

# Méthode d'interpolation rapide pour les surfaces avec des failles par optimisation multi-échelles avec la dérivée seconde

Oil & Gas Science and Technology - Rev. IFP Energies nouvelles 75, 62 (2020)

Michel Léger, Vincent Clochard

(traduction post-éditée par N. Bacaër, suggestions d'amélioration : nicolas.bacaer@ird.fr)



## Résumé

Nous présentons une méthode d'interpolation de surface lisse permettant de prendre en compte des discontinuités (par exemple des failles) qui peut être appliquée à tout jeu de données défini sur un maillage régulier. Nous utilisons une minimisation multi-échelles, qui utilise la dérivée seconde, basée sur une méthode de gradient conjugué. Notre approche multi-échelles permet à l'algorithme de traiter des millions de points en quelques secondes sur un poste de travail monobloc. La surface interpolée est continue, ainsi que sa première dérivée, sauf sur certaines lignes qui ont été spécifiées comme discontinuités. Les applications en géosciences sont nombreuses, par exemple lorsqu'un modèle structural doit être construit à partir de points choisis sur des données sismiques. Le creux d'interpolation qui en résulte étend le creux des données d'entrée. L'algorithme fonctionne également si les failles sont données par des lignes brisées. Nous présentons des résultats à partir d'exemples synthétiques et réels prenant en compte le réseau de failles.

## 1. Introduction

Le repérage de surfaces, d'isochrones ou de failles, est difficile et incertain en raison du faible rapport signal sur bruit de la sismique de surface ou d'autres raisons, de sorte que le plus souvent, seul un repérage partiel peut être réalisé. Cependant, pour certaines applications géoscientifiques (Schneider, 2002 ; Perrin et Rainaud, 2013), un repérage complet est requis dans ces zones. Un interpolateur est donc nécessaire avec les caractéristiques suivantes: il doit être rapide, capable de traiter des millions de points de données, adaptable à toutes sortes de jeux de données (Fig.1), et il doit préserver le creux des données (le cas échéant). De plus, l'interpolateur doit également tenir compte des failles.

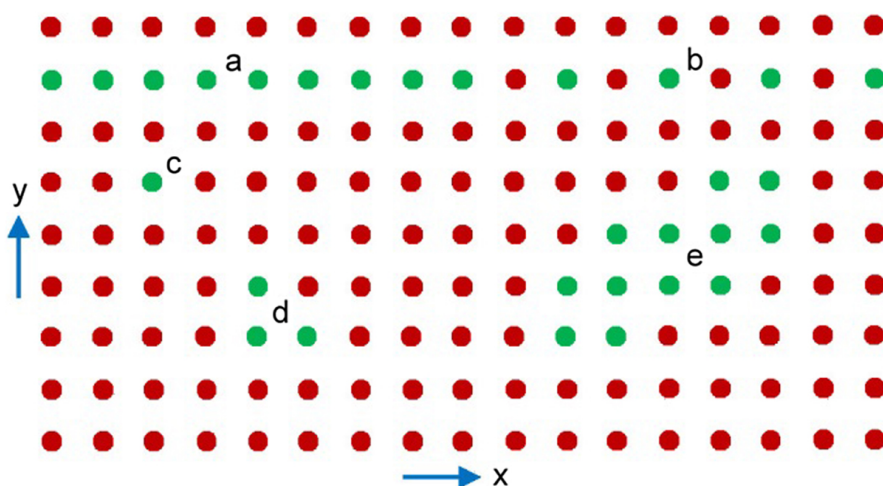


Fig. 1 Tout ensemble de données peut être interpolé sur une grille 2D régulière (tous les points verts et rouges), en tenant compte de la distribution aléatoire des valeurs données aux points verts. (a) Ces points verts pourraient représenter des lignes, (b) des lignes pointillées, (c) des points uniques, (d) de petits groupes de points, (e) de grands ensembles de données.

De nombreuses techniques d'interpolation (Awange et coll., 2018) comme les interpolations polynomiales (Léger, 1999) ne sont pas compatibles avec les exigences ci-dessus, les interpolations de splines sont également actuellement utilisées dans le cadre de la conception assistée par ordinateur (CAO) pour définir la représentation d'objets et effectuer diverses manipulations de surfaces. Néanmoins, ces représentations nécessitent des conditions obligatoires telles que le contrôle de forme global et local à l'aide du point de contrôle d'une surface. De plus, pour assurer une continuité à la jonction de deux patches, il est nécessaire de définir un facteur de qualité de lissage autour des points de contrôle, généralement du premier

ordre avec tangente et second ordre avec courbure. Une autre difficulté est également de définir des points de contrôle en dehors de la surface pour limiter les effets de bord, de sorte que les points de contrôle ne soient pas situés sur la surface.

Nous avons principalement deux types de modèles appelés surfaces de Bézier et B-spline (Bézier, 1977, 1987), les deux sont utilisés selon l'application choisie. Il existe également deux implémentations de ces modèles rationnelle ou non rationnelle. Dans ce dernier cas, un point de contrôle d'un modèle de Bézier a une influence globale et avec B-spline ce n'est qu'une influence locale. Avec l'implémentation rationnelle, on parle de B-Spline Rationnelle Non Uniforme (NURBS), qui consiste à associer à chaque point de contrôle un facteur de pondération pour donner plus ou moins d'influence dans la représentation (Rogers et Earnshaw, 1991). Un autre aspect est dû au degré d'une surface de Bézier rationnelle définie par le nombre de points de contrôle moins un, et au degré d'une surface B-spline qui pourrait être choisie, faible pour avoir plus de contrôle local sur la surface. On peut penser à l'utilisateur des difficultés à choisir les meilleurs paramètres pour construire une surface NURBS et à éviter les effets de bord pour interpoler une surface avec des discontinuités.

Une autre approche est basée sur une optimisation locale d'une surface paramétrique (Hjelle et Dæhlen, 2005). C'est une approximation locale des moindres carrés avec un terme de régularisation. Ils ont utilisé des triangulations hiérarchiques affinées de manière itérative si les données sont irrégulières, généralement des cartes de courbes de niveau. Plus précisément, le processus commence par un maillage carré régulier et en utilisant chaque centre carré, il fait quatre triangles. Ensuite, en considérant le milieu du bord le plus long, il forme deux triangles au lieu d'un. Ce type de dichotomie ne se produit que si les données sont suffisamment nombreuses et que le processus est itératif, de sorte qu'elles fonctionnent avec un pas spatial variable. La principale difficulté de la méthode est de pondérer le terme de lissage. Une autre méthode utilise une combinaison de fonctions de base radiale (RBF) et de déplacement des moindres carrés (MLS) dans Ohtake et coll.(2004). D'autres auteurs ont utilisé des bases hiérarchiques utilisant des B-splines et des splines- (pré) ondelettes avec un paramètre de régularisation (Castaño et coll., 2009). A chaque pas, une approximation des moindres carrés des données est calculée, ce qui est assez efficace pour approcher une grande quantité de points de données dispersés (jusqu'à 300 000). Plus récemment, une énorme quantité de données provient de la détection et de la télémétrie de lumière (LiDAR) pour la dérivation de modèles numériques de terrain (DTM), une méthode d'interpolation de la fonction de base radiale à prise en charge compacte (CSRBF) des moindres carrés est assez efficace et quatre fois plus rapide que krigeage (Chen et coll., 2018a,2018b).

Nous présentons ici un interpolateur multi-échelles rapide capable de travailler avec un choix très limité de paramètres d'interpolation et également de prendre en compte d'éventuelles discontinuités. Les principales différences de notre interpolation avec les derniers travaux cités sont l'utilisation d'un pas spatial constant à chaque itération et également l'utilisation d'une seule fonction de coût définie globalement. On peut noter que notre fonction de coût a un seul minimum dès que les données ont au moins trois points (non alignés).

Dans ce qui suit, l'interpolateur ne considère que les failles verticales mais son extension aux failles inclinées est simple (Léger, 2016).

Cet interpolateur repose sur la minimisation de la seconde dérivée, en utilisant un gradient conjugué (Hestenes et Stiefel, 1952). Cela conduit à une bonne solution mais le coût de calcul est trop important car de grandes régions inconnues nécessitent un nombre élevé d'itérations. Pour obtenir une bonne convergence pour un grand nombre d'inconnues avec un petit nombre d'itérations, un bon modèle initial est nécessaire. Ceci est réalisé en décimant l'ensemble de points de données avec un double pas dans  $X$  et un double pas  $Y$ (temps de calcul divisé par 4) et le nombre d'itérations étant doublé. Dans l'ensemble, le temps de calcul est divisé par 2. La réitération de cette décimation répond à la question de l'obtention du modèle initial. La réitération s'arrête lorsque le nombre d'itérations est supérieur au nombre d'inconnues, la minimisation de la fonction de coût étant ainsi complète. Cette explication va de la grille fine aux grilles plus grossières, mais le calcul va des grilles plus grossières à la grille fine.

Les défauts sont introduits, *via* des lignes discontinues, le long desquelles le calcul de toutes les dérivées les traversant est exclu. Cette astuce très simple permet l'interpolation de surfaces présentant des discontinuités telles que des failles, ce qui est une grande amélioration d'un point de vue géologique. Le temps de calcul reste le même, quels que soient les défauts présents ou non.

Le point clé de cette approche est que 100 itérations sur le modèle final sont suffisantes pour converger en cas de modèles simples ou moyennement complexes, quelle que soit la taille du modèle, et quels que soient les défauts présents ou non.

Dans la section 2, nous expliquons la méthode. Dans la section 3, nous illustrons la méthode sur différents exemples synthétiques sans défauts. La section 4 est consacrée aux exemples synthétiques avec des défauts. Un exemple concret est présenté dans la section 5 .

## 2 Méthode

### 2.1 Principe

On considère un maillage régulier rectangulaire et les coordonnées entières  $(i, j)$  discrétiser les coordonnées  $X$  et  $Y$ . L'étape  $i$  et  $j$  sont des unités, mais la distance de trace en mètres est donnée au programme pour gérer anisotropies possibles entre  $X$  et  $Y$ . Les points de données se trouvent sur le maillage, et les points inconnus également. Pour interpoler certains points donnés, nous avons choisi de calculer une surface passant par tous les points donnés et de minimiser sa seconde dérivée avec la méthode du gradient conjugué, qui est simple à mettre en œuvre. Cependant pour certains jeux de données, le nombre d'itérations peut être proche du nombre de points inconnus, ce qui rend l'algorithme  $O(N^2)$ , ce qui est trop cher pour de nombreuses applications. Le but de cet article est de supprimer cet inconvénient.

Une solution pour obtenir une optimisation rapide est d'initialiser l'optimisation du modèle par un modèle très proche de la solution. Mais comment obtenir un tel modèle initial? La solution repose sur le fait que l'optimisation d'un modèle avec un nombre de  $i$  et  $j$  deux fois inférieur à celui d'origine, et le doublement du nombre d'itérations rend globalement deux fois moins de calculs (Jespersen, 1984). Ainsi, comment obtenir un nouveau bon modèle initial? Répétez simplement! Cette réitération s'arrête si le gradient conjugué converge en un nombre d'itérations supérieur au nombre d'inconnues. Cela est vrai parce que la deuxième dérivée est une fonction linéaire des inconnues, donc la fonction de coût est une quadratique (voir Réf. Méthode du gradient conjugué). Dans ce cas, c'est la première inversion, et le résultat sera indépendant du modèle initial. En pratique, la forme générale est obtenue sur les grilles grossières, en particulier la première, puis les détails de plus en plus fins sur d'autres grilles.

Cette stratégie multi-échelles fait de l'algorithme  $O(N)$  au lieu de  $O(N^2)$ . La complexité de la mise en page des données, et en particulier du réseau de pannes, augmente le coût, c'est-à-dire que nous obtenons  $O(aN)$  avec  $a$  plus grand. Par exemple,  $a = 100$  est une valeur adéquate pour des données simples, mais  $a = 1000$  ou plus pourrait être nécessaire pour des données plus complexes. Une fois le jeu de données et les défauts donnés, le coût est vraiment en  $O(N)$  en cas de raffinement du maillage.

## 2.2 Mesure de la deuxième dérivée

A partir d'un maillage régulier de points, il est facile de construire une surface  $C^2$  polynomiale par morceaux, tels que les splines. *Via* cette possibilité, nous utiliserons également la pente et la dérivée seconde. Néanmoins, ces dérivées première et seconde (voir section 2.7) sont spécifiques à notre modèle et dépendent de ses caractéristiques.

Afin de calculer les trois types de dérivées secondes, nous utilisons simplement les formules suivantes d'une fonction  $f$  (voir Réf. Différence finie). Avec nos paramètres de convention, l'expression de la dérivée seconde  $d_{ii}$  en  $i$ , où  $h_i$  est le pas en  $i$  et  $(i, j)$  est la position du point sur le maillage régulier est donnée par (1) :

$$h_i^2 d_{ii} [f(i, j)] = f(i + 1, j) - 2f(i, j) + f(i - 1, j). \quad (1)$$

L'expression de la deuxième dérivée  $d_{jj}$  en  $j$ , où  $h_j$  est le pas de  $j$  et  $(i, j)$  est la position du point sur le maillage régulier est donnée par (2) :

$$h_j^2 d_{jj} [f(i, j)] = f(i, j + 1) - 2f(i, j) + f(i, j - 1). \quad (2)$$

L'expression de la deuxième dérivée  $d_{ij}$  en  $i$  et  $j$  pourrait être choisie comme on veut. L'équation la plus populaire est donnée par (3), complètement symétrique et facile à différencier, mais pas très compacte:

$$4h_i h_j d_{ij} [f(i, j)] = f(i + 1, j + 1) - f(i + 1, j - 1) - f(i - 1, j + 1) + f(i - 1, j - 1). \quad (3)$$

Une autre possibilité est moins symétrique et mélange deux schémas, l'un positif en  $i$  et  $j$ , l'autre négatif en  $i$  et  $j$ , mais les sept termes sont un inconvénient pour la différenciation (4) :

$$2h_i h_j d_{ij} [f(i, j)] = f(i + 1, j + 1) - f(i + 1, j) - f(i, j + 1) + 2f(i, j) - f(i - 1, j) - f(i, j - 1) + f(i - 1, j - 1). \quad (4)$$

Nous avons choisi pour le terme croisé  $d_{ij}$  une solution non symétrique (5), avec le schéma «positif en  $i$  et  $j$ », car il est compact et facile à différencier:

$$h_i h_j d_{ij} [f(i, j)] = f(i, j) + f(i + 1, j + 1) - f(i + 1, j) - f(i, j + 1), \quad (5)$$

quel que soit le schéma, ce terme croisé doit être pris en compte, car la figure 2 (partie inférieure) montre son influence.

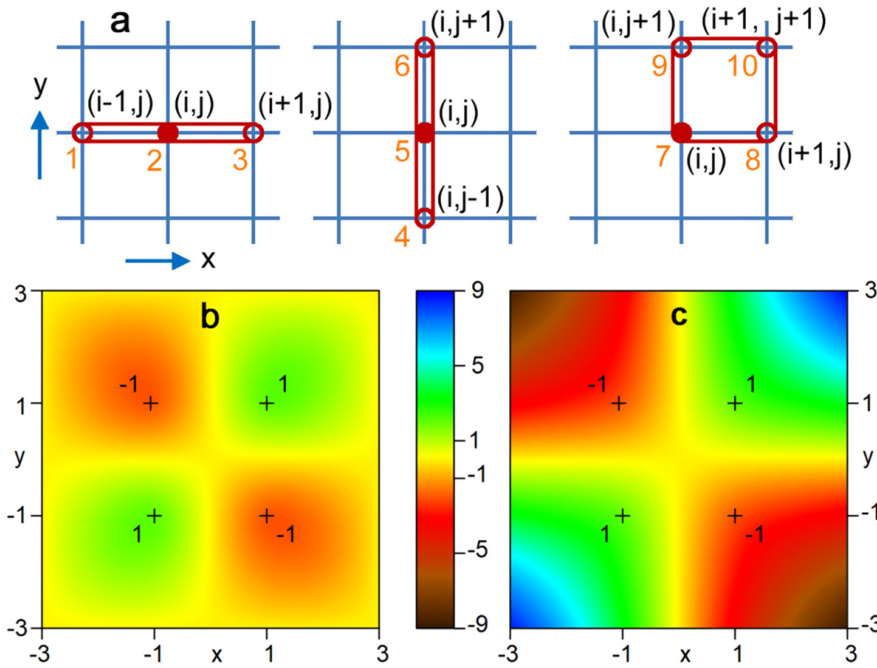


Fig. 2. La partie supérieure (a) illustre les trois secondes dérivées d'une fonction au point rouge rempli  $(i, j)$ . Les nombres en orange représentent les composantes du gradient, pour le point  $(i, j)$ . La partie inférieure montre l'influence de la deuxième dérivée (en utilisant notre schéma à quatre termes) dans le résultat de l'interpolation: (b) avec et (c) sans, les deux interpolations utilisant les mêmes données initiales (valeur de  $+1/-1$  au noir des croix). On voit sur (b) un résultat très lisse par l'arête, alors que les coins sont très éloignés des données en (c). La surface réglée affichée en (c) est un hyperboloïde.

## 2.3 Fonction de coût

La fonction de coût  $Q$  est la norme au carré  $L^2$  des dérivées secondes:

$$2Q = Q_{II} + 2Q_{IJ} + Q_{JJ}, \quad (6)$$

avec,

$$Q_{II} = \sum_{N_{II}} (h_{ii}^2 d_{ii} [f(i, j)])^2, \quad (7)$$

où  $N_{II}$  est l'ensemble des points avec au moins une inconnue dans le triplet de  $i$ ,

$$Q_{IJ} = \sum_{N_{IJ}} (h_i h_j d_{ij} [f(i, j)])^2, \quad (8)$$

où  $N_{IJ}$  est l'ensemble des points avec au moins une inconnue dans le quadruplet, et,

$$Q_{JJ} = \sum_{N_{JJ}} (h_{jj}^2 d_{jj} [f(i, j)])^2, \quad (9)$$

où  $N_{JJ}$  est l'ensemble des points avec au moins une inconnue dans le triplet de  $j$ . Les ensembles  $N_{II}$ ,  $N_{IJ}$  et  $N_{JJ}$  sont généralement quelque peu différents. Ces fonctions de coût sont calculées sur plusieurs grilles, à chaque échelle.

La fonction de coût  $Q$  est constituée d'un seul critère, le résultat sera donc indépendant de tout facteur multiplicatif global tel que les incertitudes (Foster et Evans, 2008).

Puisque la fonction de coût en un point donné dépend de quelques points, le gradient est calculé par différences finies.

## 2.4 Défauts

Les défauts correspondent à des discontinuités définies le long de lignes discontinues. La forme de ces failles est définie au départ, quelle que soit la forme de la surface qui évoluera lors de l'optimisation. En d'autres termes, les failles seront verticales et le même modèle de failles sera utilisé à toutes les échelles. Ces défauts sont utilisés tels quels, sans aucune extrapolation ce qui signifie que l'algorithme introduira des discontinuités exactement là où le client l'a spécifié.

Le principe de la gestion des défauts est de supprimer la minimisation des dérivées secondes traversant les défauts. La figure 3 montre une région où tous les points sont inconnus. Les triplets affichés en pointillé rouge ou bleu montrent les dérivées secondes qui sont supprimées dans la fonction de coût, en raison de la présence du défaut en vert. Ce faisant, les deux côtés du défaut se déconnectent. Cette astuce fonctionne pour n'importe quelle échelle dans l'inversion multi-échelle.

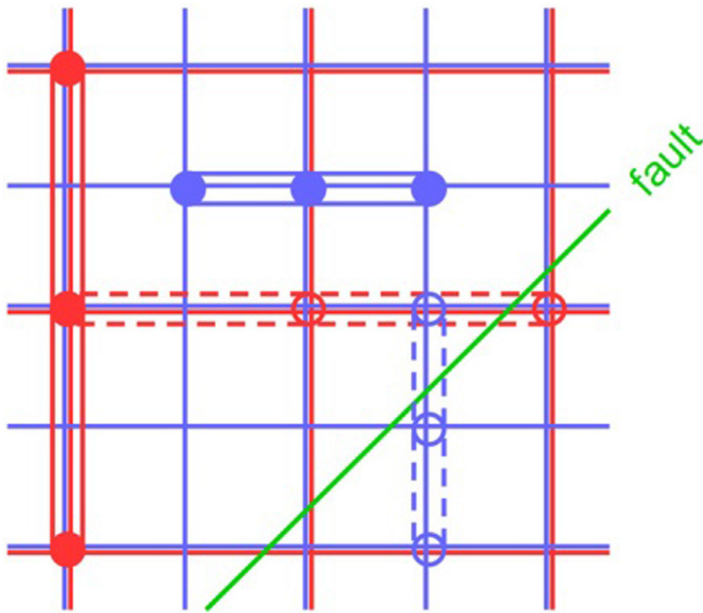


Fig. 3. Le maillage rouge et le maillage bleu représentent deux échelles de modélisation. Les triplets sont des dérivées secondes qui sont conservées en traits pleins, et supprimées en pointillés si elles traversent la faille.

## 2.5 Données et résultats multi-échelles

Les données sont données sur la grille régulière fine. Mais ils doivent être réduits de manière entièrement automatique sur les plus grossiers tout en conservant la moyenne et aussi la pente des points donnés, le cas échéant. Comme les valeurs interpolées sont des scalaires qui pourraient représenter tout ce que nous utilisons le mot pente dans un sens général. Si ces points sont sur un plan, quelle que soit leur distribution, alors le résultat doit appartenir à un tel plan. Ce travail doit être fait pour chaque échelle. Il est possible qu'un groupe de points différents sur des grilles plus fines soit isolé sur des grilles plus grossières.

Les résultats obtenus sur une grille plus grossière doivent être interpolés par une méthode bilinéaire pour initialiser la grille suivante plus fine. Le problème est simple s'il n'y a pas de faute, une interpolation bilinéaire suffit pour deviner les points manquants.

S'il y a un défaut au voisinage du point, l'interpolation bilinéaire échouera car l'erreur peut être aussi grande que la portée du défaut. Dans ce cas, le gradient conjugué finira par résoudre le problème, en faisant un renflement avec une taille croissante et une quantité décroissante, mais cela est très long à faire. Pour cette raison, nous devons l'accélérer par une phase de «gradient pré-conjugué». Avant l'optimisation, cette phase consiste à estimer la valeur des points fermés à la faille en utilisant des points du même côté de la faille mais plus éloignés. Le but est de découpler complètement la valeur d'un point de toute valeur de l'autre côté d'un défaut. La phase de «gradient pré-conjugué» est la clé de la rapidité de la procédure.

## 2.6 Optimisation

L'interpolateur est définie comme étant la fonction  $f(i, j)$  qui minimise la norme  $L^2$  de la dérivée seconde, avec des données arbitraires,  $i$  représentant l'abscisse  $X$  et  $j$  l'ordonnée  $Y$ . Sur une grille rectangulaire de  $N$  points, le nombre d'inconnues peut varier de 1 (un seul point est donné) à  $N - 1$  (tous les points sont connus sauf un). Toutes les situations intermédiaires sont autorisées.

La situation multi-échelles a des conséquences sur l'inversion. Un petit ensemble de points donnés peut montrer une pente moyenne qui influencera le résultat sur les maillages plus fins. Sur les maillages plus grossiers, ce même petit ensemble de points deviendra un seul point, et aucune pente ne peut être attribuée à ce point. Cela signifie que la pente n'a de sens qu'aux échelles les plus fines, et comme le nombre d'itérations est réduit, sa zone d'influence sera également réduite. Ceci est différent d'une inversion à échelle unique où ce petit ensemble aura une influence à long terme sur d'autres points de données.

La mise en œuvre du gradient conjugué est très simple. Le client doit choisir le nombre d'itérations lors de la dernière inversion, et le programme calcule le nombre d'itérations pour les autres inversions. Le rapport entre le nombre d'inconnues et le nombre d'itérations sur la dernière optimisation détermine le nombre d'échelles nécessaires. La formule est:

$$N_O = 1 + \text{ceiling} (\ln_2 (N_U) / 3 \ln_2 (N_I)), \quad (10)$$

$\ln_2(\cdot)$  étant le logarithme de base 2, *ceiling* la fonction qui associe l'entier immédiatement supérieur à une valeur réelle,  $N_O$  le nombre d'optimisations,  $N_U$  le nombre d'inconnues, et  $N_I$  le nombre d'itérations sur la dernière modèle. Pour augmenter (respectivement diminuer) le nombre d'échelles, le plus simple est de diviser (respectivement de multiplier) par 8 le nombre d'itérations sur le dernier modèle.

Prenons un exemple avec  $N_U = 340\,000$ . L'optimisation en un seul passage, conduit à  $N_O = 340\,000$ , et beaucoup de calculs: 3611 s sur une station de travail Dell T3610. Si on optimise en cinq étapes avec 100 itérations sur le dernier modèle, alors le temps de calcul diminue à 2 s. Le tableau 1 montre la relation entre le nombre d'itérations sur le dernier modèle et le temps de calcul.

Tableau 1. Temps de calcul en fonction du nombre d'inversions, pour 340 000 inconnues. Nous préférons 2 s (multi-échelle) à 1 h (simple échelle).

Nombre d'inversions	Nombre d'itérations	Temps (s) de calcul
1	400 000	3611
2	42 240	553
3	5300	78
4	670	11
5	100	2

La figure 4 montre le schéma de l'algorithme. En violet, on voit le nombre d'inconnues sur une échelle logarithmique, avec un facteur égal à 4 (2 pour  $X$  et 2 pour  $Y$ ). Le nombre d'inversions est en bleu. En rouge, le nombre d'itérations avec un facteur 2. Dans le cas d'une inversion et le temps de calcul est d'une heure, alors qu'il n'est que de 2 s si 5 inversions sont utilisées.

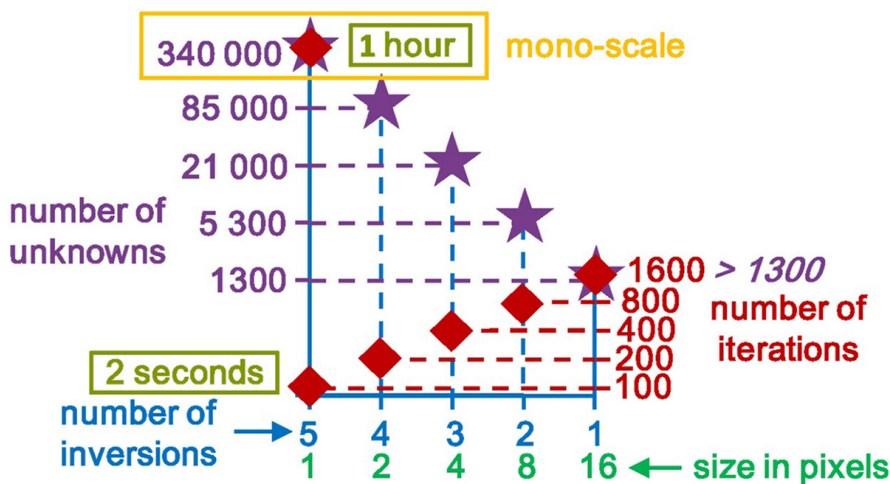


Fig. 4. Schéma de l'algorithme.

## 2.7 Lissé de la surface interpolée

En pratique, on ne voit pas dans les résultats la différence avec une surface de pente continue. En annexe, nous montrons que l'optimisation des dérivées secondes ne conduit pas à une surface résultante  $C^2$ , mais à une surface presque  $C^1$  à cause de l'inégalité de Sobolev.

## 3 Exemples synthétiques

Cette section présente des applications sur plusieurs exemples synthétiques qui développent les caractéristiques de l'interpolateur, d'abord sans défauts, et ensuite avec défauts.

### 3.1 La continuité de la pente

La figure 5a présente le produit de deux sinusoides, une dans  $X$ , l'autre en  $Y$ . La taille du maillage est de 200 par 300 points et la hauteur varie de -100 à 100 m. Nous comparons notre méthode d'interpolation avec la pondération de distance inverse (voir Réf. "Pondération de distance inverse"; Lu et Wong, 2007). Pour plus de simplicité et de robustesse, la pondération de distance inverse présente l'avantage de conserver la plage minimum-maximum entre les données et le résultat. En général, ce n'est pas vrai avec notre interpolateur car la pente est continue à une valeur maximale ou minimale, le creux descend dans une direction et nécessairement le creux monte dans la direction opposée.



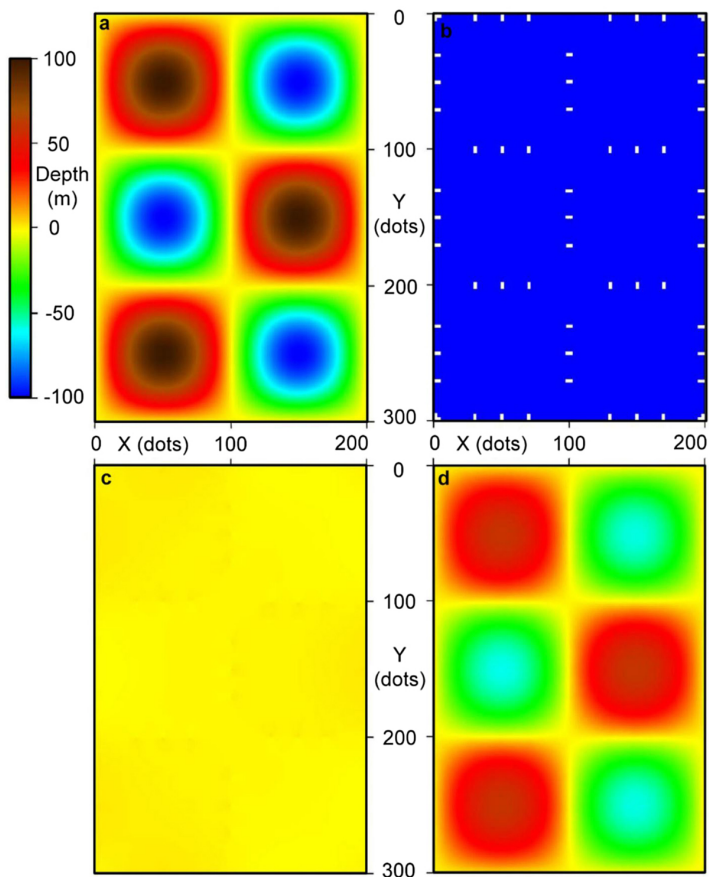


Fig. 5. (a) Modèle vrai variant entre -100 et +100. (b) Les inconnues sont en bleu foncé et les deux points donnent des creux, avec de petites valeurs incluses dans l'intervalle  $[-3, 2, 3, 2]$ . (c) Résultat de la pondération de distance inverse, qui préserve l'intervalle de données  $[-3, 2, 3, 2]$  avec une couleur jaune partout. (d) Résultat de l'algorithme multi-échelles qui préserve les pentes et donne  $[-60, 60]$  gamme globale.

La figure 5b montre des rectangles blancs qui sont des paires de points orientés le long de la longueur. Le bleu foncé représente l'inconnu. Ces points sont très proches de zéro puisque leur maximum est inférieur à 3,2 alors que l'intervalle global est  $[-100, 100]$ .

La figure 5c montre le résultat de la méthode de la distance inverse, qui préserve l'intervalle de données  $[-3, 2, 3, 2]$ . Au contraire, nous pouvons voir sur la figure 5d que notre interpolateur prend en compte les pentes elles-mêmes, et l'intervalle global  $[-60, 60]$  est beaucoup plus proche du théorique  $[-100, 100]$ .

Nous avons ajouté des points de données aux valeurs minimales et maximales, comme le montre la figure 6. Le résultat de la pondération de distance inverse apparaît sur la figure 6c avec une pente discontinue à tous les points de données, en particulier aux points orange. Notre interpolateur (Fig. 6d) donne une pente continue partout, même aux points de données, et l'erreur maximale est inférieure à 6%.

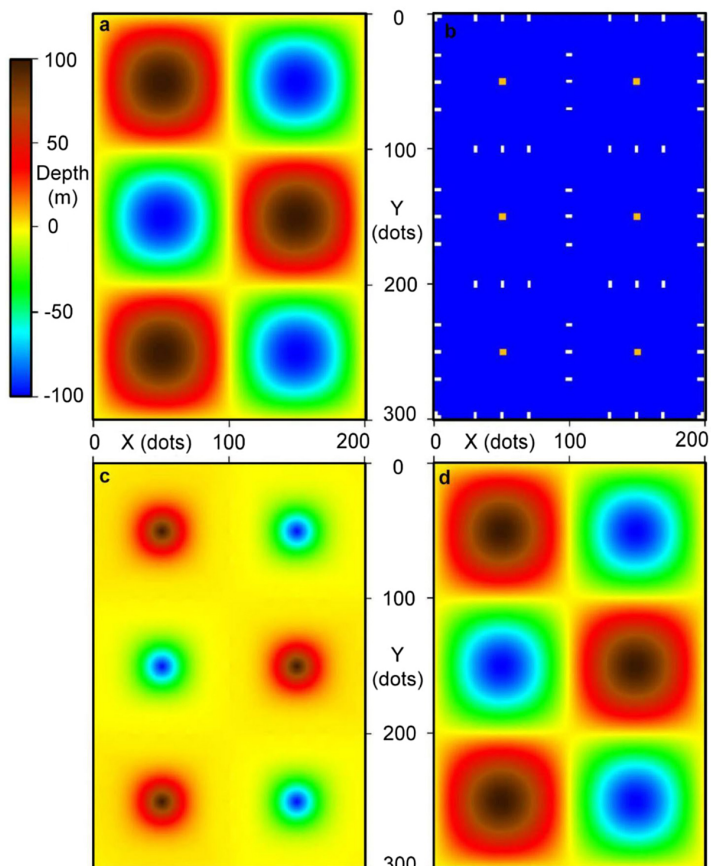


Fig. 6. (a) Même vrai modèle. (b) Six points en orange sont ajoutés aux valeurs minimales ou maximales. (c) Résultat de la pondération de distance inverse, avec discontinuité du pendage aux points de données. (d) Résultat de l'algorithme multi-échelles avec une erreur maximale inférieure à 6%.

### 3.2 Anamorphose

La propriété de distance inverse pour préserver la plage minimum-maximum peut être confortable car il n'y a aucun risque d'obtenir des valeurs aberrantes. Nous avons vu dans le paragraphe précédent que ce n'est pas le cas de notre interpolateur. Il peut arriver que certaines valeurs deviennent négatives et cela peut être un problème pour des propriétés telles que la vitesse. La figure 7a montre trois puits avec des valeurs d'impédance acoustique très contrastées, allant de 3000 à 9000 (m/s) ( $\text{g/cm}^3$ ). Le résultat de la distance inverse (figure 7a) montre des discontinuités au niveau des puits mais la forme générale est satisfaisante. Néanmoins nous pouvons remarquer un «effet de bulle» autour des points 1, 2, 3 et si nous les considérons comme des mesures bien cela pourrait avoir un impact négatif sur la construction du modèle (Sams et Carter, 2017) et plus loin pour des applications géophysiques comme la caractérisation des réservoirs. Ce n'est pas le cas de notre interpolateur (Fig. 7b) puisque l'interpolateur descend à -1500 (m/s) ( $\text{g/cm}^3$ ), ce qui n'est pas acceptable pour l'impédance acoustique par exemple.

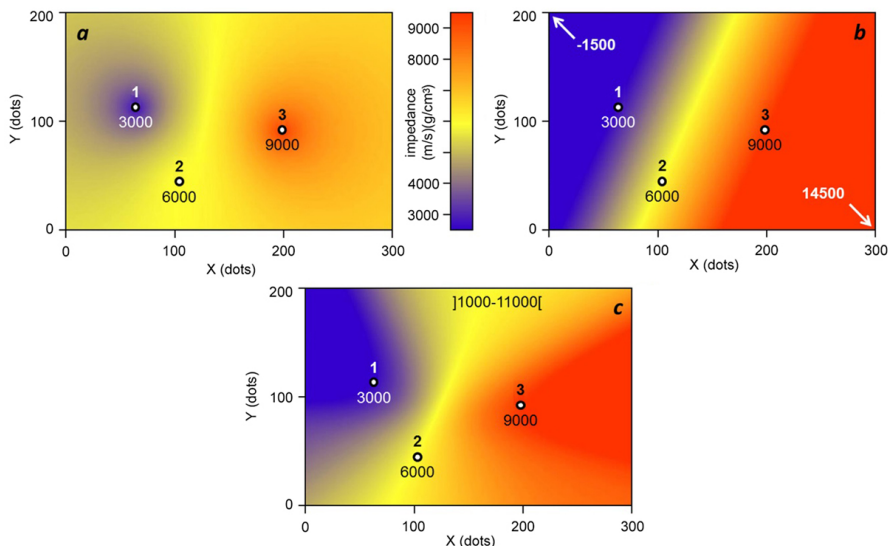


Fig. 7. (a) La pondération de distance inverse semble correcte puisque la plage minimum-maximum est préservée des données au résultat. (b) Notre interpolateur n'est pas correct car le minimum est négatif. (c) L'anamorphose rend le résultat satisfaisant à l'intérieur de l'intervalle ] 1000, 11 000 [(m/s) ( $\text{g/cm}^3$ ).



Pour résoudre ce problème, nous avons besoin d'une technique qui modifie les valeurs en dehors de l'interpolation. L'anamorphose est adéquate à cet effet. Il nécessite un changement de variables avant l'interpolation, ce qui fait que l'intervalle  $]L_1, L_2[$  devient  $] -\infty, +\infty [$ , et la variable inverse change de  $] -\infty, +\infty [$  à  $]L_1, L_2[$  après l'interpolation. Avec  $M_{12} = (L_1 + L_2)/2$  et  $D_{21} = L_2 - L_1$ , la formule de l'anamorphose directe est:

$$g_d(i, j) = M_{12} + \frac{D_{21}}{\pi} \tan \left( \frac{\pi}{D_{21}} (f_d(i, j) - M_{12}) \right), \quad (11)$$

et pour l'anamorphose arrière est:

$$g_d(i, j) = M_{12} + \frac{D_{21}}{\pi} \tan \left( \frac{\pi}{D_{21}} (f_d(i, j) - M_{12}) \right), \quad (12)$$

Les fonctions  $f_d$  et  $g_d$  concernent les données, tandis que  $g_i$  et  $h_i$  concernent le résultat d'interpolation. Les limites  $L_1$  et  $L_2$  ne doivent pas être trop proches des données. Sur la figure 7c,  $L_1 = 1000$  et  $L_2 = 11\,000$  (m/s) ( $\text{g/cm}^3$ ), ont été choisis et le résultat reste dans cet intervalle.

Notez que les limites d'anamorphose ne sont pas forcément constantes, elles peuvent varier avec  $i$  et  $j$ . Dans un tel cas, les quantités  $M_{12}$  et  $D_{21}$  deviennent  $M_{12}(i, j)$  et  $D_{21}(i, j)$ .

Enfin, si l'on compare la figure 7a et le résultat final de notre interpolateur (figure 7c), il n'inclut pas «l'effet de bulle» et respecte la plage physique des limites du paramètre considéré à interpoler.

## 4 Exemples synthétiques avec défauts

Nous introduisons maintenant des failles dans les modèles visant à montrer l'efficacité de la phase «gradient pré-conjugué», qui est critique pour la précision et la rapidité de l'interpolateur. Un autre objectif est de tester le comportement de l'algorithme aux extrémités des défauts.

### 4.1 Cercle

Cet exemple a été conçu pour montrer l'indépendance entre le résultat et le modèle initial. Le cercle blanc, représentant une discontinuité fermée (Fig. 8a et 8f), va montrer comment l'interpolateur gère toutes les orientations de la faille. Dans cet exemple, nous n'avons que deux ensembles de valeurs donnés (deux petits carrés noirs avec des couleurs marron et bleu), l'un à l'intérieur du cercle blanc et l'autre à l'extérieur.

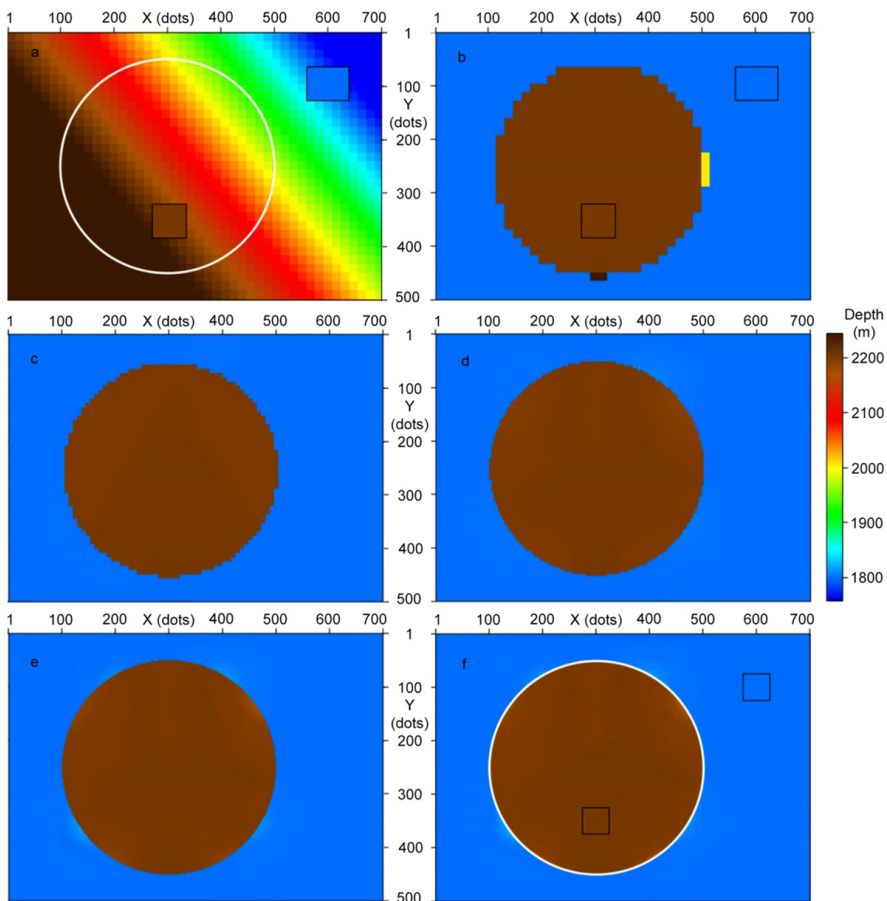


Fig. 8. (a) Modèle initial. (b) Résultat de la première itération avec 1300 paramètres et 1600 itérations. (c) Résultat de la deuxième itération avec 5300 paramètres et 800 itérations. (d) Résultat de la troisième itération avec 21 500 paramètres et 400 itérations. (e) Résultat de la quatrième itération avec 85 000 paramètres et 200 itérations. (f) Résultat de la cinquième et dernière itération avec 340 000 paramètres et 100 itérations. La norme d'erreur  $L^2$  est de 3,5 m pour une différence de 400 m dans les données.

L'ensemble de données se compose de deux carrés horizontaux, l'un à 1 800 m de profondeur, l'autre à 2 200 m de profondeur. La taille de ces carrés est de 50 points, et la taille du domaine inconnu est de 700 par 500 points. La figure 8a montre le modèle initial qui est la régression linéaire des données. Parce que la même palette est choisie, la plus grande gamme sature le brun foncé et le bleu foncé. Sur la figure 8b, le résultat de la première itération n'a rien à comparer avec le modèle initial, et les deux zones constantes à 1800 et 2200 m sont clairement retrouvées. Le nombre d'itérations de cette première inversion est de 1600 et le nombre d'inconnues est de 1300. La figure 8c montre le résultat de la seconde inversion, avec quelques détails supplémentaires. Figures 8d et 8e montrent les résultats des troisième et quatrième itérations. La figure 8f montre le résultat final de la cinquième inversion. Pour une différence de 400 m dans les données, l'écart  $L_2$  par rapport à la théorie est inférieur à 3,5 m, soit moins de 1%. Cette étude de cas synthétique démontre l'efficacité de notre interpolateur à prendre en compte toutes les orientations de faute et même le compartiment (une limite fermée). Dans le résultat d'interpolation final (Fig. 8f), nous avons une couleur brune uniforme à l'intérieur du cercle blanc et une couleur bleue uniforme à l'extérieur.

Le temps global de calcul multi-échelles sur une station de travail Dell T3610 n'est que de 2 s pour 100 itérations sur le dernier modèle, alors qu'une seule inversion directe sur les 340 000 points complets nécessite 3600 s. C'est l'avantage décisif de la technique multi-échelle.

## 4.2 Exemple de cinq défauts

Cet exemple avec des ensembles de données horizontales à différents niveaux vise à illustrer la continuité du creux.

Un carré correspond à  $25 \times 25$  points et les autres à  $50 \times 50$  points. Pour chaque carré, y compris l'intérieur, tous les points ont la même hauteur et l'interpolateur rend le pendage continu à partir des données. Dans la figure 9a, les données bleues sont de 1 700 m de profondeur, bleu – vert de 1 800 m, vert de 1 900 m, orange de 2 000 m, rouge de 2 100 m et marron de 2 200 m. Le résultat de l'interpolation est illustré à la figure 9b. Notez que la zone plate orange est déconnectée des autres. La figure 9c montre le deuxième résultat d'inversion. La zone orange devient connectée à la zone brune. Figure 9d montre le résultat final. Le temps de calcul est de 2 s sur la station de travail Dell T3610 pour cinq inversions et avec 100 itérations sur le dernier modèle. On voit que la forme générale de la solution se fonde à la fin de la première inversion, les autres itérations réglant des détails de plus en plus fins. On peut remarquer, en particulier avec les premières inversions (figure 9b), l'augmentation de la taille des données carrées (carrés noirs) due à la convolution des carrés noirs initiaux (affichés sur la figure 9d) par la taille de pixel actuelle.

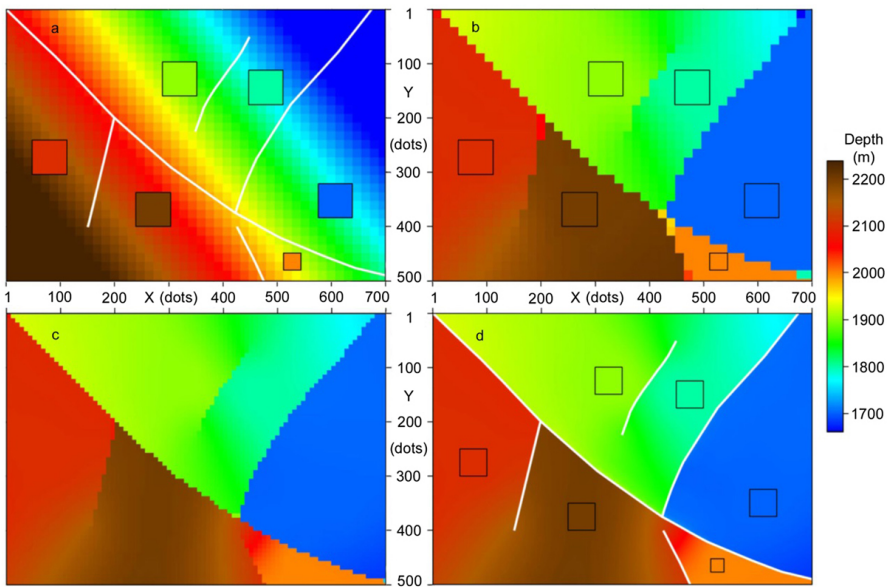


Fig. 9. (a) Modèle initial du premier modèle. (b) Premier modèle avec 1300 paramètres et 1600 itérations. (c) Deuxième modèle avec 5300 paramètres et 800 itérations. (d) Résultat final. Les défauts sont blancs et ont des points à l'intérieur de carrés noirs.

Comme l'exemple synthétique précédent, cette étude de cas plus complexe montre comment se comporte l'interpolateur pour prendre en compte toutes les orientations de faute. Dans le résultat interpolé final (Fig. 9d), nous avons un compartiment avec une limite fermée (voir zone bleue) et un compartiment partiel (voir zone orange). Remarquez également le cas intermédiaire avec le contraste rouge-brun.

### 4.3 Grand modèle avec 54 000 000 points

Cet exemple d'agrandissement des cinq failles vise à montrer la capacité de l'interpolateur à gérer de très gros maillages. Cet exemple montre 9000 points  $X$  et 6000 dans  $Y$ . La figure 10a montre le résultat de la huitième inversion avec 50 itérations sur le dernier modèle. Un pixel du modèle initial correspond à 65 536 pixels du modèle final. Ce travail nécessite 2 min 27 s sur une station de travail Dell T3610.

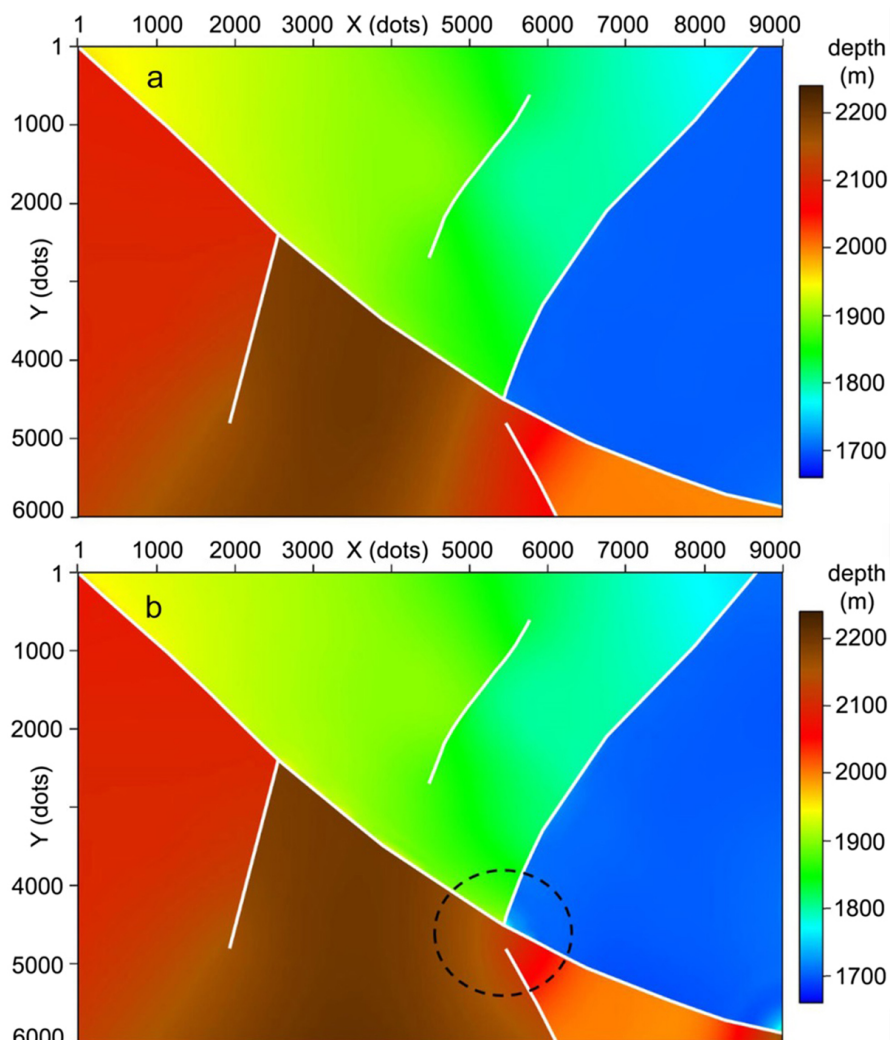


Fig. 10

(a) Résultat d'une interpolation de 54 000 000 points avec 50 itérations sur le huitième et dernier modèle. (b) Résultat avec seulement trois itérations sur le 10e et dernier modèle.

La figure 10b montre le résultat avec seulement trois itérations sur le 10e et dernier modèle. On voit que le résultat n'est pas parfait (voir à l'intérieur du cercle noir en pointillé). Un pixel du modèle initial correspond à 262 144 pixels du modèle final. Le temps de calcul est de 23 s.

## 5 Étude de cas réel: Alwyn (mer du Nord)

Nous présentons le jeu de données Alwyn comme un véritable exemple de terrain. La taille de cet exemple est de 13 km de long et 5,5 km de large, avec des marches de 12,5 m en X et Y, ce qui représente  $\sim 458\,000$  inconnues. Des coupes sismiques ont été prélevées et le résultat est une triangulation, la distance typique entre les points étant d'environ 100 m. Ces points de triangulation sont projetés vers les points les plus proches du maillage régulier, les failles étant des arêtes.

Nous avons supprimé tous les points à moins de 350 m d'une faille. Ce choix est la valeur maximale qui préserve deux points le long du bord est, les deux points permettant le calcul du pendage. La raison de supprimer des points proches du défaut est également justifiée par une plus grande incertitude de repérage due au rapport signal sur bruit.

Nous avons fait deux interpolations sur la figure 11, sans les défauts de (a), et avec les défauts de (b). A partir de cette étude de cas réel, nous pouvons voir l'efficacité de l'interpolateur pour prendre en compte un réseau de discontinuité réel (Fig. 11b). On voit clairement un petit compartiment plein étroit (à droite) et des compartiments partiels (en bas à droite et au milieu). Le temps de calcul est assez rapide, il ne prend que 3 s sur une station de travail Dell T3610 pour cette interpolation  $\sim 458\,000$  inconnues.



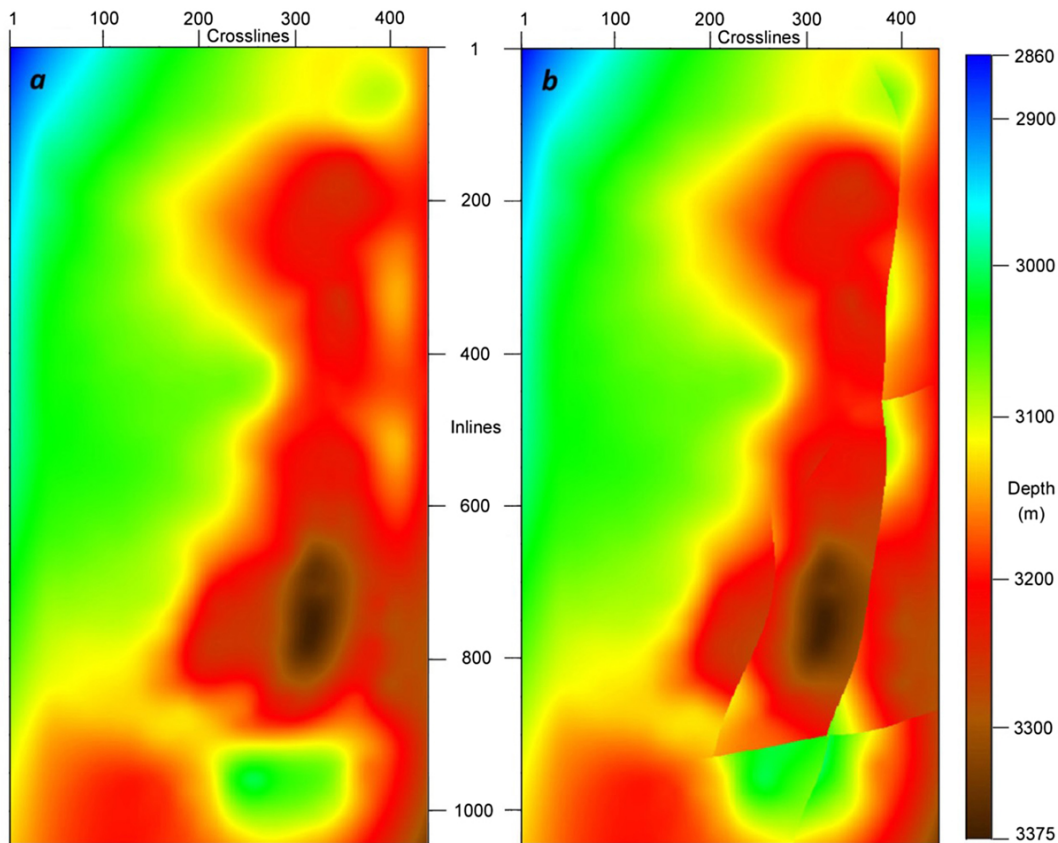


Fig. 11

Nous voyons le résultat de l'interpolation des données Alwyn, sans défauts en (a), avec des défauts en (b).

La figure 12 montre les données d'entrée qui sont affichées en (a), et en (b) nous voyons la différence d'interpolation entre (b) et (a) de la figure 11. On voit clairement ces différences suivant le réseau de discontinuité.

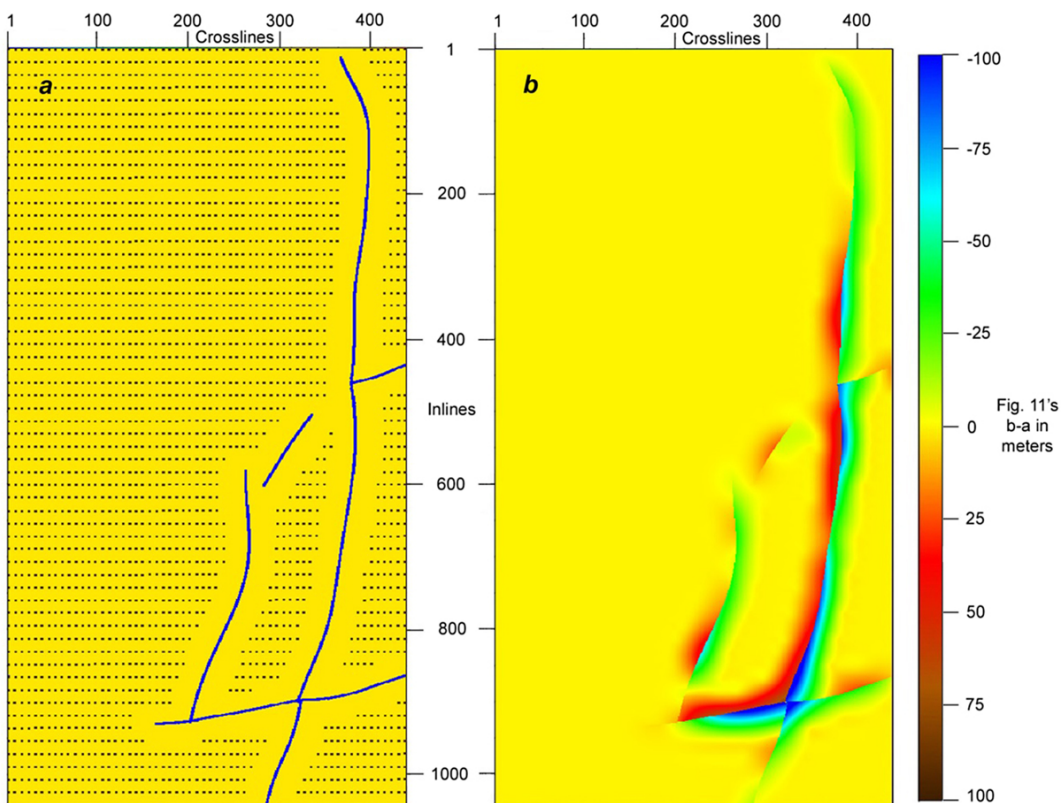


Fig. 12

Les données sont indiquées en (a). La différence entre (b) et (a) de la figure 11 est représentée en (b).

A ce stade on parle de discontinuités et non de faute car on n'a accès qu'à la position ( $X, Y$ ) des failles à une profondeur donnée, la hauteur des surfaces le long de celles-ci est inconnue. Avec plus d'informations, l'interpolateur pourrait être

appliqué plusieurs fois à différentes profondeurs afin de construire un modèle de surface discontinu en 3D. De nombreuses applications en géosciences pourraient bénéficier de l'utilisation *a priori* d'un modèle 3D déficient (Mitra et coll., 2017).

## 6. Conclusion

Pendant plusieurs années, l'interpolation multi-échelle présentée a été utilisée dans divers domaines de travail de Géosciences sans failles pour remplir des informations clairsemées dans des maillages réguliers 2D. Nous mettons en évidence un interpolateur multi-échelles qui présente de nombreux avantages. L'interpolation est rapide car un million d'inconnues ne demande que quelques secondes à calculer. Ce fait est dû à notre approche multi-échelle avec plusieurs inversions au lieu d'une, par conséquent, les passages de l'algorithme de  $N^2$  à  $N$ . De plus, l'interpolation suit non seulement la valeur des données, mais également le creux donné (le cas échéant). A partir de différents exemples traités, on peut également remarquer la distribution arbitraire du jeu de données, de 1 à  $N - 1$  points connus. Enfin, à partir de la dernière étude de cas réel d'Alwyn, l'interpolation est capable de fonctionner avec n'importe quel réseau de linéaments de failles. Le même interpolateur a été utilisé avec la même efficacité avec une dizaine d'autres études de cas réels non publiées, pour des raisons confidentielles. Certains d'entre eux comportaient une soixantaine de failles. Plus généralement, l'algorithme présenté permet d'interpoler n'importe quelle propriété réelle disponible sur un maillage régulier.

## Remerciements

Nous remercions *TOTAL* qui a remis aux membres du Consortium Energistics l'ensemble des données Alwyn North Field utilisées pour réaliser ce cas d'utilisation.

## Appendice

### La régularité de la surface interpolée

Il est bien connu qu'une somme finie de toute fonction  $C^\infty(R)$  donne une fonction  $C^\infty(R)$ , mais cette propriété disparaît si la somme devient infinie. Par exemple, si  $N \rightarrow \infty$ , la fonction

$$S_N(x) = \frac{4}{\pi} \sum_{n=1}^{n=1+2N} \frac{1}{n} \sin\left(\frac{n\pi x}{L}\right)$$

tend vers une fonction carrée de taille unitaire et de longueur  $L$  qui n'est clairement pas continue.

De même, qu'arrive-t-il à notre interpolateur lorsque la taille du pas tend vers 0? Nous travaillons avec des maillages réguliers finis, et il est possible de définir une fonction polynomiale  $C^2$  toujours intégrable, quel que soit le pas fini. Tant que les pas sont finis, il n'y a pas de problème, mais lorsque le pas tend vers zéro, nous avons besoin de l'inégalité de Sobolev.

La fonction  $f(i, j)$  appartient à  $W^{m,p}(O)$ , où  $m$  est le nombre de dérivées dans  $L^p$ ,  $O$  étant un ensemble ouvert de  $R^n$ . Sobolev a montré (voir Réf. « Sobolev Inequality »):

$$k = m - \frac{n}{p} \Rightarrow W^{m,p}(\Omega) \subset C^{k-\epsilon}(\Omega),$$

où  $k$  est le degré de continuité, et  $\epsilon$  est un réel strictement positif. Pour nos besoins, nous avons  $m = 2$  puisque nous minimisons la dérivée seconde,  $p = 2$  puisque nous utilisons des carrés dans la norme,  $n = 2$  puisque la dimension d'une surface est 2, et le résultat est  $k = 1 - \epsilon$  à cause de l'inégalité de Sobolev. Dans la suite, le résultat est dit *presque*  $C^1$ , à cause de  $\epsilon$ . Il y a une différence entre l'ordre de la dérivée que l'on minimise, ici 2, et la continuité du résultat, quel que soit le maillage, qui est ici *presque* 1. Pour les applications en géosciences, la continuité du pendage est une conséquence de celle des surfaces, y compris les points de données isolés.

Notez l'influence de la dimension, nous avons une continuité simple en 3D pour les points isolés, pas la *quasi* continuité de la première dérivée en 2D. Cependant, autour d'une courbe différentiable à propriété différentiable, le résultat sera *quasiment*  $C^1$  en 3D.

La norme a aussi une influence. Si nous choisissons  $p = 1$  au lieu de  $p = 2$ , les résultats sont plus robustes. *Lorsqu'un voyageur atteint une bifurcation de la route, la norme  $l_1$  lui dit de prendre l'un ou l'autre sens, mais la norme  $l_2$  lui demande de partir dans les buissons*. Cette citation se trouve à la page 832 de Claerbout et Muir (1973), ainsi qu'à la page 303 de Tarantola (1987). Les auteurs soutiennent que la norme  $L_1$  est plus robuste que la norme  $L_2$ . Mais l'inconvénient est que le



résultat ne sera que  $C^0$  au lieu de *presque*  $C^1$  selon l'inégalité de Sobolev. Le choix de  $p = \infty$  conduit à la conclusion opposée: la norme  $L_\infty$  prend en compte les courbures minimum et maximum, ce qui n'est pas du tout robuste, mais selon l'inégalité de Sobolev, le résultat sera *presque*  $C^2$  au lieu de *presque*  $C^1$ . La robustesse et la régularité sont opposées.

## Références

- Awange J.L., Paláncz B., Lewis R.H., Völgyesi L. (2018) Mathematical geosciences: Hybrid symbolic-numeric methods, Springer. doi: 10.1007/978-3-319-67371-4.
- Bézier P. (1977) Essai de définition numérique des courbes et des surfaces expérimentales. Contribution à l'étude des propriétés des courbes et des surfaces paramétriques polynomiales à coefficients vectoriels, Thèse d'Etat, Paris VI.
- Bézier P. (1987) Mathématiques et CAO : courbes et surfaces, Tome 4, Hermès.
- Castaño D.D., Jager G., Kunoth A. (2009) Multiscale analysis of multivariate data, in: Architecture and Civil Engineering, K. Gürlebeck, C. Könke (eds), Weimar, Germany.
- Chen C., Li Y., Zhao N., Guo B., Mou N. (2018a) Least squares compactly supported radial basis function for digital terrain model interpolation from Airborne Lidar Point Clouds, Remote Sens. 10, 4, 587. doi: 10.3390/rs10040587.
- Chen C., Li Y., Zhao N., Yan C. (2018b) Robust interpolation of DEMs from Lidar-derived elevation data, IEEE Trans. Geosci. Remote Sens. 56, 2, 1059–1068. doi: 10.1109/TGRS.2017.2758795.
- Claerbout J., Muir F. (1973) Robust modeling with erratic data, Geophysics 38, 5, 826–844.
- Conjugated gradient method: [https://en.wikipedia.org/wiki/Conjugate\\_gradient\\_method](https://en.wikipedia.org/wiki/Conjugate_gradient_method).
- Finite difference: [https://en.wikipedia.org/wiki/Finite\\_difference](https://en.wikipedia.org/wiki/Finite_difference).
- Foster M.P., Evans A.N. (2008) Performance evaluation of multivariate interpolation methods for scattered data in geoscience applications, in: IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Boston, MA, USA.
- Hestenes M.R., Stiefel E. (1952) Methods of conjugate gradients for solving linear systems, Res. Paper Nat. Bureau Stand. 49, 6, 409–436.
- Hjelle Ø., Dæhlen M. (2005) Multilevel least squares approximation of scattered data over binary triangulations, Comput. Vis. Sci. 8, 83–91. doi: 10.1007/s00791-005-0154-7.
- Inverse Distance Weighting: [https://en.wikipedia.org/wiki/Inverse\\_distance\\_weighting](https://en.wikipedia.org/wiki/Inverse_distance_weighting).
- Jespersen D.C. (1984) Multigrid methods for partial differential equations, in: Golub G.H. (ed), Studies in numerical analysis, The Mathematical Association of America 24, pp. 270–318.
- Léger M. (1999) Interpolation from Lagrange to Holberg, in: A. Cohen, C. Rabut, L.L. Schumaker (eds), Curve and surface fitting, Vanderbilt University Press, Nashville, TN, pp. 281–290.
- Léger M. (2016) Method of constructing a geological model. Patent US20160139299.
- Lu G., Wong D. (2007) An adaptive inverse-distance weighting spatial interpolation technique, Comput. Geosci. 34, 1044–1055. doi: 10.1016/j.cageo.2007.07.010.
- Mitra A., Onyia O., Frangeul J., Parsa A. (2017) Alwyn North Field Ocean Bottom Node (OBN) – A step change in seismic imagery, inversion & interpretation, in: 79th EAGE Conference & Exhibition, Paris, We A3 14.
- Ohtake Y., Belyaev A., Seidel H.-P. (2004) A multi-scale approach to 3D scattered data approximation with adaptive compactly supported radial basis functions, Solid Modeling International, pp. 31–39. doi: 10.1109/SMI.2004.1314491.
- Perrin M., Rainaud J.-F. (2013) Shared Earth modeling – knowledge driven solutions for building and managing subsurface 3D geological models. Technip. ISBN-13: 978-2710810025.
- Rogers D.F., Earnshaw R.A. (1991) State of the art in computer graphics – visualization and modeling, Springer-Verlag, New York, pp. 225–269.
- Sams M., Carter D. (2017) Stuck between a rock and a reflection: A tutorial on low-frequency models for seismic inversion, Interpretation 5, 2, B17–B27. doi: 10.1190/INT-2016-0150.1.
- Schneider S. (2002) Pilotage automatique de la construction de modèles géologiques surfaciques, Thèse, Université Jean Monnet et Ecole Nationale Supérieure des Mines de Saint-Etienne.
- Sobolev Inequality: [https://en.wikipedia.org/wiki/Sobolev\\_inequality](https://en.wikipedia.org/wiki/Sobolev_inequality).
- Tarantola A. (1987) Inverse problem theory, Elsevier, Amsterdam.