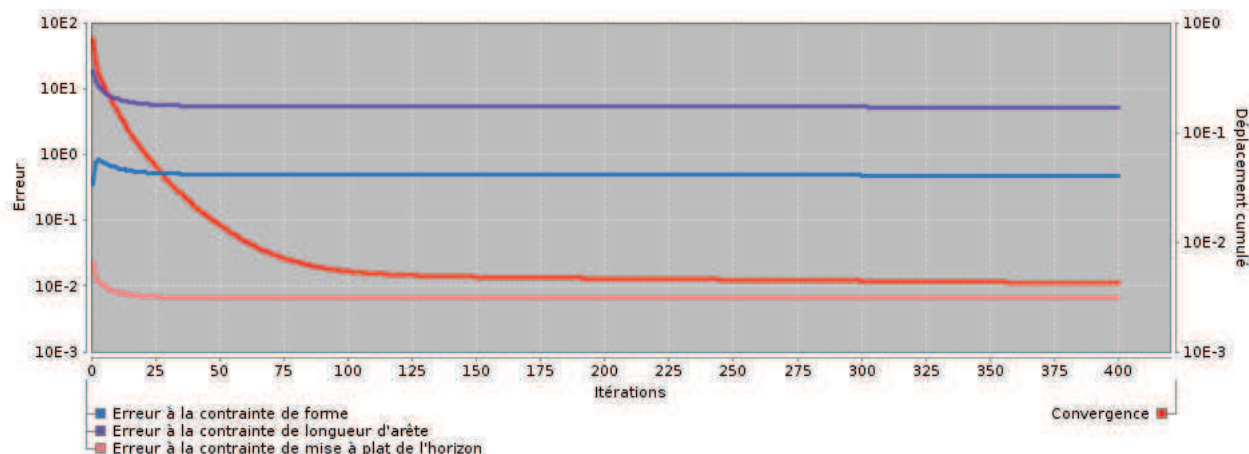


FIGURE 3.41 – Illustration de la convergence au cours des itérations pour le cas de la mise à plat de l’horizon de la figure 3.4 dans le cas d’un maillage contenant 1361 points et 2546 triangles.

contrainte de restauration (en rose) sont encore une fois fortement croissantes sur la 1<sup>ère</sup> itération, puis décroissent. La pente initialement forte se réduit au fur et à mesure des itérations et tend à la formation d’un plateau horizontal. Comme pour le cas de la courbe de convergence en rouge, on remarque que pour ces courbes d’erreur la pente n’est pas encore quasiment nulle autour de 400 itérations, ce qui indique à nouveau que les systèmes continuent de converger lentement.

On remarque également dans ce scénario que la formation du plateau horizontal sur les courbes d’erreurs (bleue, violette et rose) semble se produire avant l’observation de la formation de ce plateau sur la courbe de convergence (en rouge). Comme pour le cas de la restauration de la faille, ce phénomène indique qu’en réalité, la pente d’au moins une des courbes d’erreur n’est pas quasi-nulle.

Enfin, comme dans le scénario de restauration de la faille ci-dessus, la stabilisation de la courbe



de convergence se produit plus tardivement si le maillage est plus dense.

### 3.6.3 Analyse plus précise de ces résultats

Le premier point important est que dans chaque exemple le système converge. En effet, toutes les courbes observées ont un profil décroissant. Plus précisément, cette convergence est rapide puis ralentit, la pente des courbes est forte sur les premières itérations puis s'atténue. Concernant les courbes d'erreur, l'augmentation observée lors de la première itération correspond à l'action des contraintes de restauration qui perturbent initialement fortement le système.

Au fur et à mesure des itérations, les pentes des courbes se réduisent. On observe dans un premier temps la formation d'un plateau horizontal sur les courbes de respect des contraintes, puis finalement sur la courbe de convergence. La pente de celle-ci ne devient nulle que lorsque les pentes des courbes d'erreurs aux contraintes sont toutes quasiment nulles. La formation de ces plateaux permet d'identifier deux phases : une durant laquelle les critères étudiés décroissent, ce qui indique que le système continue à converger et une phase où ces critères n'évoluent plus, ce qui indique que le système a atteint un état d'équilibre et qu'il n'est donc pas nécessaire de continuer à itérer.

Cette transition peut être nette, comme dans le cas du scénario de la restauration de la faille en figure 3.1.a, ou bien plus graduelle, comme dans le cas du scénario de la mise à plat de l'horizon en figure 3.4, pour lequel on ne détecte pas encore clairement de plateaux même vers la 400<sup>ème</sup> itération. Dans ce deuxième cas, on se trouve dans une situation de convergence lente, et dans la deuxième partie de la courbe il faudra un grand nombre d'itérations pour une évolution du système numériquement peu significative et visuellement imperceptible, ce qui permet de dire que dans un tel cas il n'est pas très intéressant de continuer à itérer.

### 3.6.4 Conditions d'arrêt automatique

Après avoir analysé ci-dessus la convergence de notre algorithme pour les cas d'un scénario de restauration d'une faille et d'un scénario de mise à plat d'un horizon, nous pouvons maintenant proposer trois conditions d'arrêt automatique de l'algorithme qui nous semblent pertinentes.

Dans notre analyse, la phase d'équilibre du système se détecte par la formation d'un plateau horizontal sur la courbe de convergence. Nous considérons donc que l'identification de ce phénomène est une bonne condition d'arrêt de l'algorithme car nous avons vu que dans ce cas les itérations suivantes n'améliorent pas significativement la convergence et sont donc superflues.

Nous avons observé que certaines contraintes géométriques ou contraintes de restauration ont une convergence lente dans notre cas de restauration d'images sismiques. Ceci est clair en particulier pour le cas du scénario de mise à plat de l'horizon en figure 3.4, pour lequel toutes les courbes d'erreurs ne se sont pas totalement stabilisées même après un grand nombre d'itérations, comme cela peut être observé sur les figures 3.39, 3.40 et 3.41. On confirme ce fait en observant que la pente de la courbe de convergence est alors légèrement plus élevée que celle des courbes d'erreur : en effet, elle décrit le cumul des légères déformations se produisant à ce stade pour stabiliser toutes les courbes d'erreur.

Cependant, comme les valeurs relevées sur les courbes d'erreur sont faibles, l'espérance de voir une évolution significative de la déformation est également faible. Nous proposons donc une condition d'arrêt optimisée qui consiste à détecter quand l'espérance d'évolution de la déformation devient faible de la sorte. On peut considérer que lorsque les pentes des courbes d'erreur aux contraintes sont toutes très faibles et malgré une pente légèrement supérieure de la courbe de convergence, les itérations à venir produiront peu d'évolution du système. Notre condition améliorée stoppe donc l'algorithme lorsque les pentes de toutes les courbes d'erreurs (excluant précisément la courbe de convergence) sont inférieures à un seuil choisi.

Enfin, l'évolution de la convergence dépend du modèle en entrée (horizons et failles), qui peut varier assez fortement suivant les cas étudiés par les géologues. Il est alors nécessaire d'introduire une troisième condition d'arrêt pour prévenir d'éventuels cas qui ne convergeraient pas, c'est-à-dire qui ne pourraient pas être restaurés par nos techniques. Une solution simple et efficace consiste à fixer un nombre maximal d'itérations et considérer le modèle comme impossible à restaurer par nos algorithmes si le nombre maximal d'itérations est atteint.

### 3.6.5 Étude de la convergence pour le scénario complet de restauration, avec critère d'arrêt automatique

Pour illustrer le fonctionnement de ces conditions d'arrêt, nous présentons le scénario de la restauration complète (faille puis horizon) du modèle synthétique de la figure 3.1.a. Les figures 3.39, 3.40 et 3.41 montrent de manière détaillée l'évolution des critères introduits précédemment.

Dans ce scénario, le nombre d'itérations est dynamique et dépend des conditions d'arrêt automatique que nous venons d'introduire pour notre algorithme. Nous avons implémenté la condition d'arrêt par détection de pentes faibles sur les courbes d'erreur à chaque contrainte. Nous comparons à nouveau le cas de trois niveaux de résolution différents du maillage. Le graphique de la figure 3.42 correspond à un maillage contenant 300 points et 522 triangles, celui de la figure 3.43 correspond à un maillage contenant 963 points et 1789 triangles, et celui de la figure 3.44 correspond à un maillage contenant 1361 points et 2546 triangles.

Nous retrouvons les courbes de convergence (en rouge), de respect des contraintes géométriques (contrainte de forme en bleu, et contrainte de longueur d'arête en violet) et de respect des contraintes de restauration (contrainte de continuité de l'horizon en vert, de contact à la faille en orange, et de mise à plat de l'horizon en rose). Étant donné que ce scénario traite le cas de la restauration complète, nous présentons sur la partie gauche des graphiques le cas de la restauration de la faille et sur la partie droite le cas de la mise à plat de l'horizon, ainsi que les contraintes de restauration correspondantes.

Dans les figures présentées, l'arrêt automatique de la restauration de la faille se déclenche autour de la 40<sup>ème</sup> itération. Nous remarquons alors que toutes les courbes d'erreur aux contraintes ont une pente très faible, conformément au comportement attendu. On observe alors sur les graphiques un décrochement indiquant le passage à l'opération de mise à plat de l'horizon. Cette opération se termine selon les cas entre la 70<sup>ème</sup> et la 80<sup>ème</sup> itération, une fois que les pentes des courbes aux nouvelles contraintes deviennent inférieures au seuil choisi.

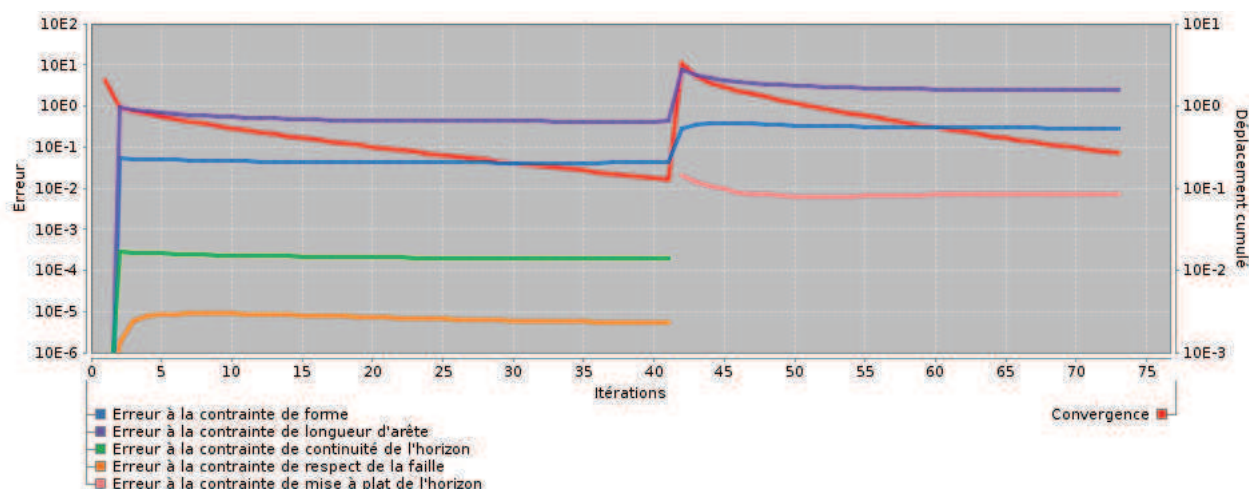


FIGURE 3.42 – Illustration de la convergence au cours des itérations pour le processus complet de restauration dans le cas d’un maillage contenant 300 points et 522 triangles.

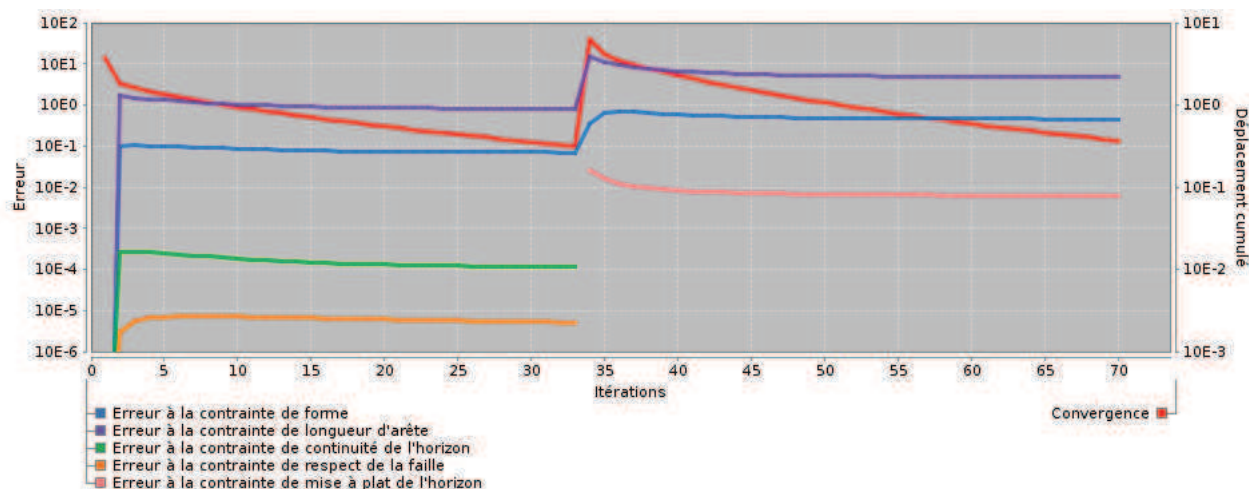


FIGURE 3.43 – Illustration de la convergence au cours des itérations pour le processus complet de restauration dans le cas d’un maillage contenant 963 points et 1789 triangles.

Ceci nous permet de conclure que les conditions d’arrêt automatique que nous avons proposées permettent d’obtenir une bonne convergence en détectant des pentes très faibles pour toutes les contraintes d’erreur. Cette méthode permet de garantir un nombre d’itérations aussi faible que possible pour une qualité de restauration souhaitée donnée en arrêtant l’algorithme lorsque l’espérance de gain de convergence supplémentaire devient trop faible pour être intéressant, en particulier par rapport à l’augmentation du coût de calcul que cela impliquerait.



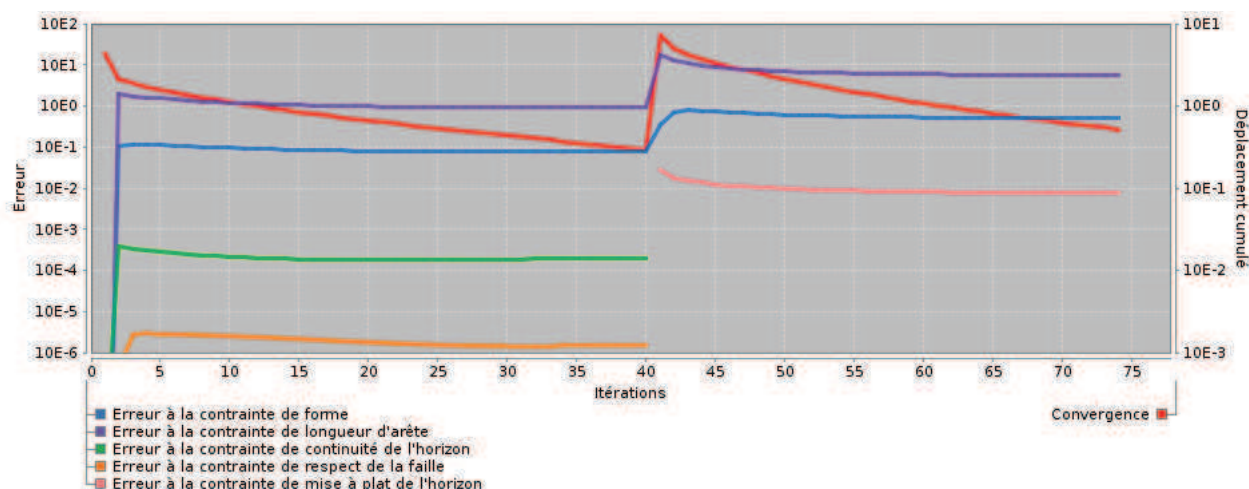


FIGURE 3.44 – Illustration de la convergence au cours des itérations pour le processus complet de restauration dans le cas d’un maillage contenant 1361 points et 2546 triangles.

### 3.7 Impact du maillage sur la déformation et choix d’un maillage optimisé

Étant donné une image sismique que l’on déforme par un processus de restauration, nous avons pu constater expérimentalement que le choix du maillage ainsi que de sa résolution peut impacter significativement la déformation finale obtenue. Nous présentons un exemple illustrant ce fait dans la section 3.7.1.

L’influence du maillage sur la déformation obtenue pose donc la question de comment choisir un bon maillage pour la restauration d’une image donnée. En particulier, il faut s’assurer que le processus de calcul de la déformation ne soit pas rendu imprécis ou biaisé par un mauvais choix de maillage, c’est-à-dire par l’utilisation d’un maillage qui n’aurait pas une qualité ou une résolution suffisante. Il est donc important de comprendre dans un premier temps la cause de l’influence du maillage sur la déformation obtenue, puis de préciser quels sont les types de maillages qui sont acceptables et permettent le calcul de déformations que nous estimons correctes, c’est-à-dire qu’un géologue utilisateur jugerait acceptables et réalistes du point de vue géologique. Nous analyserons donc la cause de cette influence dans la section 3.7.2.

Nous verrons alors que les maillages acceptables sont généralement non optimaux et nous illustrerons cet aspect par un exemple dans la section 3.7.3. Nous expliquerons également que dans une application classique, l’utilisation d’un maillage optimal est préjudiciable car il implique un nombre de points de contraintes trop important. Ce nombre complexifie le processus de calcul de la déformation et rend son calcul soit exagérément coûteux, soit impossible.

Nous proposerons alors dans la section 3.7.4 une méthode permettant de déterminer un maillage acceptable, c’est-à-dire produisant une déformation que nous estimons correcte, et optimisé d’un point de vue coût de calcul car possédant le moins de points de contraintes possible. Nous illustrerons à nouveau cette méthode par l’étude d’un cas concret de maillage d’un modèle à restaurer.

### 3.7.1 Cas d'application avec deux maillages impactant différemment la déformation

Montrons à présent que le choix du maillage peut impacter significativement la déformation finale obtenue. Si nous considérons l'image sismique simplifiée (de taille  $1024 \times 653$ ) sur la figure 3.45.a contenant une faille et un horizon, nous proposons de comparer la restauration de cette image avec deux maillages réguliers, de densité différente, que nous notons pour cet exemple  $\mathcal{M}_A$  (en figure 3.45.b) et  $\mathcal{M}_B$  (en figure 3.45.c), tel que  $\mathcal{M}_B$  soit plus dense que  $\mathcal{M}_A$ . Les images restaurées sont ensuite présentées sur les figures 3.45.d et 3.45.e.

Ces deux images restaurées semblent en apparence similaires, toutefois, une étude approfondie relève des différences notables. Nous proposons donc une méthode permettant de mesurer la différence de déformation entre les images restaurées des figures 3.45.d et 3.45.e.

Soit une image  $\mathcal{I}$  ( $w$  colonnes  $\times$   $h$  lignes) que l'on souhaite déformer par interpolation de la déformation d'un maillage  $\mathcal{M}_i$ , on génère l'image  $\mathcal{C}_i$  ( $w$  colonnes  $\times$   $h$  lignes) représentant la carte des déplacements appliqués à  $\mathcal{I}$ , telle que la couleur de chaque pixel de  $\mathcal{C}_{i,x,y}$  exprime la norme du vecteur déplacement  $\vec{d}_{i,x,y}$  appliqué au pixel  $\mathcal{I}_{x,y}$ . À titre d'exemple, nous présentons en figure 3.46.a la carte des déplacements associée à la restauration de l'image en figure 3.45.a, par déformation du maillage  $\mathcal{M}_A$  généré en figure 3.45.b. Sur cette image, la norme représente un déplacement exprimé en pixels. Une couleur chaude exprime un déplacement important, tandis qu'une couleur froide exprime un déplacement faible ou nul, tel que le montre l'axe horizontal de l'histogramme.

À présent, si l'on note  $\mathcal{C}_A$  la carte des déplacements correspondant à la déformation du maillage  $\mathcal{M}_A$  et  $\mathcal{C}_B$  la carte des déplacements correspondant à la déformation du maillage  $\mathcal{M}_B$ , il est possible de générer une carte de différence qui associe à chaque pixel la différence (absolue) des normes issues de deux cartes de déplacement. De manière générale, on définit la valeur à chaque pixel d'une carte de différence  $\mathcal{D}_{ij,x,y}$ , pour deux maillages  $\mathcal{M}_i$  et  $\mathcal{M}_j$ , par :

$$\mathcal{D}_{ij,x,y} = \sqrt{\left(\|\vec{d}_{i,x,y}\| - \|\vec{d}_{j,x,y}\|\right)^2}. \quad (3.99)$$

On génère par cette méthode la carte  $\mathcal{D}_{AB}$  qui illustre (en figure 3.47) la différence de déformation entre les deux images restaurées des figures 3.47.d et 3.47.e. Les couleurs froides indiquent localement une forte similitude du résultat de la restauration, tandis que les couleurs chaudes pointent des zones de différence. Comme on le voit de cette image, deux zones rouges sont observables, indiquant une différence de résultat, dont l'ordre est d'après l'histogramme supérieur à 6 pixels.

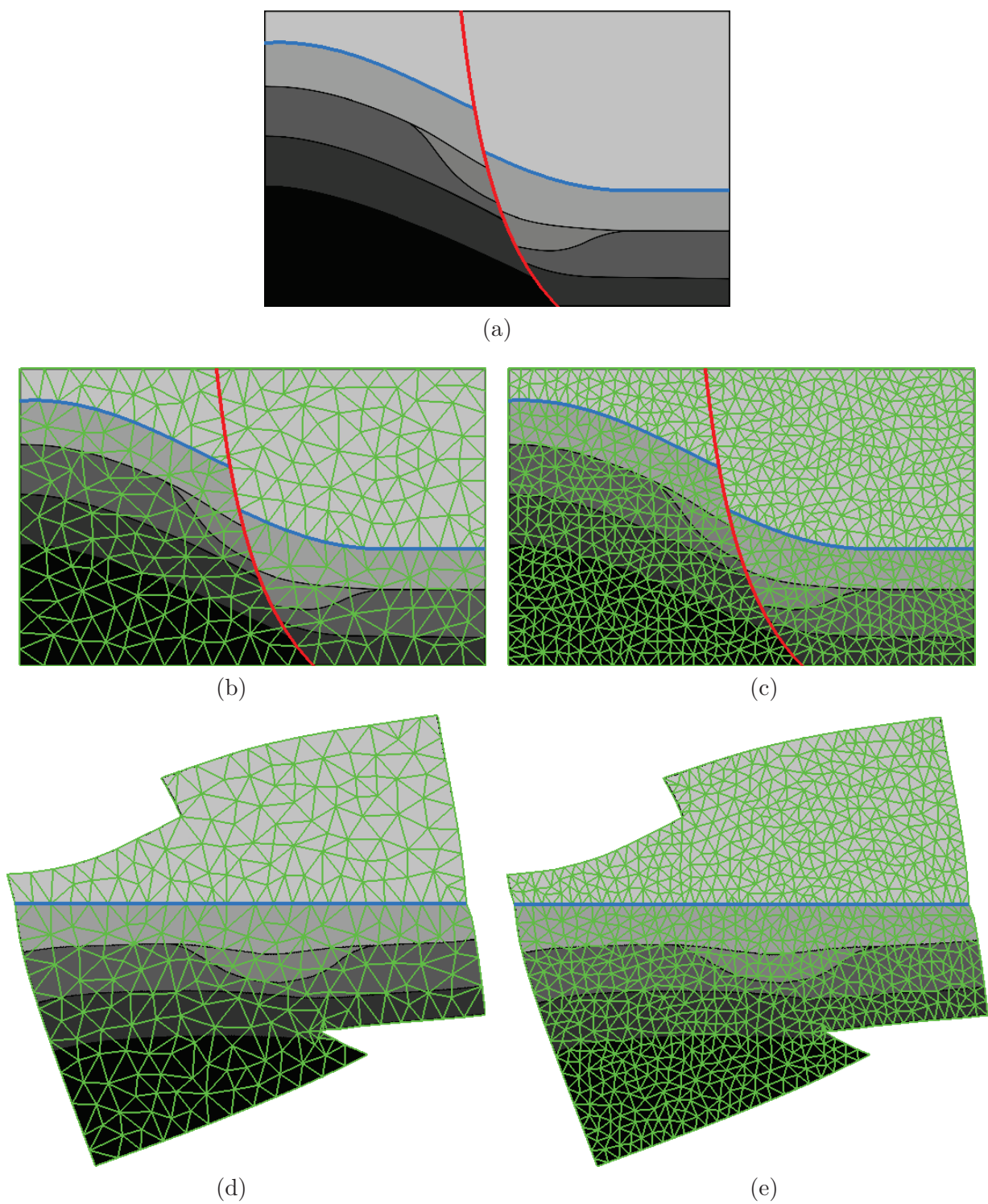


FIGURE 3.45 – **Exemple illustrant le processus de restauration avec différents choix de maillage :** (a) Image sismique initiale ( $1024 \times 653$ ), (b) image sismique échantillonnée par un maillage régulier  $\mathcal{M}_A$ , (c) image sismique échantillonnée par un maillage régulier  $\mathcal{M}_B$  plus dense que  $\mathcal{M}_A$ , (d) restauration de l'image sismique échantillonnée par le maillage  $\mathcal{M}_A$ , (e) restauration de l'image sismique échantillonnée par le maillage  $\mathcal{M}_B$ .

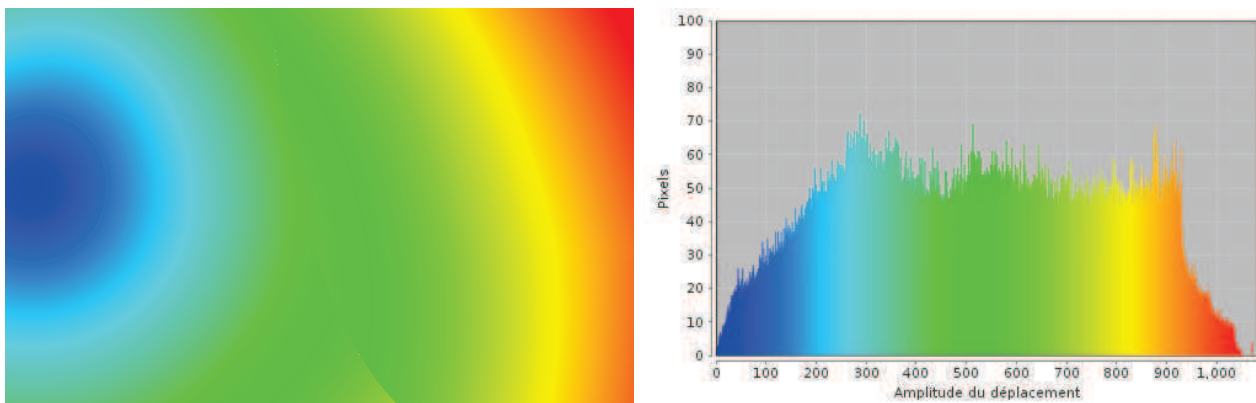


FIGURE 3.46 – Carte des déplacements  $C_1$  lors de la restauration de l’image sismique simplifiée en figure 3.45.a, par déformation du maillage  $M_A$  illustré en figure 3.45.b.

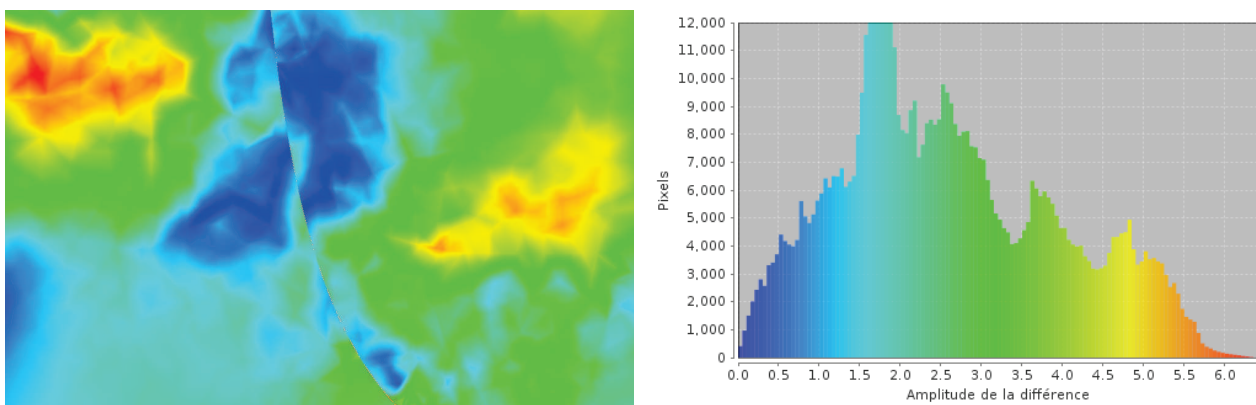


FIGURE 3.47 – Carte de différence  $D_{AB}$  entre les cartes de déplacement  $C_A$  et  $C_B$  illustrant la différence de déformation entre les deux images restaurées des figures 3.47.d et 3.47.e. Les zones rouges mettent en évidence que ces deux images présentent des différences locales supérieures à 6 pixels.

### 3.7.2 Causes de l’impact du choix du maillage sur la déformation

Les différences mises en évidence par l’exemple du paragraphe précédent s’expliquent par le mécanisme d’interpolation utilisé. Nous calculons en effet la déformation seulement aux points de contraintes dont le nombre dépend de la densité du maillage et varie dans les différents exemples présentés. Sur les pixels de l’image, le vecteur déplacement exprimant la déformation n’est pas calculé mais interpolé par une interpolation barycentrique de type linéaire (précisément par la méthode dite de “*texture mapping*”). En conséquence ce vecteur est imprécis car il résulte d’un processus d’interpolation.

Le niveau de précision obtenu est bon lorsque il y a une faible variation des vecteurs déplacement à interpoler, comme on le montre sur la figure 3.48.a illustrant l’interpolation linéaire (en bleu) des vecteurs déplacements d’un triangle (en rouge) sur une image matricielle. On y observe une faible variation des vecteurs interpolés, car les vecteurs rouges ont des orientations et des normes relativement proches. Ce n’est en revanche plus vrai sur la figure 3.48.b qui introduit un nouveau point de contrainte et son vecteur déplacement dont l’orientation est plus marquée. En conséquence, le champ de vecteurs

bleu interpolé est relativement hétérogène d'un pixel à l'autre et l'on peut légitimement douter de sa précision, en particulier dans la zone centrale de l'image. Remarquons par exemple que tous les vecteurs bleus sont orientés vers le haut, or le vecteur rouge au centre de l'image est lui orienté vers le bas. Contrairement à l'interpolation, le calcul de la déformation (donc des vecteurs déplacement rouges) n'est pas linéaire et l'on pourrait penser que l'interpolation n'a pas su saisir cette subtilité locale de la déformation. Pour se convaincre que l'orientation des vecteurs obtenus au centre de l'image est juste, on aimerait alors rajouter de nouveaux points de contraintes au centre et ainsi améliorer la précision de l'interpolation.

En général, dans les cas d'applications typiques de restauration sismique, l'utilisation de gros triangles tend à accentuer ce phénomène d'imprécision des vecteurs à interpoler.

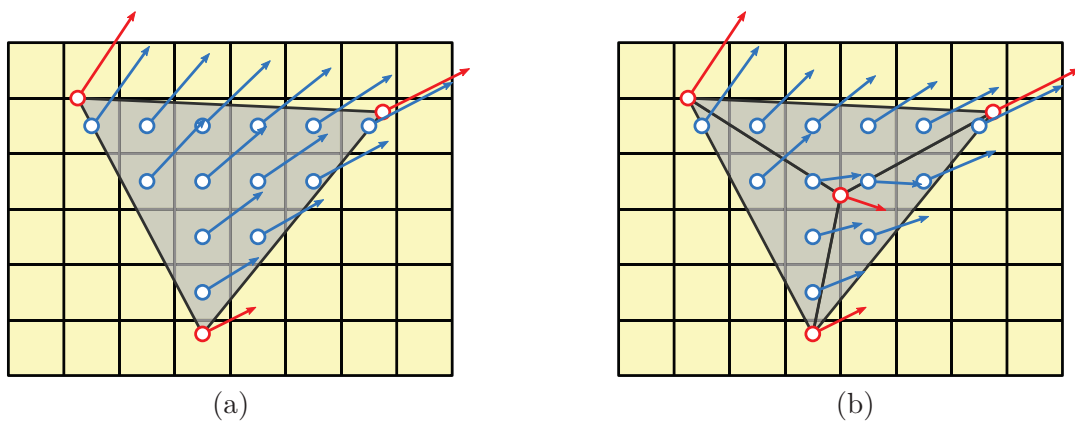


FIGURE 3.48 – **Exemples d'interpolation linéaire (vecteurs bleus) de vecteurs déplacements (rouges) sur une image matricielle.** (a) L'interpolation est bonne car il y a peu de différences d'orientation ou de longueur des vecteurs à interpoler. (b) Le vecteur rouge à interpoler au centre présente une différence marquée avec les autres vecteurs et l'on ne sait pas si le champ de vecteur interpolé au centre de l'image décrit correctement la déformation. On aimerait introduire de nouveaux vecteurs à interpoler pour s'en convaincre.

Puisque l'utilisation d'une technique d'interpolation explique les variations observées, on pourrait choisir de ne pas interpoler et donc de prendre le maillage strictement défini par les pixels de l'image. On appelle alors un tel maillage "optimal", car la déformation serait calculée en chacun de ses nœuds. Cependant, nous allons montrer qu'un tel maillage ne peut en pratique être utilisé.

### 3.7.3 Limite d'utilisation d'un maillage optimal

Avec un maillage "optimal", c'est-à-dire dont tous les pixels sont des points de contraintes, si nous prenons un cas d'application typique simple, comme par exemple une petite image de taille  $1000 \times 1000$ , nous aurons à manipuler des matrices contenant de l'ordre de  $10^{14}$  éléments, ce qui est un nombre remarquablement élevé. Dans ce cas, selon les implémentations, soit on échouera à calculer la déformation (car la manipulation de ces éléments causera des erreurs d'allocation mémoire), soit le temps nécessaire pour la calculer sera très élevé et inacceptable en pratique pour un utilisateur qui



souhaite pouvoir faire des interprétations en temps quasi-réel. Aussi, au sens de ce critère supplémentaire de faisabilité ou de durée réaliste du calcul, on se demande si un nombre de points aussi élevé est optimisé.

L'intérêt d'interpoler est précisément d'optimiser le processus de déformation en réduisant le nombre de points de contraintes et en s'affranchissant de cette limite d'utilisation d'un maillage optimal. Ainsi, nous pensons qu'interpoler reste une bonne approche à condition de choisir un maillage suffisamment dense pour éviter les problèmes d'imprécisions, liés à l'interpolation, décrits plus haut. Toute la difficulté réside alors dans le choix de cette densité minimale.

De plus, selon les différentes images sismiques à restaurer, la déformation calculée est différente et en conséquence la densité minimale requise est variable. De gros triangles peuvent suffire si la déformation est faible et/ou linéaire, mais de petits triangles seront nécessaires si la déformation est plus complexe.

Dans le prochain paragraphe, nous allons proposer une méthode permettant pour un scénario de restauration donné, de déterminer une densité de maillage suffisante et déterminerons finalement un maillage "optimisé" (c'est-à-dire avec la densité requise pour produire une interpolation de qualité suffisante, mais minimale pour limiter le nombre de points de contraintes et garantir la rapidité du calcul de la déformation) au scénario de restauration.

### 3.7.4 Méthode permettant de déterminer un maillage "optimisé" acceptable

En vue de déterminer un maillage "optimisé", nous proposons d'étudier pour le scénario de restauration de l'image sismique en figure 3.45.a, l'impact du raffinement du maillage sur la déformation finale. Nous présentons un processus itératif qui part d'un maillage initial (grossier), le raffine et compare les deux restaurations obtenues. Nous montrerons alors qu'au fur et à mesure des raffinements, ces restaurations deviennent quasi-identiques et considérons alors que la densité du maillage est suffisante pour produire une interpolation de qualité et une déformation de l'image sismique équivalente à celle que l'on obtiendrait avec le maillage "optimal". Cette méthode permet de choisir plusieurs maillages candidats (qui présentent alors des restaurations quasi-identiques) et nous choisirons finalement celui qui possède le moins de points de contraintes.

#### 3.7.4.1 Analyse de l'impact du raffinement

Pour analyser l'impact du raffinement d'un maillage sur la déformation obtenue, nous partons d'un maillage grossier  $\mathcal{M}_1$  associé à une carte des déplacements  $\mathcal{C}_1$ , puis un nouveau maillage  $\mathcal{M}_2$  plus dense est généré et sa carte des déplacements  $\mathcal{C}_2$  est calculée. Nous visualisons alors l'image  $\mathcal{D}_{12}$  exprimant la différence entre les deux déformations obtenues, qui illustre en particulier l'amplitude de cette différence (dont la distance se mesure en pixels) pour chaque point de l'image  $\mathcal{I}$ . Il est alors intéressant de visualiser l'histogramme de  $\mathcal{D}_{12}$  qui exprime la distribution de l'amplitude. En d'autres termes, l'histogramme nous permet directement d'évaluer si deux déformations sont proches ou non, en vérifiant que la majorité des pixels de  $\mathcal{D}_{12}$  se situent dans une zone de faible amplitude.

Afin d'illustrer ce procédé, nous choisissons un maillage initial régulier  $\mathcal{M}_1$  avec 152 points et un maillage raffiné  $\mathcal{M}_2$  également régulier, avec 318 points. La carte de différence  $\mathcal{D}_{12}$  et son histogramme sont présentés en figure 3.49. Pour permettre la comparaison avec les différents histogrammes qui seront présentés, nous avons choisi une échelle fixe. Dans certains cas toutefois, il est possible que des valeurs se situent au delà de l'échelle choisie. Nous le précisons alors et indiquerons les valeurs numériques intéressantes.

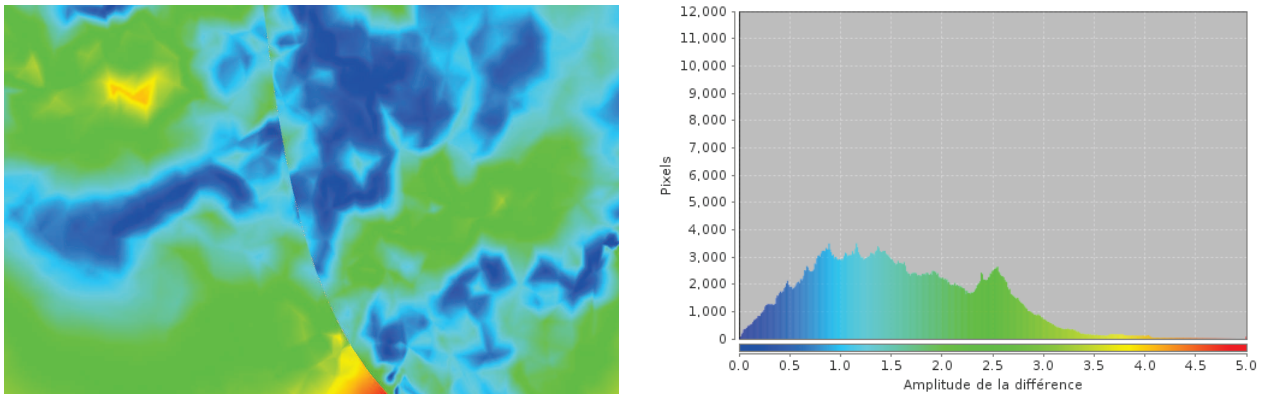


FIGURE 3.49 – Carte de différence  $\mathcal{D}_{12}$  et son histogramme.

Nous observons sur la carte de différence que la déformation obtenue après raffinement est différente sur de nombreuses zones, en particulier sur celles peintes en vert, jaune et rouge. Cette tendance se confirme sur l'histogramme, où pour 27.7% des points de l'image, l'amplitude de la différence est inférieure à 1 pixel (différence infime ou invisible) et pour 69%, l'amplitude est inférieure à 2 pixels (différence infime). Pour 30.4% des pixels, l'amplitude de la différence est donc significative, atteignant des valeurs allant jusqu'à 4.8 pixels.

La différence entre les résultats donnés par déformation des maillages  $\mathcal{M}_1$  et  $\mathcal{M}_2$  étant trop importante, nous considérons que  $\mathcal{M}_1$  n'est pas un bon maillage.

Nous raffinons à présent  $\mathcal{M}_2$  (318 points) et générons le maillage régulier  $\mathcal{M}_3$  avec 504 points. La carte de différence  $\mathcal{D}_{23}$  et son histogramme sont présentés en figure 3.50.

Sur la carte de différence obtenue, nous observons des similarités avec celle de la figure 3.49 : en effet, de nombreuses zones indiquent une amplitude significative de la différence. On remarque par ailleurs que la teinte prédominante est le vert, alors qu'elle était plutôt bleue sur la figure 3.49, indiquant ainsi une amplitude de différence moyenne plus forte. L'histogramme confirme cette tendance : 62.2% des points de l'image ont une amplitude supérieure à 2 pixels et l'amplitude maximum atteint 4.6 pixels. En revanche, seulement 11.5% des points de l'image ont une amplitude de différence invisible et inférieure à un pixel.

L'analyse de cette image nous permet donc de considérer que  $\mathcal{M}_2$  n'est pas un bon maillage, car il produit des résultats trop différents de la restauration effectuée avec  $\mathcal{M}_3$ .

Poursuivons donc notre processus de raffinement et générons un maillage  $\mathcal{M}_4$  plus dense (705

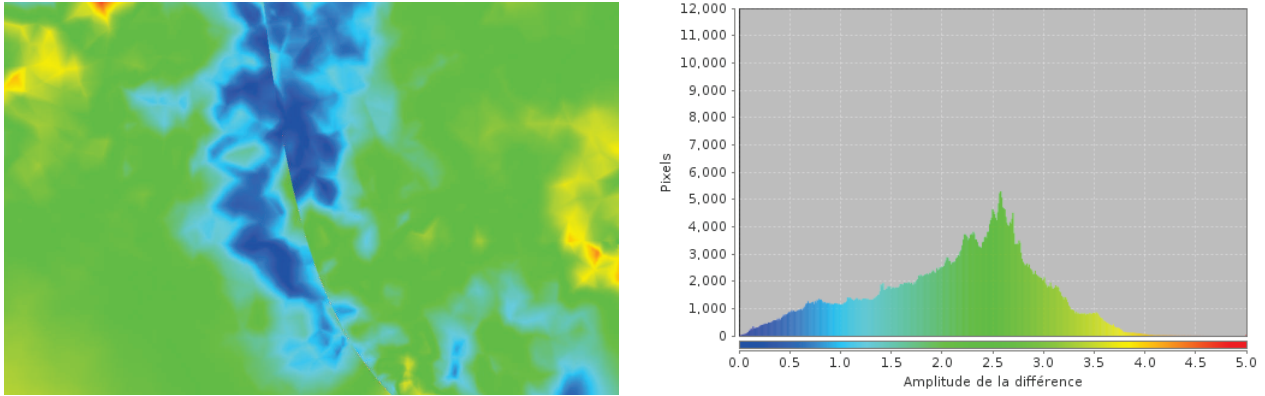


FIGURE 3.50 – Carte de différence  $\mathcal{D}_{23}$  et son histogramme.

points), que l'on va comparer avec le maillage  $\mathcal{M}_3$  (504 points). La carte de différence  $\mathcal{D}_{34}$  et son histogramme sont présentés en figure 3.51.

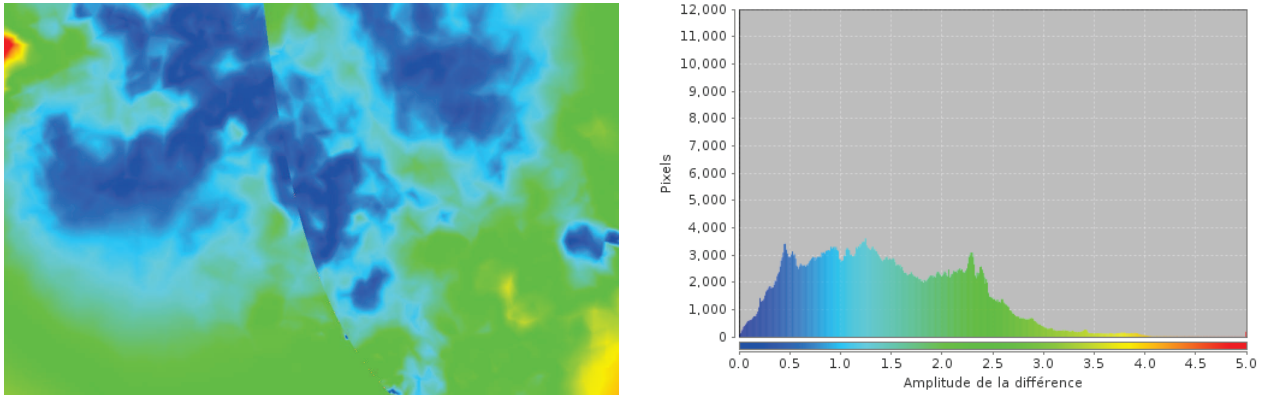


FIGURE 3.51 – Carte de différence  $\mathcal{D}_{34}$  et son histogramme.

La carte de différence obtenue est intéressante, car on observe sur le centre de l'image une prédominance bleue, témoignant dans cette zone d'une similarité avec la déformation obtenue par interpolation de  $\mathcal{M}_3$ , alors que les amplitudes de différences plus fortes se localisent plutôt sur les bords de l'image. Cette tendance se remarque également (mais en proportion moindre) sur la carte de différence précédente (en figure 3.50). Cela indique qu'au fil des raffinements, la déformation change peu au centre de l'image et l'on peut à ce stade penser que les raffinements de maillages successifs permettent de converger vers une déformation (donc une restauration) stable. Cette stabilité se propage alors du centre de l'image vers l'extérieur.

L'histogramme indique que 33.2% des points de l'image ont une amplitude de différence inférieure à un pixel et 73.5% des points inférieure à 2 pixels. Bien qu'insatisfaisantes, ces mesures sont jusqu'ici les meilleures observées. Nous notons que l'amplitude de différence maximum est de 6 pixels (l'échelle horizontale de l'histogramme ne s'arrêtant qu'à 5 pixels, nous avons tronqué ces valeurs et l'histogramme présente alors un pic rouge sur la partie droite).

L'analyse de cette figure indique donc qu'une stabilisation de la déformation semble s'amorcer,



mais la différence encore significative ne permet pas de considérer  $\mathcal{M}_3$  comme un bon maillage.

Nous poursuivons le processus, générons le maillage raffiné  $\mathcal{M}_5$  (876 points) et comparons les résultats obtenus par déformations des maillages  $\mathcal{M}_4$  (705 points) et  $\mathcal{M}_5$ . La carte de différence  $\mathcal{D}_{45}$  et son histogramme sont présentés en figure 3.52.

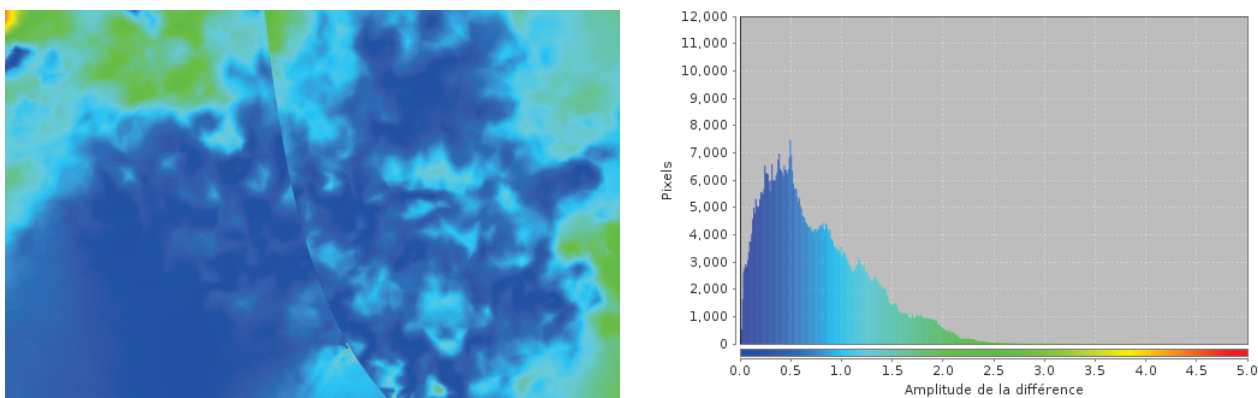


FIGURE 3.52 – Carte de différence  $\mathcal{D}_{45}$  et son histogramme.

Sur la carte de déformation obtenue, la première observation immédiate est la prédominance de bleu, indiquant une forte similarité des déformations par interpolation des maillages  $\mathcal{M}_4$  et  $\mathcal{M}_5$ . L'histogramme confirme cette similarité, car la majorité des pixels se trouvent sur sa partie gauche (où l'amplitude de la différence est faible). Précisément, il indique que 71% des points de l'image ont une différence de déplacements inférieure au pixel et 98% ont une différence inférieure à 2 pixels. Le raffinement de  $\mathcal{M}_4$  en  $\mathcal{M}_5$ , n'a donc réellement profité qu'aux 2% de points dont l'amplitude maximum de la différence est 4.3 pixels. On remarque d'ailleurs que les amplitudes de différence les plus fortes se trouvent comme précédemment sur les bords de l'image, confirmant l'hypothèse avancée d'une stabilisation de la déformation se propageant du centre de l'image, vers l'extérieur.

À la lecture de ces résultats, on peut considérer que  $\mathcal{M}_4$  est un bon maillage car les restaurations obtenues en déformant  $\mathcal{M}_4$  et  $\mathcal{M}_5$  sont similaires pour 98% des points.  $\mathcal{M}_4$  est alors une version optimisée de  $\mathcal{M}_5$ , puisque qu'il produit quasiment les mêmes résultats avec 180 points de moins, améliorant ainsi le temps de calcul, ce qui est important pour une application en temps-réel.

Nous présentons toutefois une dernière itération du processus pour le valider et montrons que les images obtenues par déformation du maillage  $\mathcal{M}_4$ , ou de versions plus denses, produisent des résultats quasi-identiques. On génère dans ce but un dernier maillage raffiné  $\mathcal{M}_6$  (1050 points) et l'on compare les résultats obtenus par déformation des maillages  $\mathcal{M}_5$  (876 points) et  $\mathcal{M}_6$ . La carte de différence  $\mathcal{D}_{56}$  et son histogramme sont présentés en figure 3.53.

Cette dernière carte de différence  $\mathcal{D}_{56}$  présente des caractéristiques très proches de celles observées sur la carte  $\mathcal{D}_{45}$  (en figure 3.52). On y relève encore une fois la dominance du bleu indiquant une similarité forte. On peut également remarquer la disparition du jaune et la faible présence du vert renforçant cette similarité, ce que confirme le centrage de l'histogramme sur la gauche (dont la hauteur

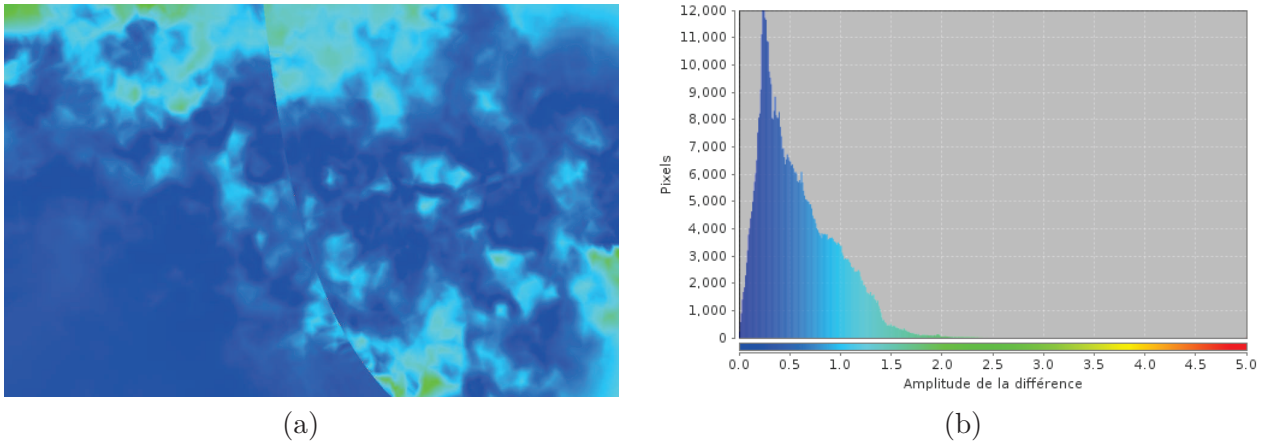


FIGURE 3.53 – Carte de différence  $\mathcal{D}_{56}$  et son histogramme.

dépasse la valeur maximum de l'échelle verticale par empilement des pixels et enregistre un pic à 14000 pixels), ainsi que l'absence de valeurs sur la droite. Pour 84.9% de ces points, la différence relevée est inférieure à un pixel. Pour 99.9%, elle est inférieure à 2 pixels et nous la considérons infime. Enfin, pour les 0.01% de points restants, la valeur maximum de cette différence n'est que de 2.4 pixels, ce qui est la meilleure valeur relevée.

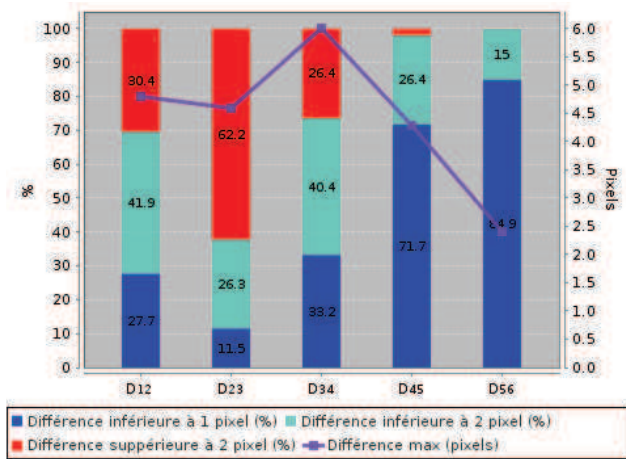
L'analyse de  $\mathcal{D}_{56}$  montre que les restaurations obtenues par déformation des maillages  $\mathcal{M}_5$  et  $\mathcal{M}_6$  sont quasi-identiques (la différence est infime pour 99.9% des pixels) ce qui nous permet de valider notre processus, car nous avons montré en comparant  $\mathcal{M}_4$ ,  $\mathcal{M}_5$  et  $\mathcal{M}_6$ , que des raffinements successifs de maillage produisent des déformations similaires.

### 3.7.4.2 Choix d'un maillage optimisé

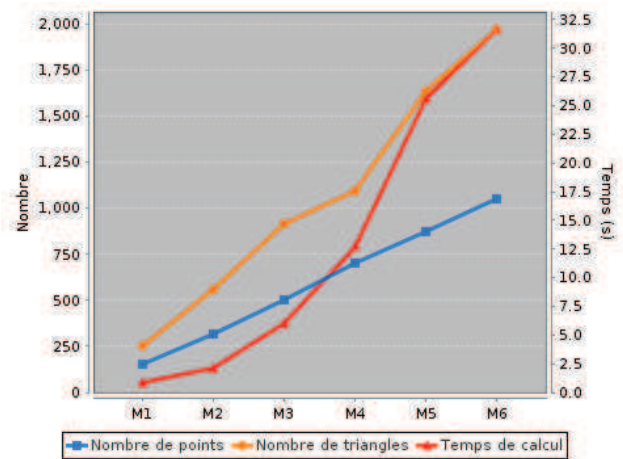
Au long de cette étude, nous avons présenté plusieurs maillages réguliers  $\mathcal{M}_1$  à  $\mathcal{M}_6$  de finesse différente et comparé les restaurations obtenues. Nous avons en particulier montré qu'elles sont similaires avec l'usage des maillages  $\mathcal{M}_4$ ,  $\mathcal{M}_5$  et  $\mathcal{M}_6$  et nous posons à présent la question de choisir un maillage "optimisé", c'est-à-dire produisant des résultats similaires au maillage "optimal" (décrit en fin de section 3.7.2) au moindre coût. Nous résumons alors l'ensemble des mesures relevées au cours du processus sur la figure 3.54.a et introduisons en figure 3.54.b des temps de calculs indicatifs du processus d'optimisation, pour les maillages  $\mathcal{M}_1$  à  $\mathcal{M}_6$ .

Le graphique de gauche (figure 3.54.a) rappelle que les restaurations successives avec des maillages de plus en plus denses ont produit des résultats différents avec les maillages  $\mathcal{M}_1$ ,  $\mathcal{M}_2$  et  $\mathcal{M}_3$  (illustrés par la forte proportion de différences supérieures à 2 pixels, en rouge, sur les cartes  $\mathcal{D}_{12}$ ,  $\mathcal{D}_{23}$  et  $\mathcal{D}_{34}$ , ainsi qu'une différence maximum élevée en violet) puis quasi-identiques avec les maillages  $\mathcal{M}_4$ ,  $\mathcal{M}_5$  et  $\mathcal{M}_6$  (de par la proportion négligeable ou nulle de rouge sur les cartes  $\mathcal{D}_{45}$  et  $\mathcal{D}_{56}$  et les faibles différences de maximum en violet). En conséquence,  $\mathcal{M}_4$ ,  $\mathcal{M}_5$  et  $\mathcal{M}_6$  sont de bons candidats de maillages optimisés.

Le graphique de droite (figure 3.54.b) nous permet de remarquer que le temps de calcul (en rouge) suit un profil général exponentiel, par rapport à l'ajout successifs d'un nombre de points constant (en



(a)



(b)

FIGURE 3.54 – **Résumé des mesures relevées et temps de calcul.** (a) Mesures relevées aux cartes de différences  $\mathcal{D}_{12}$  à  $\mathcal{D}_{56}$ , (b) informations et temps de calcul avec les maillages  $\mathcal{M}_1$  à  $\mathcal{M}_6$ .

bleu). Cette information illustre en particulier que les raffinements successifs de  $\mathcal{M}_4$  en  $\mathcal{M}_5$ , puis de  $\mathcal{M}_5$  en  $\mathcal{M}_6$ , ont des impacts importants sur le temps de calcul (l'optimisation est deux fois plus lente avec  $\mathcal{M}_6$  qu'avec  $\mathcal{M}_4$ ), alors que le gain réel entre  $\mathcal{M}_4$  et  $\mathcal{M}_6$  ne profite qu'à 2% des pixels de l'image.

Pour conclure cette étude, nous avons déterminé plusieurs candidats de maillages produisant des déformations quasi-identiques et choisissons finalement  $\mathcal{M}_4$ , qui est le plus rapide, donc le plus optimisé. Au travers de cet exemple, nous avons surtout présenté un processus général permettant de déterminer un maillage optimisé pour tout scénario de restauration. Ce maillage est optimisé car il possède un nombre de points minimum (donc est rapide) et garantit la qualité du résultat puisque son raffinement n'impacte plus (ou de façon négligeable) la déformation finale. Bien que ce processus ait un coût (il requiert en effet d'effectuer plusieurs restaurations), il peut cependant s'avérer rentable, particulièrement dans le cas où l'on choisit arbitrairement un maillage trop dense. Le temps nécessaire pour déterminer le maillage optimisé est alors semblable au temps de calcul de la déformation du maillage trop dense.



## Chapitre 4

# Processus de restauration et outils dérivés

Ayant décrit dans le chapitre 1 le besoin d'un nouvel outil de restauration sismique, spécifique pour faciliter l'interprétation sismique, puis détaillé dans les chapitres 2 et 3 le fonctionnement de cet outil, nous consacrons ce dernier chapitre à illustrer, par différentes applications, ce travail de thèse.

Nous commencerons en section 4.1 par présenter les résultats d'un cas classique de restauration d'image sismique, puis détaillons en section 4.2, les différentes applications dérivées. En détail, nous montrons dans la section 4.2.1 qu'il est possible de corrélérer des positions dans des images initiales ou restaurées, ce qui permet alors de pointer des horizons directement sur une image restaurée. Enfin, nous présentons les cas de la restauration multi-failles et multi-horizons en section 4.2.2.

### 4.1 Processus de restauration

Dans cette section, nous présentons le processus complet de restauration appliqué à une image sismique typique illustrée sur la figure 4.1. Sur cette image, l'objectif consiste à restaurer la faille principale (discontinuité au milieu de l'image) puis à aplanir la structure bombée. La restauration fournira une meilleure compréhension du système de dépôt à partir de son état de déformation actuel et permettra de clarifier la continuité de l'horizon autour de la faille principale.

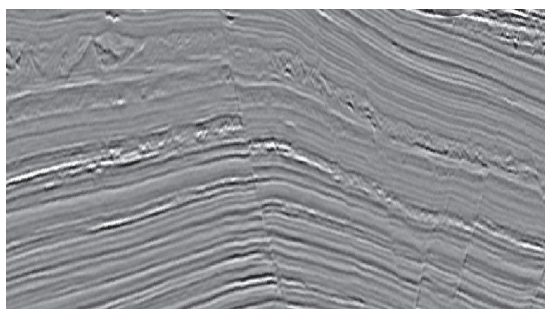


FIGURE 4.1 – Image sismique à restaurer.



La figure 4.2 présente les étapes successives intervenant au cours du processus de restauration. Dans un premier temps, un interpréteur pointe un horizon (en bleu) et une faille (en rouge) sur l'image 4.2.a. Le bloc à droite de la faille est ensuite déplacé le long de celle-ci afin de restaurer la continuité de l'horizon bleu (en figure 4.2.b), selon le processus détaillé dans la section 3.1.1. Finalement, l'horizon est mis à plat par le processus de la section 3.1.2, ce qui permet d'obtenir le résultat final sur la figure 4.2.c. Sur cette image, les interpréteurs observent une pile sédimentaire continue et plane, qui leur permet de procéder facilement à de nouveaux pointés des structures sismiques et géologiques.

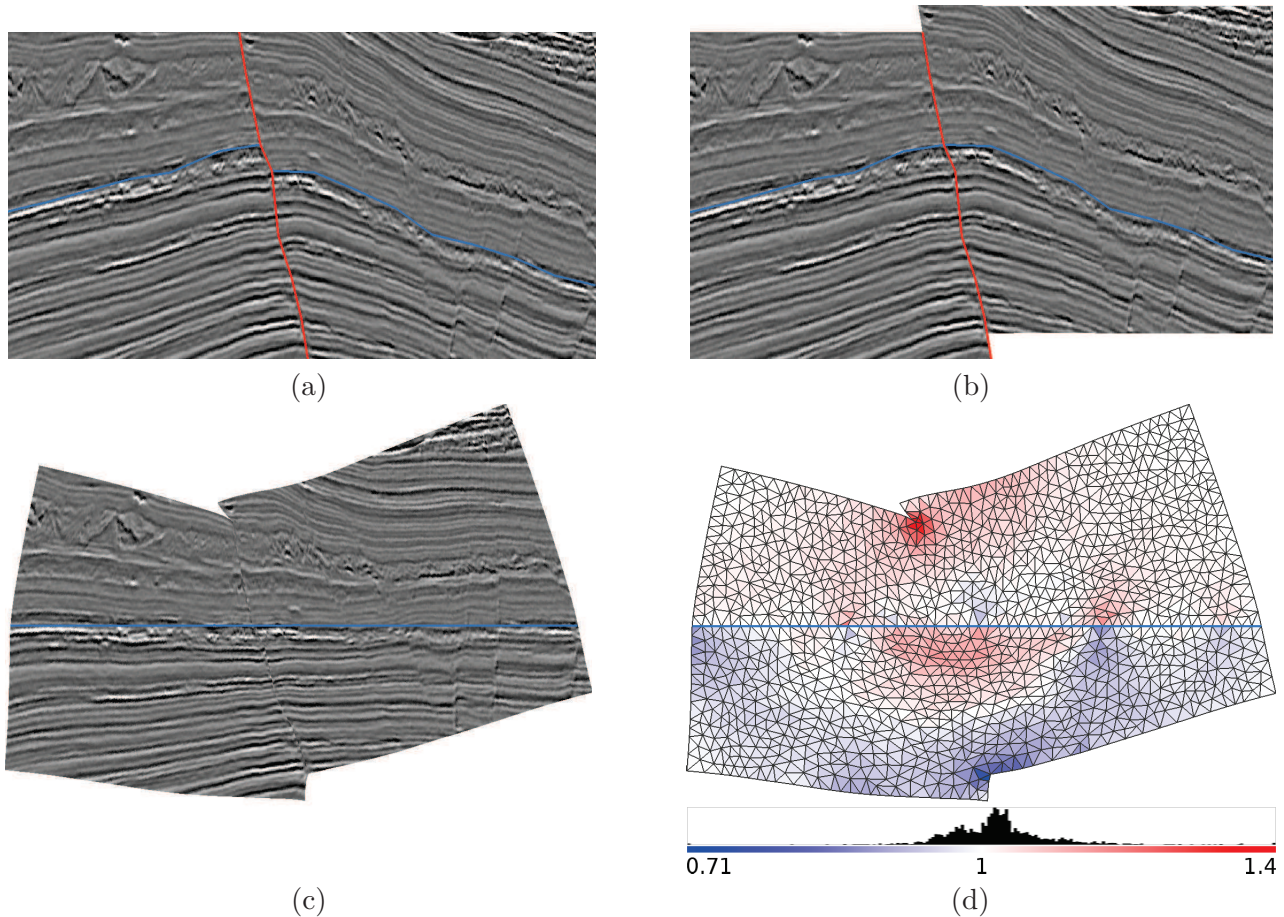


FIGURE 4.2 – **Restauration d’une image sismique.** (a) Image interprétée, (b) restauration de la faille, (c) image finale, (d) carte de déformation et histogramme : le rouge montre la compression, le bleu la dilatation.

Dans notre procédé, les contraintes géométriques entrent en concurrence et la minimisation au sens des moindres carrés ne permet généralement pas de toutes les respecter parfaitement. La figure 4.2.d montre un critère de qualité de déformation (le rapport entre l’aire initiale et l’aire finale) pour chaque triangle qui permet de quantifier le respect de la contrainte d’aire. Cette visualisation révèle les régions de compression (rouge) et dilatation (bleu) résultant de la restauration. L’histogramme associé atteste du bon comportement de la minimisation de contraintes puisque la majorité des triangles ne présente que de très faibles distorsions.

La qualité et la rapidité de l'opération de restauration (1 à 3 secondes de temps de calcul typiquement en pratique) permet d'intégrer cet outil à la chaîne des traitements fournie aux interpréteurs. Le fait qu'un tel outil de restauration soit disponible nous a également permis d'élaborer plusieurs outils dérivés que nous abordons dans la section suivante.

## 4.2 Outils dérivés

Nous détaillons dans les sections suivantes des outils permettant le pointé de structures sur une section sismique initiale ou restaurée, grâce à une fonction de correspondance bijective qui les lie, et la gestion des configurations complexes qu'un interpréteur sera à même de rencontrer dans des situations réelles, à savoir la restauration multi-failles et multi-horizons.

### 4.2.1 Correspondance des pointés entre l'espace initial et l'espace restauré

La restauration facilite l'interprétation de la structure du sous-sol en montrant des évènements géologiques plus simples à identifier sur des données sismiques planes et continues. Il est alors pratique de visualiser en parallèle l'espace initial et l'espace restauré en offrant un curseur lié entre les deux vues par une fonction de correspondance.

Le scénario suivant illustre l'intérêt d'une telle fonction de correspondance. Sur la figure 4.3.a, une interface continue (en orange) semble traverser la faille (en rouge). Pour vérifier cette hypothèse, un géologue pointe un ensemble de positions pertinentes dans l'image initiale (comme illustré sur la figure 4.3.b) puis observe les positions correspondantes sur l'image restaurée (en figure 4.3.c). Dans cet exemple, il apparaît clairement que l'interface orange de la figure 4.3.a n'est pas continue.

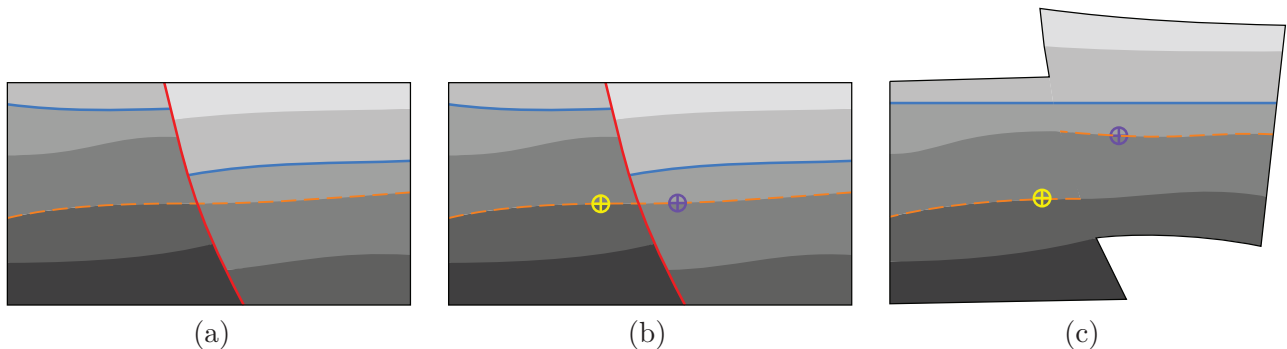


FIGURE 4.3 – **Transformations de positions de l'image initiale vers l'image restaurée :** (a) Dans l'image initiale, le marqueur orange illustre une interface qui semble continue. (b) Pointé de positions en jaune et violet pour vérifier cette hypothèse. (c) Dans l'espace restauré, les pointés apparaissent clairement sur des horizons différents.

Prenons un autre scénario. Un géologue discerne aisément le chenal sur l'image restaurée présentée sur la figure 4.4.a. En y plaçant quelques points (en jaune sur la figure 4.4.b) il peut en retrouver les contours sur l'image initiale (en figure 4.4.c) malgré la discontinuité introduite par la faille (en rouge).

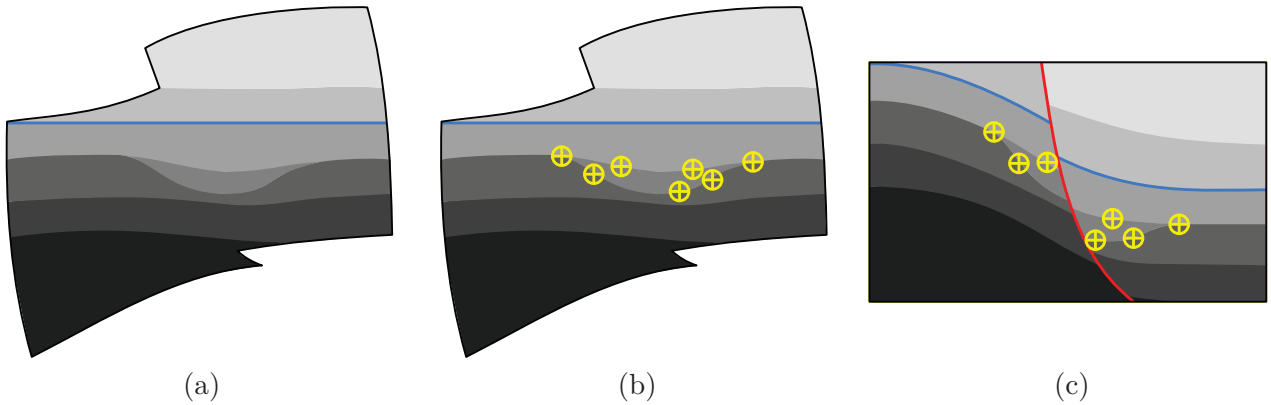


FIGURE 4.4 – **Transformations de positions de l’image restaurée vers l’image initiale** : (a) Image restaurée sur laquelle on peut identifier un chenal (b) Pointé de positions dans l’image restaurée pour définir sa géométrie (c) Transformation de cette géométrie dans l’image initiale.

Une extension naturelle de ce scénario est le pointé d’horizons depuis l’espace restauré et la transformation de ces pointés dans l’espace initial. Ces tâches nécessitent une fonction de correspondance bijective que nous détaillons dans les sections suivantes.

#### 4.2.1.1 Fonctions de correspondance d’un espace à un autre

Le passage d’une position dans une image sismique à la position correspondante dans la version restaurée doit tenir compte des deux étapes de transformation qui composent l’opération de restauration : glissement le long de la faille puis aplatissement de l’horizon (voir section 4.1).

La première étape détermine pour un ensemble discret de positions appartenant à un espace initial  $\mathcal{A}$  l’ensemble des vecteurs déplacement qui transforment ces positions dans un espace non faillé  $\mathcal{B}$ . En définissant une fonction  $f_{AB}$  qui interpole les vecteurs déplacement, toute position de l’espace  $\mathcal{A}$  trouve une position correspondante dans l’espace  $\mathcal{B}$ .

De manière similaire, pour la seconde étape, une fonction  $f_{BC}$  transforme les positions de l’espace non faillé  $\mathcal{B}$  vers l’espace restauré  $\mathcal{C}$ . La fonction composée  $f_{AC}$  de ces deux fonctions nous permet ainsi de transformer la position  $p$  de tout point de  $\mathcal{A}$  en sa position correspondante  $p'$  dans  $\mathcal{C}$  :

$$p' = f_{AC}(p) = (f_{BC} \circ f_{AB})(p) = f_{BC}(f_{AB}(p)). \quad (4.1)$$

La transformation inverse  $f_{CA}$  qui transforme les positions de l’espace restauré vers l’espace initial nécessite de recourir à l’inverse des fonctions  $f_{AB}$  et  $f_{BC}$  :

$$\begin{aligned} f_{CA} &= f_{AC}^{-1} \\ &= (f_{BC} \circ f_{AB})^{-1} \\ &= f_{AB}^{-1} \circ f_{BC}^{-1} \\ &= f_{BA} \circ f_{CB}, \end{aligned} \quad (4.2)$$



avec  $f_{BA} = f_{AB}^{-1}$  et  $f_{CB} = f_{BC}^{-1}$ . Ces notations sont reprises sur la figure 4.5.

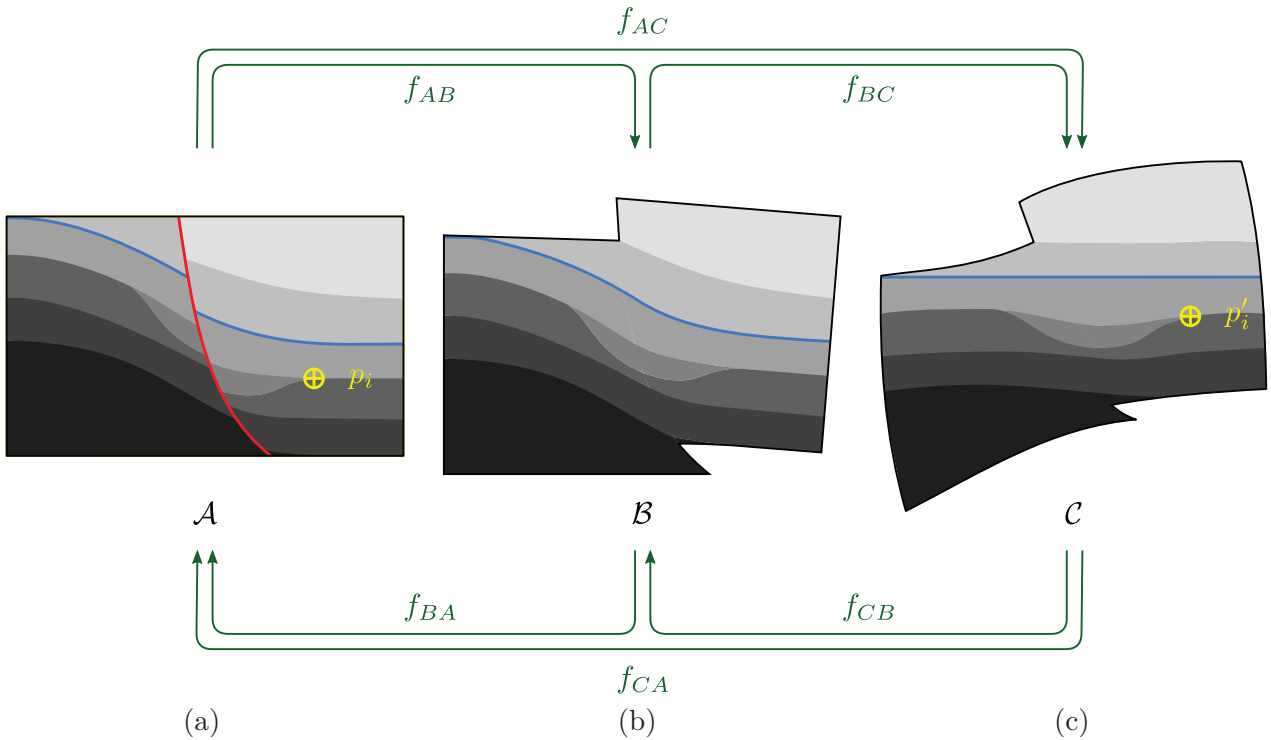


FIGURE 4.5 – **Fonctions de transformation entre différents espaces** : (a) espace initial, (b) espace non faillé, (c) espace restauré.

Les fonctions  $f_{AB}$  et  $f_{BC}$  étant définies comme l'interpolation d'un ensemble de vecteurs déplacement  $\{\vec{d}_{AB_i}\}$  et  $\{\vec{d}_{BC_i}\}$ , une approximation du premier ordre nous permet de définir  $f_{BA}$  et  $f_{CB}$  comme l'interpolation des vecteurs  $\{-\vec{d}_{AB_i}\}$  et  $\{-\vec{d}_{BC_i}\}$ .

Quel que soit l'espace de départ, le vecteur déplacement à appliquer en une position particulière  $p$  est obtenu par une interpolation barycentrique linéaire des vecteurs déplacement, se trouvant aux sommets  $p_1$ ,  $p_2$  et  $p_3$  du triangle contenant cette position. Ainsi, si  $\vec{d}_1$ ,  $\vec{d}_2$  et  $\vec{d}_3$  sont les vecteurs associés aux points  $p_1$ ,  $p_2$  et  $p_3$ , le vecteur  $\vec{d}$  au point  $p$  est obtenu en appliquant la formule :

$$\vec{d} = \frac{A_1 \vec{d}_1 + A_2 \vec{d}_2 + A_3 \vec{d}_3}{A_1 + A_2 + A_3} \quad (4.3)$$

où  $A_1$  est l'aire du triangle  $pp_2p_3$ ,  $A_2$  l'aire du triangle  $p_1pp_3$  et  $A_3$  l'aire du triangle  $p_1p_2p$  (voir figure 4.6).

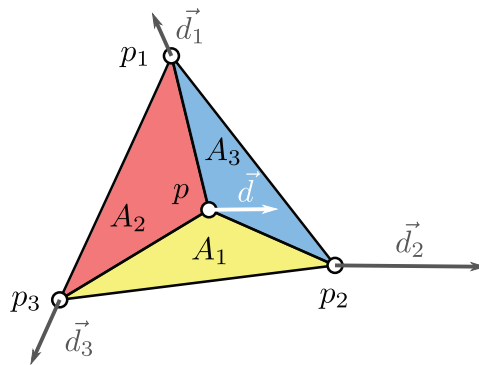


FIGURE 4.6 – Interpolation barycentrique des vecteurs déplacement à l’intérieur d’un triangle.

#### 4.2.1.2 Pointé d’horizons dans l’espace restauré

La fonction de correspondance  $f_{CA}$  (équation 4.2) entre l’espace restauré et l’espace initial nous a permis de définir un outil plus confortable pour le pointé d’horizons. Un scénario d’utilisation de cet outil est illustré sur la figure 4.7 : à partir d’une image sismique initiale (figure 4.7.a), un premier

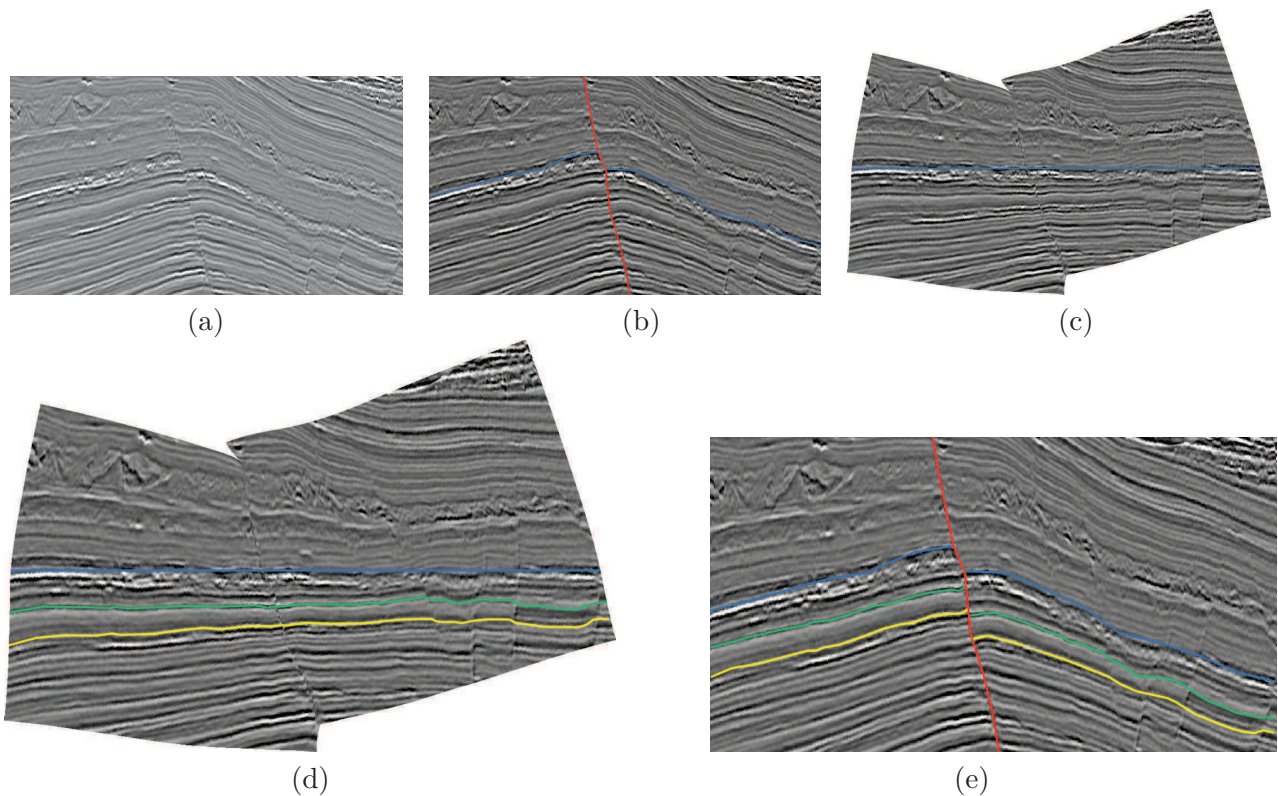


FIGURE 4.7 – Pointé facilité d’horizons sur l’image restaurée et exportation dans l’image initiale. (a) Image sismique initiale, (b) pointés d’une faille et d’un horizon dans l’image sismique initiale, (c) image sismique restaurée, (d) pointé facilité d’horizons dans l’image restaurée, (e) export des pointés dans l’image initiale grâce à la fonction de transformation  $f_{CA}$ .

horizon est interprété de part et d’autre d’une faille (figure 4.7.b), puis restauré (figure 4.7.c). Les

géologues peuvent alors effectuer un pointé facilité d'autres horizons (figure 4.7.d). Finalement, par application de la fonction de transformation  $f_{CA}$  sur l'ensemble de la géométrie des nouveaux pointés, les horizons vert et jaune sont exportés vers l'image initiale (figure 4.7.e).

## 4.2.2 Restauration multi-failles et multi-horizons

En nous appuyant sur les opérations de restauration de faille et de mise à plat d'horizon détaillées dans les parties précédentes, nous envisageons à présent des scénarios de restauration plus complexes et donc également plus réalistes, s'appliquant à des sections sismiques comportant plusieurs failles et plusieurs horizons.

### 4.2.2.1 Restauration d'une section multi-faillée

L'extension de notre processus de restauration au cas multi-failles est relativement simple puisqu'il s'agit d'appliquer ce processus à chacune des failles, tour à tour. Une seule faille est alors active à un moment donné et le cas multi-failles revient à une succession de restaurations mono-faille. Il convient toutefois de considérer les failles dans le bon ordre : en effet, la restauration n'est pas commutative et en pratique plusieurs restaurations faisant varier l'ordre des failles produisent des résultats finaux différents. Il appartient donc aux géologues de déterminer la chronologie de formation des failles et d'appliquer à chacune d'entre elles le processus de restauration dans l'ordre inverse de leur formation, c'est-à-dire tel que la plus récente soit restaurée en priorité.

La figure 4.8 illustre le déroulement complet de la restauration d'une section présentant deux failles. Dans cet exemple, la faille rouge (en figure 4.8.a) est considérée comme la dernière formée et sera donc restaurée en premier. À ce stade la deuxième faille est ignorée et la totalité du bloc de droite est maillé (comme le montre la figure 4.8.b). Il est alors déplacé et déformé (en figure 4.8.c), puis l'image sismique initiale y est interpolée (en figure 4.8.d). Dans cet exemple, la seconde faille a également subi la déformation du bloc. Le processus se poursuit de la même manière avec cette seconde faille et le bloc situé à sa droite (comme détaillé sur les figures 4.8.e, f, g et h). Finalement, l'horizon qui a subi la flexure est mis à plat, permettant d'obtenir l'image restaurée en figure 4.8.i. Notons enfin que le processus de restauration présenté dans cet exemple avec deux failles, se généralise aisément pour un nombre quelconque de failles.

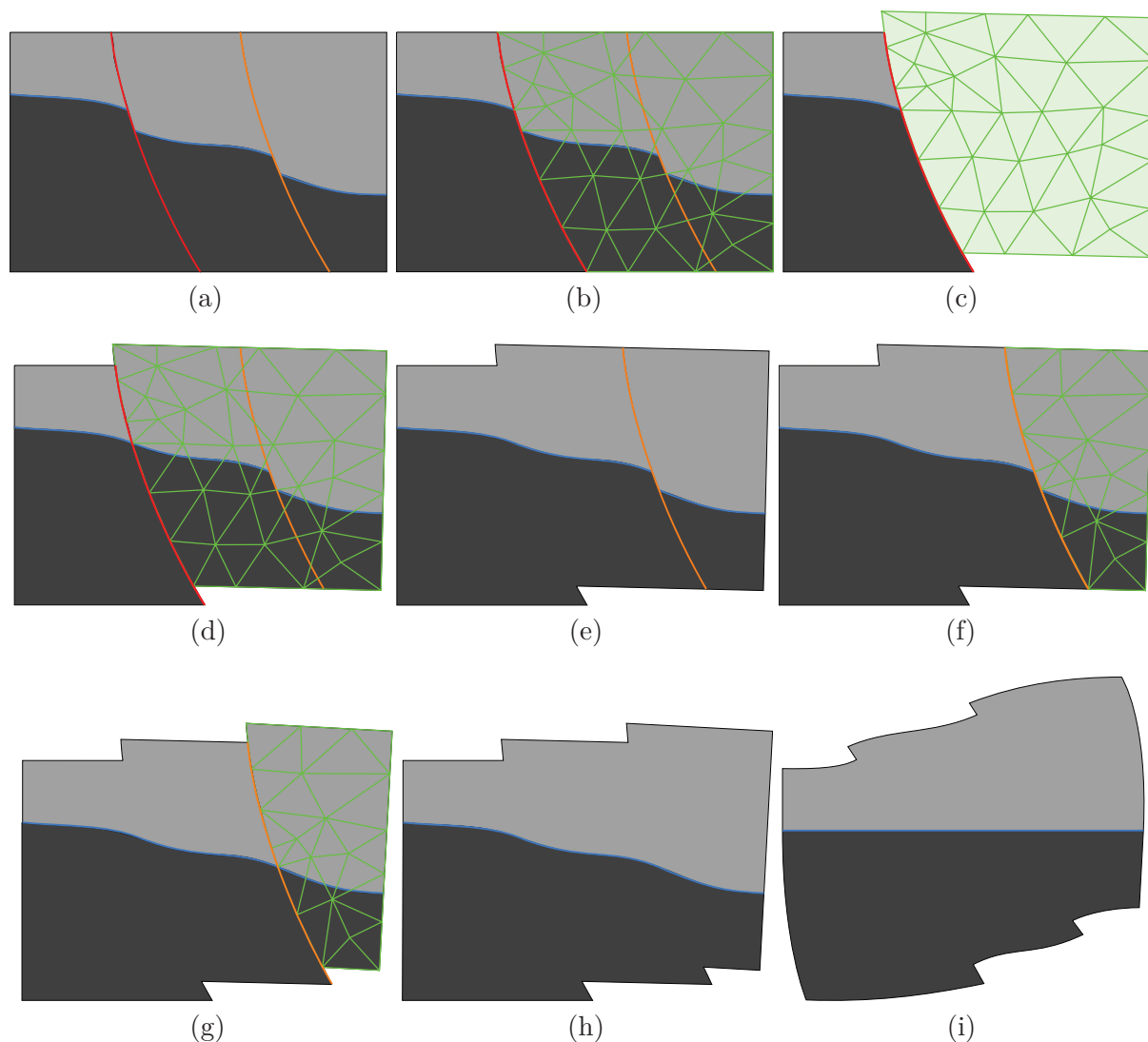


FIGURE 4.8 – **Processus de restauration d’une section sismique contenant plusieurs failles.** (a) Image sismique initiale présentant deux failles et un horizon, (b) maillage de la partie droite de l’image pour restaurer la faille rouge, (c) maillage déformé après l’opération de restauration de la faille rouge, (d) application de la fonction de déformation obtenue par interpolation des déplacements aux points du maillage sur la partie droite de l’image sismique et sur la faille orange, (e) suppression de la faille rouge restaurée, (f) maillage de la partie droite de l’image pour restaurer la faille orange, (g) restauration de la faille orange, (h) suppression de la faille orange restaurée, (i) mise à plat de l’horizon pour obtenir l’image restaurée.

L’ensemble de ce processus est illustré sur un cas réel en figure 4.9, où nous présentons des captures d’écran de notre outil de restauration. La figure 4.9.a illustre l’image sismique à restaurer, comportant trois failles (rouge, jaune et rose) et un horizon (bleu). Nous restaurons alors successivement la faille rouge (en figure 4.9.b), la faille jaune (en figure 4.9.c) et la faille rose (sur la figure 4.9.d), puis mettons l’horizon à plat en figure (4.9.e) et obtenons une image restaurée sans discontinuités, dont les différentes couches sédimentaires sont planes.



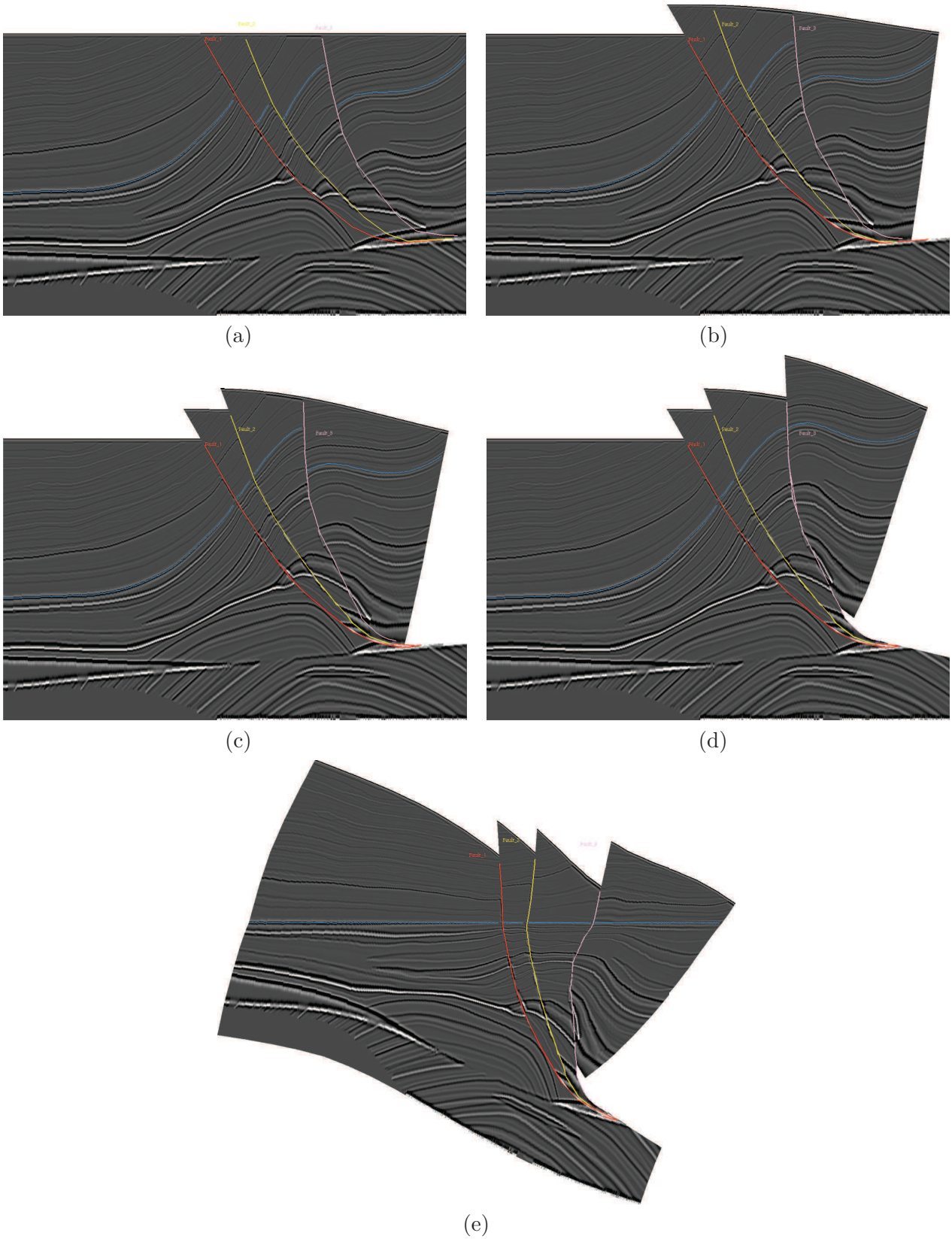


FIGURE 4.9 – Cas réel de restauration d’une section sismique contenant plusieurs failles. (a) Image sismique initiale présentant deux failles et un horizon, (b) restauration de la faille rouge, (c) restauration de la faille jaune, (d) restauration de la faille rose, (e) mise à plat de l’horizon bleu pour obtenir l’image restaurée.

#### 4.2.2.2 Restauration d'une section contenant plusieurs horizons

Nous nous intéressons maintenant au cas de la restauration multi-horizons (consistant à restaurer des couches sismiques). Le mode de résolution est ici différent du cas multi-failles puisque les horizons seront mis à plat en même temps, lors d'une unique opération de minimisation globale de contraintes. En effet, si cette mise à plat n'était pas simultanée la restauration d'un horizon qui affecte l'ensemble de l'image sismique déformerait les horizons précédemment mis à plat. Cette approche implique donc d'intégrer un jeu de contraintes par horizon à traiter. Nous présentons donc dans cette section la restauration d'une image sismique contenant une faille et deux horizons qui s'effectuera en deux temps : par la restauration de la faille, puis la mise à plat de la couche sismique segmentée par les deux horizons. Précisons toutefois que nous nous limitons volontairement à une faille afin de simplifier la description du processus. Nous verrons alors que celui-ci se généralise à plusieurs failles.

##### a. Restauration de la faille

Le but de cette première opération est de restaurer la continuité de plusieurs horizons de part et d'autre d'une faille avant de les mettre à plat. La figure 4.10.a montre le cas d'une section sismique dans laquelle deux horizons (bleu et violet) ont été interprétés de chaque côté d'une faille (rouge).

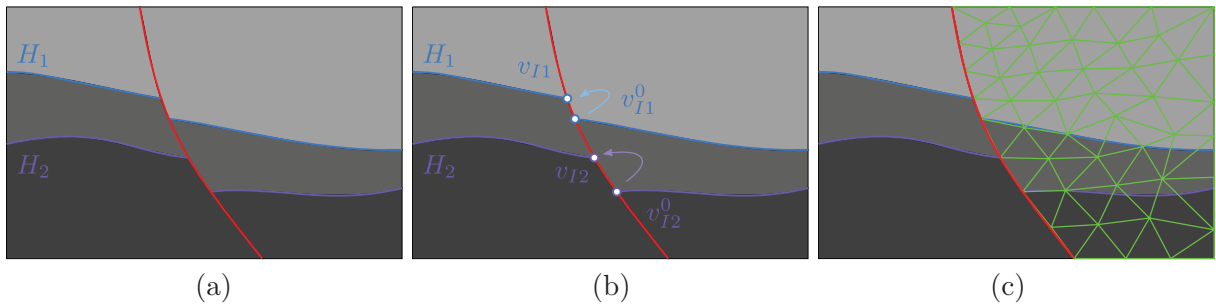


FIGURE 4.10 – **Processus de restauration d'une section sismique contenant une faille et plusieurs horizons.** (a) Image sismique initiale présentant une faille et deux horizons  $H_1$  et  $H_2$ . (b) Illustration des couples de points  $\{v_{I1}^0, v_{I1}\}$  et  $\{v_{I2}^0, v_{I2}\}$  devant être confondus pour restaurer la continuité des horizons  $H_1$  et  $H_2$ . (c) Maillage de l'image, conforme à la géométrie de la faille et incluant les points de discontinuité des horizons  $\{v_{I1}^0, v_{I1}, v_{I2}^0, v_{I2}\}$ .

La continuité des horizons  $H_1$  et  $H_2$  sera assurée en déplaçant le point  $v_{I1}^0$  à la position  $v_{I1}$  et le point  $v_{I2}^0$  à la position  $v_{I2}$  visibles sur la figure 4.10.b. Ces points devront faire partie des sommets du maillage afin de pouvoir leur associer les contraintes de déplacement le long de la faille (figure 4.10.c).

Nous nous intéressons à présent aux contraintes de restauration intervenant dans cette opération et rappelons que lors du cas simple dans lequel n'intervient qu'un seul horizon, les contraintes de restauration se décomposent en une contrainte de continuité de l'horizon et une contrainte de contact à la faille. Étant donné que ce nouveau scénario présente un horizon supplémentaire, deux contraintes de continuité seront requises. Ainsi, similairement à l'équation (3.7), les deux critères  $J_{F1}$  et  $J_{F2}$  à minimiser pour restaurer la continuité des horizons s'expriment par :

$$\begin{aligned} J_{F1} &= \|\mathbf{v}_{I1}^0 - \mathbf{v}_{I1}\|^2 \\ &= \|C_{I1}V - \mathbf{v}_{I1}\|^2, \end{aligned} \quad (4.4)$$

$$\begin{aligned} J_{F2} &= \|\mathbf{v}_{I2}^0 - \mathbf{v}_{I2}\|^2 \\ &= \|C_{I2}V - \mathbf{v}_{I2}\|^2, \end{aligned} \quad (4.5)$$

où  $V$  est la matrice (de taille  $n \times 2$ ) des positions définies par l'équation (3.5), et  $C_{I1}$  et  $C_{I2}$  sont les matrices (de taille  $1 \times n$ ) isolant  $v_{I1}^0$  et  $v_{I2}^0$  parmi les points de  $\mathcal{V}$ . On écrit :

$$C_{I1,s} = \begin{cases} 1 & \text{lorsque } v_{I1}^0 \text{ est le } s^{\text{ème}} \text{ élément de } \mathcal{V}, \\ 0 & \text{sinon.} \end{cases} \quad (4.6)$$

$$C_{I2,s} = \begin{cases} 1 & \text{lorsque } v_{I2}^0 \text{ est le } s^{\text{ème}} \text{ élément de } \mathcal{V}, \\ 0 & \text{sinon.} \end{cases} \quad (4.7)$$

Après assemblage des contraintes  $J_{F1}$  et  $J_{F2}$  avec la contrainte de contact que nous appelons  $J_{F3}$  selon l'équation (3.54), le terme final à minimiser pour restaurer la faille en retrouvant la continuité des horizons est :

$$J1 = \omega_G J_{S_g} + \omega_G J_{S_p} + \omega_G J_L + \omega_R J_{F1} + \omega_R J_{F2} + \omega_R J_{F3}, \quad (4.8)$$

où  $\omega_G$  et  $\omega_R$  sont des constantes choisies tel que nous l'avons décrit en section 3.3.1.3.

Sur l'image sismique à restaurer de la figure 4.11.a, contenant une faille et deux horizons, l'application de cette opération produit l'image intermédiaire en figure 4.11.b, où les horizons sont continus.

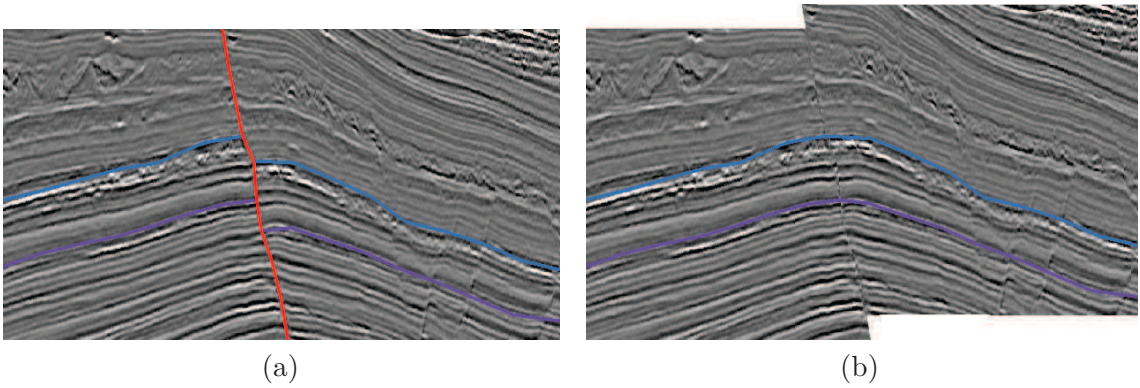


FIGURE 4.11 – **Restauration de la faille sur une image sismique contenant deux horizons.** (a) Image sismique à restaurer contenant deux horizons (bleu et violet), coupés par une faille rouge, (b) restauration de la faille rétablissant la continuité des horizons.



## b. Mise à plat de la couche sismique segmentée par les deux horizons

Afin d'achever la restauration, intéressons nous à présent au procédé permettant de mettre à plat une couche sismique segmentée par deux horizons. La section en figure 4.12.a doit être déformée de manière à ce que ses deux horizons ( $H_1$  en bleu et  $H_2$  en violet) convergent vers les positions verticales distinctes  $y_{H_1}$  et  $y_{H_2}$ , illustrées en figure 4.12.b.

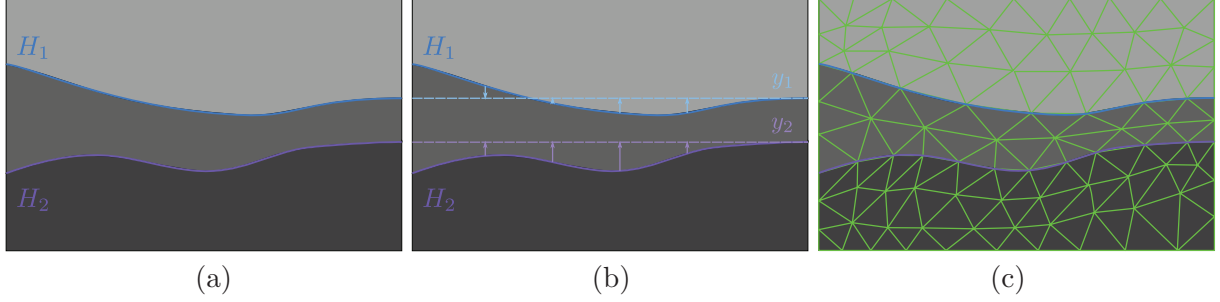


FIGURE 4.12 – **Processus de restauration d'une section sismique contenant plusieurs horizons.** (a) Image sismique initiale présentant deux horizons  $H_1$  et  $H_2$ . (b) Illustration des droites horizontales d'équations  $y = y_{H_1}$  et  $y = y_{H_2}$ , grâce auxquelles on souhaite respectivement remettre à plat  $H_1$  et  $H_2$ . (c) Maillage de l'image, conforme à la géométrie des horizons.

Le domaine est alors recouvert d'une triangulation contrainte aux géométries des deux horizons (figure 4.12.c) et un jeu de contraintes est associé aux points disposés sur chacun des horizons.

Rappelons que le critère de mise à plat d'un horizon consiste à minimiser la somme quadratique de l'erreur verticale à la droite horizontale cible (voir équation 3.14). Ainsi, en considérant  $\mathcal{V}_{h_1}$  le sous-ensemble des points de  $\mathcal{V}$  appartenant à l'horizon bleu, et  $\mathcal{V}_{h_2}$  celui appartenant à l'horizon violet, nous souhaitons minimiser les critères  $J_{H_1}$  et  $J_{H_2}$  suivants :

$$\begin{aligned} J_{H_1} &= \sum_{v_i \in \mathcal{V}_{h_1}} (y_{H_1} - y_i)^2, \\ J_{H_2} &= \sum_{v_j \in \mathcal{V}_{h_2}} (y_{H_2} - y_j)^2. \end{aligned} \quad (4.9)$$

Ces termes se réécrivent sous la forme matricielle suivante :

$$\begin{aligned} J_{H_1} &= \|C_{H_1}V - c_{H_1}(V)\|^2, \\ J_{H_2} &= \|C_{H_2}V - c_{H_2}(V)\|^2, \end{aligned} \quad (4.10)$$

où  $V$  est la matrice (de taille  $n \times 2$ ) des positions définie dans l'équation (3.5),  $C_{H_1}$  est la matrice (de taille  $\text{card}(\mathcal{V}_{h_1}) \times n$ ) qui extrait l'ensemble  $\mathcal{V}_{H_1}$  de  $\mathcal{V}$  et  $c_{H_1}(V)$  la matrice (de taille  $\text{card}(\mathcal{V}_{h_1}) \times 2$ ) contenant les projetés de  $\mathcal{V}_{H_1}$  à la position verticale  $y_1$ .  $C_{H_2}$  et  $c_{H_2}(V)$  ont des définitions similaires pour l'horizon  $H_2$ . On écrit :

$$C_{H_1 r, s} = \begin{cases} 1 & \text{lorsque } v_{H_1} \text{ est le } r^{\text{ème}} \text{ élément de } \mathcal{V}_{h_1} \text{ et le } s^{\text{ème}} \text{ élément de } \mathcal{V}, \\ 0 & \text{sinon,} \end{cases} \quad (4.11)$$

$$c_{H_1}(V)_i = [x_i, y_{H_1}], \forall v_i \in \mathcal{V}_{h_1}, \quad (4.12)$$

$$C_{H_2r,s} = \begin{cases} 1 & \text{lorsque } v_{H_2} \text{ est le } r^{\text{ème}} \text{ élément de } \mathcal{V}_{h_2} \text{ et le } s^{\text{ème}} \text{ élément de } \mathcal{V}, \\ 0 & \text{sinon,} \end{cases} \quad (4.13)$$

$$c_{H_2}(V)_j = [x_j, y_{H_2}], \forall v_j \in \mathcal{V}_{h_2}. \quad (4.14)$$

Ainsi, conformément à l'équation (3.54), le terme à minimiser pour obtenir la restauration de deux horizons tout en respectant les contraintes géométriques est :

$$J = \omega_G J_{S_g} + \omega_G J_{S_p} + \omega_G J_L + \omega_R J_{H_1} + \omega_R J_{H_2}, \quad (4.15)$$

où  $\omega_G$  et  $\omega_R$  sont des constantes choisies tel que nous l'avons décrit en section 3.3.1.3. Nous obtenons finalement l'image sismique restaurée en figure 4.13.

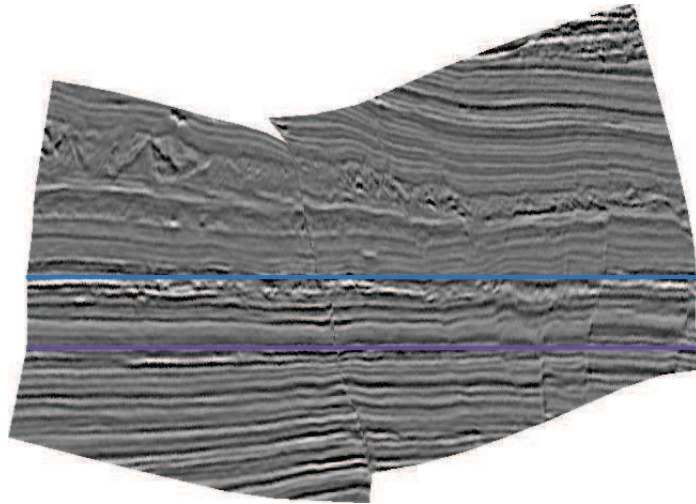


FIGURE 4.13 – Section sismique restaurée contenant deux horizons.

Il est à toutefois à noter que l'objectif de mise à plat d'une couche segmentée par deux horizons peut provoquer de fortes distorsions sur les triangles qui la compose. Dans notre approche, la seule marge de manœuvre disponible pour le moment consiste à choisir des coordonnées verticales limitant leur étirement vertical. Une amélioration de l'algorithme consisterait à fixer la coordonnée verticale d'un seul horizon et à déduire les autres lors du processus de minimisation.

### c. Généralisation au cas multi-faille

Nous avons détaillé dans les sections précédentes tous les outils nécessaires à la restauration d'une section sismique comportant plusieurs failles et plusieurs horizons. En s'appuyant sur les sections

4.2.2.1 et 4.2.2.2.a, les failles sont considérées l'une après l'autre afin de mettre en contact les horizons de part et d'autre. Une fois toutes les failles traitées, les horizons sont mis à plat selon le processus expliqué dans la section 4.2.2.2.b.

### 4.3 Conclusions

Ce chapitre fut l'occasion de présenter divers exemples de cas réels qui illustrent le fonctionnement de notre outil lors de l'interprétation préliminaire d'images sismiques. La fonction de correspondance entre images initiales et restaurées offre à l'interpréteur des curseurs liés dans les vues respectives. Elle permet en pratique de pointer des structures dans la vue la plus adéquate, ou de faciliter l'identification de structures discontinues sur les images non restaurées.

En nous basant sur les opérations élémentaires de restauration de faille et d'horizon, détaillées dans le chapitre 3, nous avons montré qu'il était simple de les combiner pour élaborer des outils permettant de travailler sur des configurations complexes multi-failles ou multi-horizons. Cette dernière fonctionnalité permet de tester interactivement différentes hypothèses (comme par exemple l'ordre de restauration des failles) ce qui s'avère utile pour reconstituer l'histoire du sous-sol.

Enfin, précisons que l'outil de restauration présenté est aujourd'hui en cours d'intégration dans la plateforme logicielle d'interprétation SISMAGE, utilisée par plus de 500 personnes à travers le monde.

# Conclusions

Dans ce mémoire, j'ai présenté mes travaux de thèse, qui ont consisté à concevoir un outil de restauration d'images sismiques dédié aux étapes préliminaires de l'interprétation sismique.

L'interprétation d'images sismiques consiste à décrire la structure du sous-sol observée en pointant sa structure géologique (failles et horizons), puis en tâchant d'expliquer les mécanismes qui ont conduit à sa formation en élaborant différentes hypothèses de déformation. Nous avons montré qu'aux étapes préliminaires d'une interprétation, c'est-à-dire lors de l'analyse initiale d'images sismiques, le pointé de certains éléments pouvait s'avérer ardu car les images à interpréter sont déformées et présentent des discontinuités dues à la présence de failles géologiques dans le milieu. Ce processus d'interprétation se termine par une étape de validation appelée la restauration qui permet de tester la cohérence des différents pointés avec les hypothèses de déformation. Pour une image sismique, la restauration consiste à appliquer la déformation inverse de manière à obtenir un modèle représentant le sous-sol dans une configuration temporellement antérieure aux différentes déformations produites. Nous avons alors considéré que la restauration utilisée comme outil d'aide à l'interprétation pouvait améliorer ce processus, en restaurant l'image et en proposant à l'interpréteur une vision plus claire de l'empilement de couches sédimentaires. Cette approche permet d'une part d'aider l'interpréteur dans le cas de pointés ambigus, puisque la sismique restaurée est plane et continue, et permet d'autre part d'introduire une dimension de validation dès les premiers pointés. En procédant de la sorte par raffinements successifs, elle renforce donc la confiance du géoscientifique dans son interprétation, qui, rappelons le, repose sur des hypothèses géologiques faites par l'interpréteur, et de ce fait toute aide à l'interprétation est précieuse.

Dans le but d'introduire un outil de restauration dès les premières étapes de l'interprétation, nous avons dans un premier temps étudié les différentes méthodes de restauration décrites dans la littérature. Nous avons constaté qu'elles requéraient une interprétation terminée, c'est-à-dire l'ensemble des pointés de failles et d'horizons ainsi qu'une hypothèse décrivant la chronologie et la nature des déformations successives qui se sont produites dans le sous-sol. Or, comme notre application sera utilisée dans le contexte de l'interprétation sismique préliminaire, l'ensemble de ces données n'est pas encore connu et donc pas disponible pour nos méthodes de traitement. Nous avons donc choisi de concevoir un outil de restauration d'images sismiques, en nous appuyant sur des critères de déformation classiques.

Afin de mesurer ce que la création d'un outil de déformation d'image implique, nous nous sommes intéressés au vaste domaine de la déformation de modèles numériques. Nous avons remarqué que malgré des domaines d'applications multiples, toute déformation se décompose essentiellement en trois étapes distinctes : l'application de contraintes sur un ensemble de points représentant le modèle, le calcul du déplacement de ces points, et la déformation du modèle par interpolation de ces déplacements, et que pour chacune de ces étapes un nombre important d'approches existent.

Ainsi, concevoir notre outil a consisté lors de chaque étape à déterminer l'approche la plus adaptée. Nous avons en particulier identifié le fait que l'image sismique serait soumise à des contraintes externes de restauration (pour rétablir la continuité d'un horizon rompue par une faille, pour faire glisser une partie de l'image le long de cette faille, ou encore pour mettre à plat l'horizon) et que ces contraintes externes s'appliqueraient à des ensembles de points spécifiques (qui selon les cas, définissent la géométrie d'une faille ou d'un horizon). Nous avons également identifié le fait que pour produire une déformation réaliste, l'image sismique devait posséder des contraintes intrinsèques (ou internes). Nous avons choisi ces contraintes internes selon les critères classiques de la restauration et avons retenu les critères géométriques de préservation des aires et des longueurs. Sachant toutefois qu'appliquer ces contraintes internes à l'ensemble des pixels d'une image sismique serait coûteux et donc peu réaliste en vue d'une application en temps quasi-réel, nous avons choisi d'échantillonner l'image par un maillage régulier, ce qui permet de n'appliquer les contraintes qu'aux points du maillage.

Pour calculer la déformation du maillage, nous avons traduit chaque critère par un ensemble d'équations mathématiques exprimant à un état donné du maillage, l'erreur par rapport à l'objectif imposé par la contrainte. En combinant les différents termes nous avons ainsi obtenu une formulation aux moindres carrés à minimiser. Comme cet ensemble d'équations n'est pas linéaire, ce qui interdit sa résolution directe, nous avons utilisé une méthode itérative de *Gauss-Newton*, qui est bien adaptée pour résoudre ce type de problèmes. Nous avons distingué les contraintes externes (imposées) et les contraintes internes (à optimiser) par l'introduction de pondérations permettant de privilégier certains termes, mais avons également et alternativement proposé une formulation permettant une résolution stricte mais plus coûteuse fondée sur des multiplicateurs de *Lagrange*. Dès lors, l'optimisation du système d'équations permet de déterminer le maillage déformé qui respecte les contraintes externes imposées tout en respectant également au mieux les contraintes internes.

Ayant obtenu un tel maillage déformé, c'est-à-dire un vecteur déplacement au niveau de l'ensemble de ses points, nous avons dû transmettre cette déformation discrète à toute l'image sismique pour obtenir la version restaurée de celle-ci. Nous avons choisi d'utiliser la technique classique du *texture mapping*, qui permet de lier une image à un maillage puis de déformer ce maillage pour obtenir une image déformée, ce qui revient à effectuer en chaque maille une interpolation barycentrique linéaire des vecteurs déplacements. Nous avons toutefois remarqué que la déformation calculée par notre processus d'optimisation est généralement non linéaire (non homogène pour l'ensemble de l'image) et qu'en conséquence l'interpolation pouvait se révéler imprécise lorsque l'on utilise une taille de

maille un peu trop grande. Nous avons alors proposé un processus qui permet, pour un scénario de restauration donné, de déterminer une taille de maille optimisée, c'est-à-dire une taille assurant une interpolation précise tout en garantissant un nombre de points minimisé et donc gardant un coût de calcul raisonnable.

À travers les résultats présentés et par différents exemples d'applications, nous avons montré que notre contribution répondait à la problématique posée, qui était d'assister les géologues dans l'interprétation préliminaire d'images sismiques. Cette contribution a été intégrée dans la plateforme logicielle SISMAGE qui est l'outil de référence utilisé par TOTAL pour les travaux d'interprétation. Notre outil permet de restaurer une image sismique pour présenter l'information de manière simplifiée, contribuer à lever les doutes dans les cas ambigus, ou permettre les pointés directement dans cet espace simplifié. Il est également capable de traiter des cas complexes impliquant plusieurs failles ou plusieurs horizons, ce qui permet de tester interactivement différentes hypothèses comme par exemple l'ordre de restauration des failles et facilite la compréhension de l'histoire du sous-sol.

Nous avons choisi une formulation modulaire des différentes contraintes, ce qui permet de pouvoir aisément supprimer, modifier ou ajouter de nouvelles contraintes ou bien encore modifier (pondérer) leur importance. De ce fait, dans le futur notre outil pourra facilement être enrichi ou adapté à de nouvelles applications. Par exemple, proposer plusieurs modes de déformations (avec différentes contraintes internes) semble être une première piste qui pourrait être envisagée rapidement, et est actuellement discutée à TOTAL.

Sur une station de travail standard, notre outil présente des performances qui permettent de l'utiliser en temps quasi-réel, ce qui était une des contraintes que nous nous étions fixées comme objectif pour nos travaux : quelques secondes suffisent pour effectuer une opération de restauration, ce qui est acceptable pour l'utilisateur (sachant que ce temps diminuera rapidement dans les prochaines années avec l'évolution constante de la rapidité des ordinateurs). Néanmoins, nous avons remarqué que l'utilisation de maillages denses, au delà de plusieurs milliers de points, peut rapidement dégrader la performance. Ce ralentissement provient du processus d'optimisation itératif, qui doit manipuler des matrices dont le nombre d'éléments augmente quadratiquement en fonction du nombre de points. Toutefois, deux pistes que je n'ai pas eu le temps d'explorer durant mon travail se dessinent pour améliorer ce point : le calcul matriciel peut se paralléliser (en particulier, les implémentations sur les puces GPU des cartes graphiques accélératrices produisent d'excellentes performances avec leurs bibliothèques matricielles) et il peut être largement optimisé quand les matrices qui interviennent sont des matrices creuses, ce qui est majoritairement le cas dans nos algorithmes.

Enfin, dans cette thèse nous nous sommes limités au cas 2D, c'est-à-dire à la restauration d'images sismiques, ce qui est pertinent en phase préliminaire d'interprétation sismique et géologique. On pourrait toutefois envisager une extension 3D dans le futur, sachant que nous avons développé notre outil en tâchant de le garder aussi général que possible et en ayant une telle extension future en

tête. Il serait par exemple possible de remplacer la triangulation utilisée par un maillage tétraédrique, conforme aux horizons et aux failles décrites par des surfaces car les différentes contraintes peuvent se formuler en 3D et les interpolations barycentriques linéaires de maillages volumiques sont des techniques qui existent. Toutefois, la difficulté de l'introduction d'une dimension à une application de restauration réside dans la nécessité de travailler avec un modèle structurel parfaitement segmenté. Si nous imaginons par exemple le glissement d'un bloc le long de plusieurs failles, la description des contacts faille/faille et horizon/faille devra être parfaite pour garantir l'étanchéité de chaque volume du modèle et assurer un bon fonctionnement de l'algorithme. Or ce n'est pas le cas en pratique lors d'une phase préliminaire d'interprétation, durant laquelle les surfaces pointées sont généralement grossières vu qu'il s'agit d'une phase préliminaire d'analyse. De ce fait une telle perspective requerra d'introduire en amont un outil de modélisation assurant la qualité des surfaces à restaurer.



# Bibliographie

- [ATC<sup>+</sup>08] O.K.C. Au, C.L. Tai, H.K. Chu, D. Cohen-Or, and T.Y. Lee. Skeleton extraction by mesh contraction. In *ACM Transactions on Graphics (TOG)*, volume 27, page 44. ACM, 2008.
- [Bar84] Alan H. Barr. Global and local deformations of solid primitives. *SIGGRAPH Comput. Graph.*, 18(3) :21–30, January 1984.
- [Bar85] David Barr. 3-d palinspastic restoration of normal faults in the inner moray firth : implications for extensional basin development. *Earth and Planetary Science Letters*, 75(2) :191–203, 1985.
- [Bax92] Bradley John Charles Baxter. The interpolation theory of radial basis functions, 1992.
- [BHZN10] Péter Borosán, Reid Howard, Shaoting Zhang, and Andrew Nealen. Hybrid mesh editing. In *Proc. of Eurographics*, 2010.
- [Bjo96] Ake Bjorck. *Numerical methods for least squares problems*. Number 51. Society for Industrial and Applied Mathematics, 1996.
- [BK05] M. Botsch and L. Kobbelt. Real-time shape editing using radial basis functions. In *Computer graphics forum*, volume 24, pages 611–621. Wiley Online Library, 2005.
- [Bro92] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM computing surveys (CSUR)*, 24(4) :325–376, 1992.
- [BS07] Alexander I Bobenko and Boris A Springborn. A discrete laplace–beltrami operator for simplicial surfaces. *Discrete & Computational Geometry*, 38(4) :740–756, 2007.
- [CAA10] Robert Carroll, Aseem Agarwala, and Maneesh Agrawala. Image warps for artistic perspective manipulation. *ACM Transactions on Graphics (TOG)*, 29(4) :127, 2010.
- [CBC<sup>+</sup>01] Jonathan C Carr, Richard K Beatson, Jon B Cherrie, Tim J Mitchell, W Richard Fright, Bruce C McCallum, and Tim R Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76. ACM, 2001.
- [CCCJ04] Robert N Clark, David Cox, and Howard C Curtiss Jr. *A modern course in aeroelasticity*, volume 116. Springer Science+ Business Media, 2004.
- [CF98] TA Clarke and JG Fryer. The development of camera calibration methods and models. *The Photogrammetric Record*, 16(91) :51–66, 1998.

- [Cha10] Rollin T Chamberlin. The appalachian folds of central pennsylvania. *The Journal of Geology*, 18(3) :228–251, 1910.
- [Cia88] Philippe G Ciarlet. *Introduction à l'analyse numérique matricielle et à l'optimisation*. 1988.
- [Coq90] S. Coquillart. Extended free-form deformation : a sculpturing tool for 3D geometric modeling. In *SIGGRAPH '90 : Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 187–196, New York, NY, USA, 1990. ACM.
- [CR94] Yu-Kuang Chang and Alyn P. Rockwood. A generalized de casteljau approach to 3d free-form deformation. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, pages 257–260. ACM, 1994.
- [DAAM07] B Dunkin, GL Adrales, K Apelgren, and JD Mellinger. Surgical simulation : a current review. *Surgical endoscopy*, 21(3) :357–366, 2007.
- [Dah69] CDA Dahlstrom. Balanced cross sections. *Canadian Journal of Earth Sciences*, 6(4) :743–757, 1969.
- [DM06] Tony DeRose and Mark Meyer. Harmonic coordinates. In *Pixar Technical Memo 06-02, Pixar Animation Studios*. Citeseer, 2006.
- [DN07] Zhigang Deng and Junyong Noh. Computer facial animation : A survey. In *Data-Driven 3D Facial Animation*, pages 1–28. Springer, 2007.
- [Doy07] Philippe Doyen. *Seismic reservoir characterization : An earth modelling perspective*, volume 2. EAGE publications, 2007.
- [DRCM10] Pauline Durand-Riard, Guillaume Caumon, and Pierre Muron. Balanced restoration of geological volumes with relaxed meshing constraints. *Comput. Geosci.*, 36(4) :441–452, April 2010.
- [DRSF<sup>+</sup>11] Pauline Durand-Riard, Lise Salles, Mary Ford, Guillaume Caumon, and Jeanne Pellerin. Understanding the evolution of syn-depositional folds : Coupling decompaction and 3d sequential restoration. *Marine and Petroleum Geology*, 28(8) :1530–1539, 2011.
- [DSCM02] Márcio Rodrigues De Santi, João Luiz Elias Campos, and Luiz Fernando Martha. A finite element approach for geological section reconstruction. In *Proceedings of the 22th Gocad Meeting, Nancy, France*, pages 1–13, 2002.
- [DSCM03] Márcio Rodrigues De Santi, João Luiz Elias Campos, and Luiz Fernando Martha. 3-d geological restoration using a finite element approach. In *Gocad Proceedings : 23th Gocad Meeting, Association Scientifique pour la Geologie et ses Applications*, page 1, 2003.
- [ED08] Nikolaos Ersotelos and Feng Dong. Building highly realistic facial modeling and animation : a survey. *The Visual Computer*, 24(1) :13–30, 2008.
- [Flo03] Michael S. Floater. Mean value coordinates. *Comput. Aided Geom. Des.*, 20(1) :19–27, March 2003.

- [FOKGM07] Sven Forstmann, Jun Ohya, Artus Krohn-Grimberghe, and Ryan McDougall. Deformation styles for spline-based skeletal animation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 141–150. Eurographics Association, 2007.
- [Fra82] Richard Franke. Scattered data interpolation : tests of some methods. *Mathematics of computation*, 38(157) :181–200, 1982.
- [GG93] Jean-Pierre Gratier and Bertrand Guillier. Compatibility constraints on folded and faulted strata and calculation of total displacement using computational restoration (unfold program). *Journal of structural geology*, 15(3) :391–402, 1993.
- [GGDO91] Jean-Pierre Gratier, Bertrand Guillier, Alain Delorme, and Francis Odonne. Restoration and balance of a folded and faulted surface by best-fitting of finite elements : Principle and applications. *Journal of Structural Geology*, 13(1) :111–115, 1991.
- [GGJK13] Mathieu Gilardet, Sebastien Guillon, Bruno Jobard, and Dimitri Komatitsch. Seismic image restoration using nonlinear least squares shape optimization. *Procedia Computer Science*, 18 :732–741, 2013.
- [Gib83] AD Gibbs. Balanced cross-section construction from seismic sections in areas of extensional tectonics. *Journal of Structural Geology*, 5(2) :153–160, 1983.
- [GJS<sup>+</sup>02] Paul Griffiths, Serena Jones, Neil Salter, Frauke Schaefer, Robert Osfield, and Herbert Reiser. A new technique for 3-d flexural-slip restoration. *Journal of Structural Geology*, 24(4) :773–782, 2002.
- [GL05] Philippe Girodroux-Lavigne. Progress in steady/unsteady fluid-structure coupling with navier-stokes equations. In *Int. Forum on Aeroelasticity and Structural Dynamics, Munich, Germany*, 2005.
- [GM97] S.F.F. Gibson and B. Mirtich. A survey of deformable modeling in computer graphics. *Technical Report TR-97-1*, 9, 1997.
- [GM98] C. A. Glasbey and K. V. Mardia. A review of image-warping methods. *Journal of Applied Statistics*, 25(2) :155–171, 1998.
- [Gui91] Bertrand Guillier. *Dépliage automatique de strates plissées et faillées : application à l'équilibrage de structures naturelles*. PhD thesis, Université Joseph-Fourier-Grenoble I, 1991.
- [HBK01] M. Hisada, A.G. Belyaev, and T.L. Kunii. A 3d voronoi-based skeleton and associated surface features. In *Computer Graphics and Applications, 2001. Proceedings. Ninth Pacific Conference on*, pages 89–96. IEEE, 2001.
- [Hes07] Roland Hess. *The essential Blender : guide to 3D creation with the open source suite Blender*. No Starch Press, 2007.
- [Hos79] John R Hossack. The use of balanced cross-sections in the calculation of orogenic contraction : A review. *Journal of the Geological Society*, 136(6) :705–711, 1979.

- [HSL<sup>+</sup>06] Jin Huang, Xiaohan Shi, Xinguo Liu, Kun Zhou, Li-Yi Wei, Shang-Hua Teng, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Subspace gradient domain mesh deformation. *ACM Trans. Graph.*, 25(3) :1126–1134, July 2006.
- [JC76] JC Jaeger and NG Cook. *Fundamentals of rock mechanics*, 1976.
- [JMD<sup>+</sup>07] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki. Harmonic coordinates for character articulation. In *ACM Transactions on Graphics (TOG)*, volume 26, page 71. ACM, 2007.
- [JSW05] Tao Ju, Scott Schaefer, and Joe Warren. Mean value coordinates for closed triangular meshes. *ACM Trans. Graph.*, 24(3) :561–566, July 2005.
- [KCOZ06] Ladislav Kavan, Steven Collins, Carol O’Sullivan, and Jiri Zara. Dual quaternions for rigid transformation blending. *Technical report*, 2006.
- [KI03] Taejung Kim and Yong-Jo Im. Automatic satellite image registration by combination of matching and random sample consensus. *Geoscience and Remote Sensing, IEEE Transactions on*, 41(5) :1111–1117, 2003.
- [Kom88] K. Komatsu. Human skin model capable of natural shape variation. *The visual computer*, 3(5) :265–271, 1988.
- [KSSH02] Nikita Kojekine, Vladimir Savchenko, Mikhail Senin, and Ichiro Hagiwara. Real-time 3d deformations by means of compactly supported radial basis functions. In *In Short papers proceedings of Eurographics*, pages 35–43, 2002.
- [KŽ05] Ladislav Kavan and Jiří Žára. Spherical blend skinning : a real-time deformation of articulated models. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pages 9–16. ACM, 2005.
- [LCF00] John P Lewis, Matt Cordner, and Nickson Fong. Pose space deformation : a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 165–172. ACM Press/Addison-Wesley Publishing Co., 2000.
- [LH11] Jin Li and Andrew D Heap. A review of comparative studies of spatial interpolation methods in environmental sciences : Performance and impact factors. *Ecological Informatics*, 6(3) :228–241, 2011.
- [LLCO08] Y. Lipman, D. Levin, and D. Cohen-Or. Green coordinates. In *ACM Transactions on Graphics (TOG)*, volume 27, page 78. ACM, 2008.
- [LR10] Colin Levy and Ton Roosendaal. Sintel. In *ACM SIGGRAPH ASIA 2010 Computer Animation Festival*, page 82. ACM, 2010.
- [LSCO<sup>+</sup>04] Y. Lipman, O. Sorkine, D. Cohen-Or, C. Rössl, and H. P. Seidel. Differential coordinates for interactive mesh editing. In *Proceedings of the Shape Modeling International 2004, SMI ’04*, pages 181–190, Washington, DC, USA, 2004. IEEE Computer Society.

- [LTGM97] Michel Léger, Muriel Thibaut, Jean-Pierre Gratier, and Jean-Marie Morvan. A least-squares method for multisurface unfolding. *Journal of structural geology*, 19(5) :735–743, 1997.
- [LZY10] Yong-Jin Liu, Dong-Liang Zhang, and Matthew Ming-Fai Yuen. A survey on cad methods in 3d garment design. *Computers in Industry*, 61(6) :576–593, 2010.
- [MBLD02] Mark Meyer, Alan Barr, Haeyoung Lee, and Mathieu Desbrun. Generalized barycentric coordinates on irregular polygons. *Journal of graphics tools*, 7(1) :13–22, 2002.
- [MDS<sup>+</sup>02] Mark Meyer, Mathieu Desbrun, Peter Schröder, Alan H Barr, et al. Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization and mathematics*, 3(2) :52–58, 2002.
- [MHTG05] Matthias Müller, Bruno Heidelberger, Matthias Teschner, and Markus Gross. Meshless deformations based on shape matching. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 471–478. ACM, 2005.
- [MJ96] Ron MacCracken and Kenneth I. Joy. Free-form deformations with lattices of arbitrary topology. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 181–188, New York, NY, USA, 1996. ACM.
- [MLG06] I. Moretti, F. Lepage, and M. Guiton. Kine3d : a new 3d restoration method based on a mixed approach linking geometry, 2006.
- [MM06] L. Maerten and F. Maerten. Chronologic modeling of faulted and fractured reservoirs using geomechanically based restoration : Technique and industry applications. *AAPG bulletin*, 90(8) :1201–1226, 2006.
- [MR90] I Moretti and JJ Raoult. Geological restoration of seismic depth images. *First Break*, 8(7), 1990.
- [MTE90] I Moretti, S Triboulet, and L Endignoux. Some remarks on the geometrical modeling of geological deformations. *J. Letouzey.—Paris : Technip editions*, pages 155–162, 1990.
- [MtLTM88] Nadia Magnenat-thalmann, Richard Laperrire, Daniel Thalmann, and Université De Montréal. Joint-dependent local deformations for hand animation and object grasping. In *In Proceedings on Graphics interface '88*, pages 26–33, 1988.
- [Mur05] Pierre Muron. Méthodes numériques 3d de restauration des structures géologiques failées. *Th. Doct. Institut National Polytechnique de Lorraine. Nancy, France*, 30, 2005.
- [MV98] JB Antoine Maintz and Max A Viergever. An overview of medical image registration methods. *UU-CS*, (1998-22), 1998.
- [NMK<sup>+</sup>06] Andrew Nealen, Matthias Mueller, Richard Keiser, Eddy Boxerman, and Mark Carlson. Physically Based Deformable Models in Computer Graphics. *Computer Graphics Forum*, 25(4) :809–836, December 2006.
- [Pie91] Les Piegl. On nurbs : a survey. *IEEE Computer Graphics and Applications*, 11(1) :55–71, 1991.

- [PTS09] Simon Pabst, Bernhard Thomaszewski, and Wolfgang Straßer. Anisotropic friction for deformable surfaces and solids. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 149–154. ACM, 2009.
- [RK89] Mark G Rowan and Roy Kligfield. Cross section restoration and balancing as aid to seismic interpretation in extensional terranes. *AAPG bulletin*, 73(8) :955–966, 1989.
- [RXS00] D. Rouby, H. Xiao, and J. Suppe. 3-d restoration of complexly folded and faulted surfaces using multiple unfolding mechanisms. *AAPG bulletin*, 84(6) :805–829, 2000.
- [SCOL<sup>+</sup>04] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, SGP '04, pages 175–184, New York, NY, USA, 2004. ACM.
- [She68] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524. ACM, 1968.
- [SK04] A. Sheffer and V. Kraevoy. Pyramid coordinates for morphing and deformation. In *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, pages 68–75. IEEE, 2004.
- [Sor06] Olga Sorkine. *Laplacian mesh processing*. PhD thesis, Tel Aviv University, 2006.
- [SP86] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.*, 20(4) :151–160, August 1986.
- [TO99] Greg Turk and James F O’Brien. Variational implicit surfaces. *Technical Report GIT-GVU-99-15*, 1999.
- [TS11] Shaun Shi Yan Tan and Sudip K Sarker. Simulation in surgery : a review. *Scottish Medical Journal*, 56(2) :104–109, 2011.
- [Und83] Philip Underwood. Dynamic relaxation(in structural transient analysis). *Computational methods for transient analysis(A 84-29160 12-64)*. Amsterdam, North-Holland, 1983,, pages 245–265, 1983.
- [WJM86] NJ White, JA Jackson, and DP McKenzie. The relationship between the geometry of normal faults and that of the sedimentary layers in their hanging walls. *Journal of Structural Geology*, 8(8) :897–909, 1986.
- [WKBR97] GD Williams, SJ Kane, TS Buddin, and AJ Richards. Restoration and balance of complex folded and faulted rock volumes : flexural flattening, jigsaw fitting and decompaction in three dimensions. *Tectonophysics*, 273(3) :203–218, 1997.
- [WXW<sup>+</sup>06] Yanlin Weng, Weiwei Xu, Yanchen Wu, Kun Zhou, and Baining Guo. 2d shape deformation using nonlinear least squares optimization. *Vis. Comput.*, 22(9) :653–660, September 2006.
- [ZF03] Barbara Zitova and Jan Flusser. Image registration methods : a survey. *Image and vision computing*, 21(11) :977–1000, 2003.

## Résumé

Nous présentons une nouvelle méthode pour la restauration d'images sismiques. Quand on l'observe, une image sismique est le résultat d'un système de dépôt initial qui a été transformé par un ensemble de déformations géologiques successives (flexions, glissement de la faille, etc) qui se sont produites sur une grande période de temps. L'objectif de la restauration sismique consiste à inverser les déformations pour fournir une image résultante qui représente le système de dépôt géologique tel qu'il était dans un état antérieur. Classiquement, ce procédé permet de tester la cohérence des hypothèses d'interprétations formulées par les géophysiciens sur les images initiales.

Dans notre contribution, nous fournissons un outil qui permet de générer rapidement des images restaurées et qui aide donc les géophysiciens à reconnaître et identifier les caractéristiques géologiques qui peuvent être très fortement modifiées et donc difficilement identifiables dans l'image observée d'origine. Cette application permet alors d'assister ces géophysiciens pour la formulation d'hypothèses d'interprétation des images sismiques.

L'approche que nous introduisons est basée sur un processus de minimisation qui exprime les déformations géologiques en termes de contraintes géométriques. Nous utilisons une approche itérative de *Gauss-Newton* qui converge rapidement pour résoudre le système. Dans une deuxième partie de notre travail nous montrons différents résultats obtenus dans des cas concrets afin d'illustrer le processus de restauration d'image sismique sur des données réelles et de montrer comment la version restaurée peut être utilisée dans un cadre d'interprétation géologique.



## Abstract

We present a new method for seismic image restoration. When observed, a seismic image is the result of an initial deposit system that has been transformed by a set of successive geological deformations (folding, fault slip, etc) that occurred over a large period of time. The goal of seismic restoration consists in inverting the deformations to provide a resulting image that depicts the geological deposit system as it was in a previous state.

With our contribution, providing a tool that quickly generates restored images helps the geophysicists to recognize geological features that may be too strongly altered in the observed image. The proposed approach is based on a minimization process that expresses geological deformations in terms of geometrical constraints.

We use a quickly-converging *Gauss-Newton* approach to solve the system. We provide results to illustrate the seismic image restoration process on real data and present how the restored version can be used in a geological interpretation framework.