

# Chapitre 3

## Construction d'une paramétrisation 3D GeoChron

Dans le chapitre précédent, nous avons présenté les grandes lignes du modèle GeoChron, basé sur une paramétrisation 3D de l'espace géologique. Nous avons établi un certain nombre de relations et de propriétés mathématiques de ce modèle.

Nous allons maintenant montrer comment une telle paramétrisation peut être construite en pratique sur un maillage donné. Le paragraphe 3.1 présente quelques principes généraux concernant en particulier la construction du paramètre  $t$ . Nous montrerons aussi que les maillages tétraédriques sont, d'un point de vue informatique et numérique, les plus adaptés à notre problème. Les parties 3.2 et 3.3 traiteront de deux approches différentes, l'une locale et l'autre globale, permettant d'aboutir à une fonction de paramétrisation  $(u, v, t)$  conforme à un modèle GeoChron.

### 3.1 Principe général de construction

Dans un premier temps, nous allons présenter les grandes lignes qui guident nos méthodes de construction. Nous avons vu dans le chapitre précédent que, conceptuellement, toutes les dimensions de l'espace paramétrique GeoChron ne jouaient pas le même rôle. Il semble donc naturel de modéliser séparément  $u$  et  $v$ , les coordonnées paléo-géographiques, et  $t$ , le temps géologique.

On peut remarquer qu'il existe plusieurs relations sur les coordonnées paramétriques. Le *temps* est contraint par la relation d'ordre entre les horizons et par ses relations avec  $u$  et  $v$ , alors que les coordonnées paléo-géographiques ne sont contraintes qu'entre elles et par leurs relations avec  $t$ . En conséquence, il apparaît que  $t$  peut être modélisé en premier, sans connaître  $u$  et  $v$ , alors que le contraire n'est pas possible.

Nous avons donc décidé, dans un premier temps, de construire le *temps*  $t$ , puis de construire les coordonnées paléo-géographiques en s'appuyant sur  $t$ . Il est important de noter que, de ce fait, durant le calcul de  $u$  et de  $v$ ,  $t$  sera toujours considéré comme une donnée fixe, qui ne variera plus jamais. Cela signifie que si des corrections doivent

être apportées sur le *temps* calculé par les méthodes suivantes, ces corrections doivent s'appliquer avant tout calcul de  $u$  ou de  $v$ .

La méthode de construction de  $t$  est assez simple et s'appuie principalement sur le fait que  $t$  est constant le long de chaque horizon du modèle. En ce qui concerne les coordonnées paléo-géographiques, le problème est plus complexe et nous proposons deux approches : une locale, qui construit un ensemble de lignes iso-paléo-géographiques, et une globale, par le biais de contraintes entre  $u$ ,  $v$  et  $t$ . Ces deux méthodes permettent de prendre en compte des informations sur le style de déformation qu'ont subi les terrains.

## L'interpolateur DSI

Dans tout ce chapitre, les modèles de propriété sont construits en utilisant le moteur d'interpolation DSI, pour *Discrete Smooth Interpolation*, qui a déjà été évoqué à plusieurs reprises. Cette méthode a été introduite par J.-L. Mallet ([Mallet, 1992], [Mallet, 2002]), et permet l'interpolation d'une fonction scalaire ou vectorielle sur les nœuds d'un maillage discret, tout en respectant un ensemble de contraintes.

DSI vise à minimiser un critère de *rugosité généralisée*, défini à partir de la rugosité locale du maillage, c'est-à-dire la dérivée seconde de la propriété, et d'un ensemble de contraintes linéaires. Dans notre cas, seules les contraintes souples, c'est-à-dire qui doivent être respectées au sens des moindres carrés, seront utilisées.

Ces contraintes sont toutes de la forme :

$$A_c^t \cdot \varphi \simeq b_c$$

où  $\varphi$  est un vecteur composé des valeurs de la propriété aux différents nœuds du maillage, la matrice  $A_c^t$  et le scalaire  $b_c$  étant des coefficients caractéristiques de la contrainte  $c$ .

Le principe général de DSI est expliqué dans l'annexe A et l'annexe B définit un formalisme concernant les propriétés définies aux nœuds de tétraèdres. Le cadre théorique décrit dans ces annexes sera largement utilisé dans ce chapitre, pour définir un certain nombre de contraintes DSI. Le lecteur est cependant invité à se reporter à [Mallet, 2002] pour la théorie complète de l'interpolateur, à [Mallet, 2003] pour la définition de contraintes aux sommets de tétraèdres, ou encore à [Cognot, 1996] et [Muron *et al.*, 2005a] pour des détails d'implémentation.

### 3.1.1 Calcul du *temps*

Le paramètre *temps* est le premier à être modélisé. Nous avons vu dans le chapitre précédent qu'il doit respecter les contraintes suivantes :

- chaque horizon géologique  $H_t$  doit apparaître comme une surface d'isovaleur de  $t$  ;
- les valeurs de  $t$  associées aux horizons doivent respecter la relation d'ordre définie par l'âge géologique des terrains ;
- le *temps* doit être le plus régulier possible et en particulier doit évoluer continûment entre les différents horizons.

## Principe du calcul

Ces deux objectifs s'expriment facilement sous forme de contraintes DSI. Le respect des horizons initiaux est représenté par les contraintes de type *points de contrôle* (paragraphe B.2.1) et la régularité globale de la propriété correspond à la contrainte de *gradient constant* (annexe B, paragraphe B.3.2).

En pratique, l'algorithme suivant est appliqué :

1. pour chaque horizon initial, un *temps* géologique  $t$  est choisi ;
2. des points de contrôle sont installés, en chaque nœud de cet horizon, avec la valeur de contrôle  $t$  ;
3. une contrainte globale de gradient constant est installée et joue le rôle de « rugosité » suivant la terminologie de la méthode DSI (voir annexe A, paragraphe A.2.1) ;
4. l'interpolation DSI permet de calculer la solution, c'est-à-dire la valeur de  $t$  à tous les nœuds du volume.

Bien qu'exprimée sous forme d'une contrainte DSI, la contrainte de gradient constant représente bien une rugosité dans le cadre de DSI, car elle permet aux valeurs en un nœud d'influer sur les valeurs aux nœuds voisins. Cette relation a été démontrée théoriquement par R. Cognot (communication personnelle). De plus, elle a l'avantage de prendre en compte la géométrie des cellules autour d'un nœud.

La figure 3.1 illustre, sur un jeu de données non-faillées (fourni par Chevron), puis sur un autre jeu de données faillées (fourni par Total), le calcul du *temps* dans des volumes tétraédrisés.

## Origine des données

Plusieurs remarques peuvent être faites sur cet algorithme. D'abord, au niveau des données utilisées en entrée, l'algorithme ne mentionne que des horizons. Dans le cas le plus simple, on dispose en effet d'un modèle structural déjà construit dans lequel plusieurs surfaces triangulées, couvrant une grande partie du domaine d'étude et correspondant à des horizons majeurs, existent.

On peut aussi travailler depuis des données plus en amont, au stade de l'interprétation sismique. Dans ce cas, il est rare de disposer d'horizons complets, ceux-ci étant généralement construits à partir des données plus ou moins denses fournies par la sismique. Il est important de noter que cela ne pose aucun problème au niveau de la construction du paramètre *temps* : à partir du moment où on peut affecter une valeur initiale de  $t$  à chaque fragment d'horizon, en affectant la même valeur aux différents fragments du même horizon, l'interpolation donnera des résultats cohérents. Ceci permet d'utiliser des données très précoces, issues par exemple d'extraction automatique d'horizons (voir par exemple [Labrunye, 2004]).

Enfin, si on se situe à l'inverse, plus en aval par rapport au modèle structural, c'est-à-dire que l'on a déjà commencé à intégrer des données sédimentologiques dans le modèle, ces données peuvent fournir des indications sur le *temps*. Ainsi, des corrélations effectuées entre puits voisins peuvent être considérées comme des isovaleurs de  $t$ . Le modèle Neptune

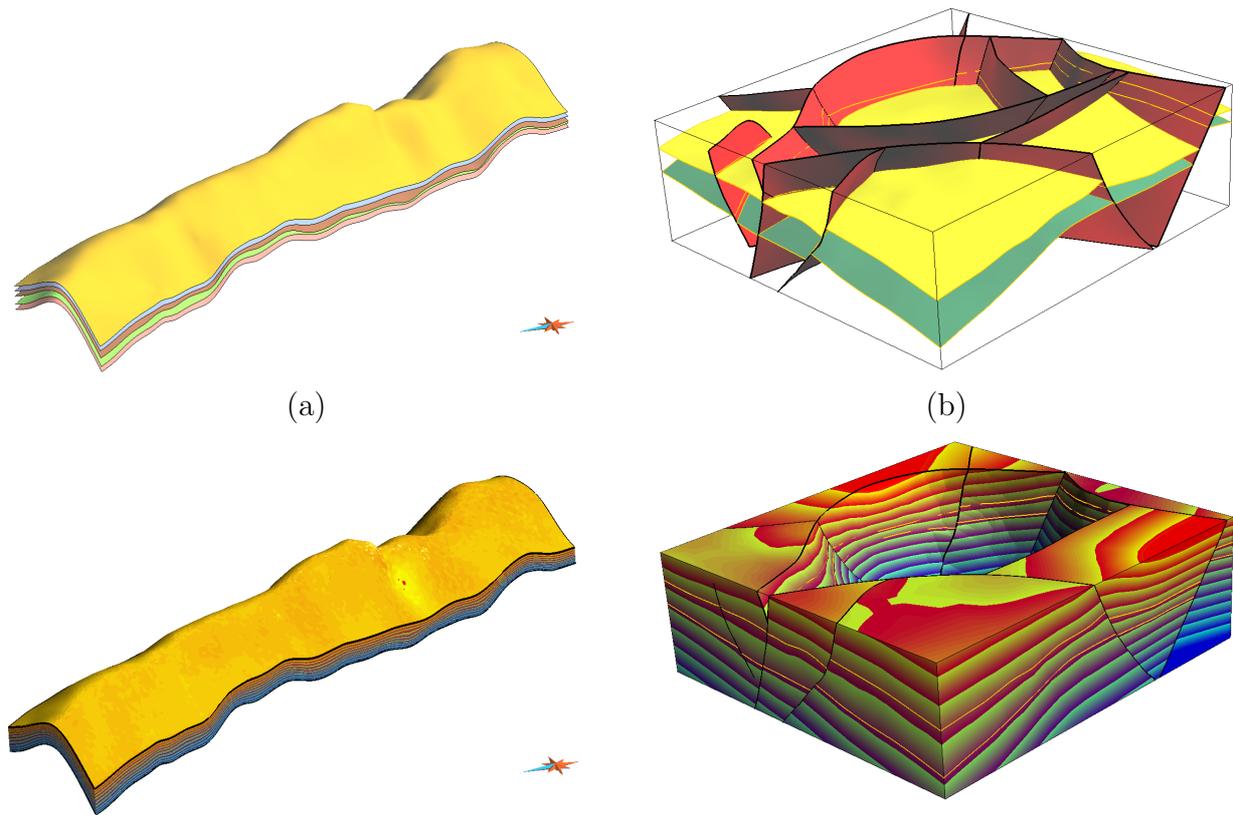


FIG. 3.1 : Exemples de construction de la composante  $t$  de la paramétrisation. En haut, le modèle surfacique, en bas, le temps calculé sur un volume tétraédrique. (a) : exemple non faillé (données Chevron). (b) : modèle faillé (données Total).

(voir [Massonnat, 1999] pour une présentation théorique ou [Leflon et Massonnat, 2004] pour plus de détails) définit de telles corrélations et on peut intégrer les données résultantes sous forme de points de contrôle supplémentaires le long des puits. L'interpolation finale n'en sera que plus précise.

### Choix des valeurs initiales de $t$

Le principal problème de cet algorithme réside dans le choix des valeurs de  $t$ . En effet, la seule contrainte initiale sur ce paramètre est l'ordre que doivent respecter les valeurs sur des horizons d'âge géologique différent :

$$t_0 < t_1 \quad \Longleftrightarrow \quad \{H_{t_0} \text{ est plus âgé que } H_{t_1}\}$$

La solution idéale consisterait à choisir des valeurs de  $t$  correspondant aux âges réels des terrains, mesurés depuis une référence arbitraire plus vieille que le plus vieux terrain inclus dans le modèle. De la sorte, on obtiendrait un espace paramétrique le plus identique possible à chaque couche au moment de sa formation. Malheureusement, il est très rare de connaître avec précision l'âge des différents terrains, en particulier dans les modèles de réservoirs, où l'échelle verticale est très petite.

Quelques tests simples montrent que toutes les valeurs ne sont pas réalistes. Ainsi, la figure 3.2 illustre un cas extrême, où des valeurs de  $t$  mal choisies génèrent des aberrations. Dans cet exemple, le volume d'étude est un cube, dans lequel trois horizons parallèles et équidistants  $H_0$ ,  $H_1$  et  $H_2$  sont définis. La succession des valeurs de  $t$  ne respecte pas ces intervalles, l'écart  $\Delta t_0 = t_1 - t_0$  entre les valeurs des deux premiers horizons étant bien plus faible que l'écart  $\Delta t_1 = t_2 - t_1$  entre les valeurs des deux horizons suivants :

$$\left. \begin{array}{l} t_0 = 1 \\ t_1 = 2 \\ t_2 = 11 \end{array} \right\} \iff \left\{ \begin{array}{l} \Delta t_0 = 1 \\ \Delta t_1 = 9 \end{array} \right.$$

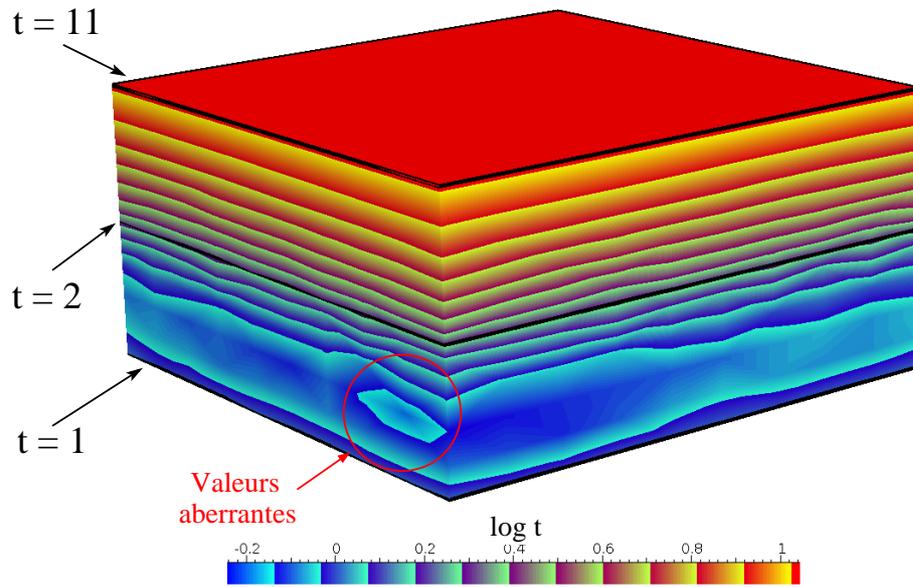


FIG. 3.2 : Influence des valeurs initiales  $t_i$  sur le calcul du temps. Des valeurs de  $t$  constantes ( $t_0 = 1$ ,  $t_1 = 2$  et  $t_2 = 11$ ) ont été affectées sur trois horizons parallèles,  $H_0$ ,  $H_1$  et  $H_2$ , puis le temps a été interpolé dans le volume. Le temps, représenté ici en échelle logarithmique pour mieux mettre en évidence les variations, est erroné entre  $H_0$  et  $H_1$ , car il n'est pas toujours croissant (en particulier dans le cercle rouge). Cette erreur est engendrée par le trop grand écart entre  $\Delta t_0$  et  $\Delta t_1$  pour des couples d'horizons séparés de la même distance  $\Delta h$ .

Le *temps* est bien interpolé entre les deux horizons supérieurs, mais entre les deux horizons inférieurs, l'exigence de continuité de la dérivée imposée par la contrainte de gradient constant provoque des valeurs inférieures à  $t_0$  entre  $H_0$  et  $H_1$ .

Cet exemple caricatural met en évidence le fait que le choix des valeurs de  $t$  n'est pas anodin. Une autre approche du problème initial permet de proposer quelques pistes. En effet, l'interpolation de  $t$  dans le volume peut être considérée comme similaire à un problème de diffusion : chaque horizon est analogue à une plaque le long de laquelle une température constante  $t_i$  est imposée, la valeur de  $t$  dans le volume correspondant à la position d'équilibre du système.

Cela nous a amené à considérer les techniques d'éléments finis pour résoudre ce problème. Des tests ont été effectués avec un code d'éléments finis développé au sein du laboratoire ([Voillemont, 2001]). Il est cependant difficile, avec ces méthodes, d'imposer la continuité du gradient de  $t$  au travers des « plaques » : la solution est donc continue entre chaque paire d'horizons, mais le gradient n'est pas continu au niveau des horizons.

Cette vision du problème permet toutefois d'imaginer une contrainte supplémentaire. En effet, dans les problèmes de mécanique des fluides, on considère souvent le *laplacien* d'une fonction, dont la valeur sur un élément de volume est nulle sauf si cet élément contient une source ou un puits, c'est-à-dire une apparition ou disparition de la quantité considérée (voir par exemple [Lavau, 2004]). Dans notre cas, la continuité que nous voulons obtenir dans le volume traduit le fait qu'il ne devrait y avoir ni sources ni puits, en dehors des horizons. En d'autres termes, le laplacien de  $t$  devrait être nul sur chaque tétraèdre.

Cependant, la formulation mathématique de cette contrainte est très lourde, car elle fait intervenir tous les tétraèdres partageant au moins un nœud avec le tétraèdre considéré, et son implémentation était très lente et n'était pas stable numériquement.

Le problème peut toutefois être contourné, en cherchant non pas à imposer une valeur donnée a priori sur chaque horizon, mais en essayant simplement d'obtenir une répartition des valeurs de  $t$  la plus harmonieuse possible, en fixant la valeur du gradient de  $t$ . Ainsi, on aboutit à la méthode alternative suivante :

1. Le long d'un horizon de référence, fixer  $t = t_0$ , par le biais de points de contrôle ;
2. sur chacun des autres horizons, imposer  $t = \text{constante}$ , sans fixer la valeur de cette constante (par exemple avec une *contrainte delta*, voir paragraphe B.2.2) ;
3. interpoler, en tout point du volume, un champ de vecteurs  $\mathbf{N}(\mathbf{x})$  contraint à être normal aux horizons (en imposant des points de contrôle sur les horizons) ;
4. en tout point du volume, imposer une contrainte sur la norme du gradient de  $t$ , de type  $\|\mathbf{grad} t\| \cdot \mathbf{N} = \text{constante}$ , en fixant arbitrairement la valeur de cette constante (voir paragraphe B.3.1 pour la contrainte DSI correspondante) ;
5. enfin, imposer une contrainte sur la continuité du gradient, en remplacement de la rugosité standard de DSI.

### 3.1.2 Calcul global de $(u, v)$

Une fois la composante  $t$  de la paramétrisation calculée, nous allons construire  $u$  et  $v$ , c'est-à-dire les coordonnées paléo-géographiques.

Quelle que soit la méthode choisie, il est nécessaire de connaître un ensemble de valeurs de référence : de même que pour le *temps* une valeur  $t_0$  était choisie arbitrairement (les autres valeurs pouvant être déduites de cette première valeur), il faut ici définir des valeurs initiales de  $(u, v)$ .

Ces valeurs initiales peuvent aussi être vues comme des « points de calage » des propriétés. Comme toutes les contraintes (continuité du gradient, mais aussi orthogonalité des gradients, etc.) s'appliquent sur les gradients de  $u$ ,  $v$  et  $t$ , seules les variations de ces

fonctions sont fixées et pas leurs valeurs. Pour les fixer, il faut fournir des valeurs exactes en au moins un point.

Dans le cas des coordonnées paléo-géographiques, nous fixons des valeurs initiales le long d'un horizon, appelé *horizon de référence*. Par la suite, le calcul dans le volume peut se faire de plusieurs manières. Nous en proposons deux dans ce chapitre, la première (paragraphe 3.2) se base sur les lignes iso-paléo-géographiques, la deuxième (paragraphe 3.3) sur des contraintes DSI globales. Dans le paragraphe qui suit, nous discutons uniquement du choix de l'horizon de référence et de sa paramétrisation.

### Choix d'un horizon de référence

Le choix de cet horizon de référence n'est pas neutre. D'un point de vue numérique, il est important que cet horizon couvre au mieux la plupart des blocs de faille du domaine. En effet, alors que la continuité des propriétés au sein d'un bloc est fortement garantie par la contrainte de gradient constant, la continuité au travers des failles est plus complexe à imposer. Le chapitre suivant propose une méthode de modélisation des rejets permettant d'ajouter cette continuité mais, même si les rejets sont connus, la convergence est bien plus longue et moins stable en l'absence de points de donnée dans un bloc.

Il est à noter que l'horizon de référence n'est pas obligatoirement un horizon initial du modèle surfacique. En effet, à partir du moment où le *temps* est modélisé, n'importe quelle surface d'isovaleur de  $t$  correspond à un horizon, au sens « un ensemble de points formés au même moment ». En conséquence, on peut sélectionner une surface d'isovaleur de  $t$  qui couvre tout le domaine, si aucun horizon initial n'est satisfaisant.

D'autre part, nous verrons que dans le cadre de l'étude des déformations (paragraphe 5.2), l'horizon de référence définit la *fibres neutre* du modèle et sa position a donc une influence sur le tenseur de déformation calculé au final.

D'un point de vue algorithmique, toutes les méthodes présentées ici sont indépendantes du choix de l'horizon de référence.

### Paramétrisation d'un horizon de référence

Afin de fixer les valeurs initiales de  $u$  et  $v$ , on utilise une paramétrisation 2D de l'horizon de référence. Le problème est alors similaire à celui de la paramétrisation initiale d'une surface dans le cadre de la construction des grilles stratigraphiques par extrusion (paragraphe 1.2.2). Pour cela, plusieurs méthodes sont disponibles et nous avons utilisé la méthode LSCM (*Least Square Conformal Mapping*), présentée par B. Lévy ([Lévy *et al.*, 2002]). La paramétrisation ainsi obtenue respecte les deux contraintes suivantes :

1. les lignes de coordonnées (c'est-à-dire les lignes d'isovaleur de  $u$  et de  $v$ ) sont perpendiculaires en tout point de la surface ;
2. les « cellules » formées par des lignes de coordonnées équidistantes sont de taille similaire.

Ces deux contraintes peuvent se formuler de manière plus mathématique :

1.  $\mathbf{grad} u \perp \mathbf{grad} v$
2.  $\|\mathbf{grad} u\| = \|\mathbf{grad} v\|$

Le paragraphe 3.3.2 montre comment ces deux relations simples peuvent être exprimées sous forme de contraintes DSI. Cette paramétrisation est donc facile à calculer, sur une surface, avec l'interpolateur DSI (il est uniquement nécessaire de fixer la valeur de la paramétrisation en un point, pour les raisons évoquées plus haut à propos du calcul volumique) et donne de très bons résultats (figure 3.3).

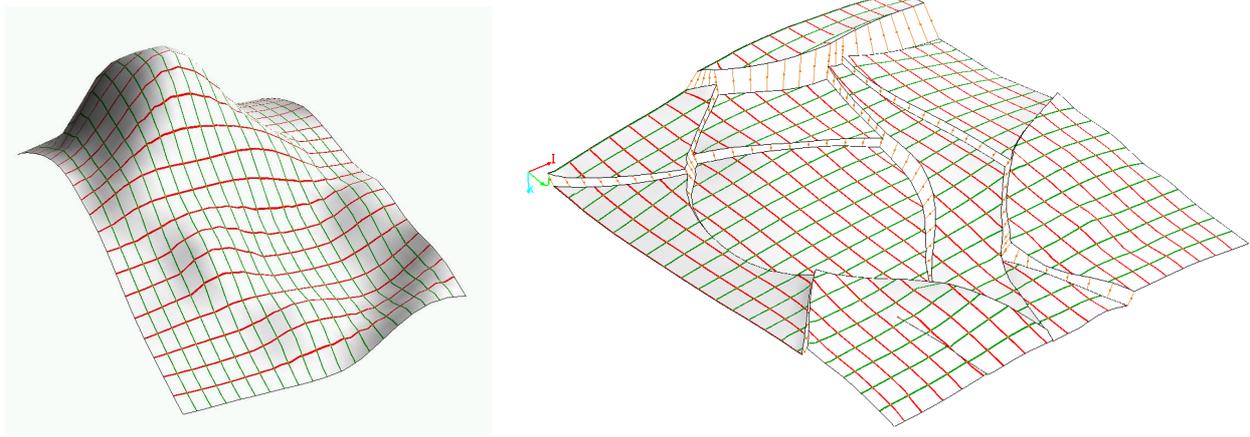


FIG. 3.3 : Paramétrisation 2D par conformal mapping de surfaces triangulées. La paramétrisation est figurée par les lignes d'isovaleurs de  $u$  (en rouge) et de  $v$  (en vert). À gauche, exemple d'une surface fortement plissée, noter les distorsions des lignes dans les zones non dépliables et de forte courbure. À droite, exemple d'une surface faillée, la paramétrisation est mise en cohérence entre les différents blocs grâce aux liens vectoriels (traits oranges, voir chapitre 4 pour plus de détails).

Il est à noter qu'une contrainte supplémentaire peut être ajoutée sur la paramétrisation, afin d'assurer que les cellules définies par des lignes de coordonnées équidistantes sont non seulement de taille constante, mais sont en plus de taille unitaire. En d'autres termes :

$$\|\mathbf{grad} u\| = 1 \quad \text{et} \quad \|\mathbf{grad} v\| = 1$$

Dans ce cas, la paramétrisation obtenue est dite *isométrique* (voir [Samson, 1996] et [Mallet, 2002] ou [Massot, 2002] pour des applications dans le domaine du dépliage surfacique). Cette contrainte est cependant plus difficile à imposer sous forme de contrainte DSI.

J.-L. Mallet a montré récemment ([Mallet, 2005]) que la paramétrisation isométrique est telle que la déformation nécessaire pour passer de la surface réelle à son image dans l'espace paramétrique est minimale. Comme on considère en général que le chemin de moindre effort est privilégié dans les processus naturels, cette paramétrisation peut donc être considérée comme un bon indicateur pour le dépliage d'une surface.

### 3.1.3 Le choix d'un support

Nous avons vu dans le premier chapitre que la construction d'une paramétrisation et ultérieurement son utilisation dans le cadre de la modélisation des propriétés pétrophysiques étaient théoriquement indépendantes du maillage choisi pour porter cette paramétrisation. De plus, le modèle théorique présenté au chapitre 2 est lui aussi indépendant du type de maillage.

En pratique cependant, les caractéristiques des différents maillages ne sont pas neutres, et dans un premier temps différents essais dans des cas simplifiés ont été réalisés afin de déterminer quel type de maillage serait le plus adapté.

#### Les grilles structurées

L'idée initiale a été d'utiliser les maillages les plus faciles à construire, c'est-à-dire des grilles structurées régulières, de type « Voxet ».

Ces grilles ont l'avantage d'être extrêmement faciles à construire, quel que soit le modèle initial. Il s'agit aussi du type de maillage le plus léger en mémoire, puisque la géométrie tout comme les voisinages de chaque nœud sont reconstruits à partir des coordonnées de l'origine et des trois axes. De plus, la résolution du maillage est très facile à adapter à la taille de n'importe quel objet. Basé sur des algorithmes bien rodés de discrétisation de lignes (comme l'algorithme de Bresenham, voir par exemple [Thobie, 1994]) les premiers essais avec ces grilles, sur des cas simples, ont donné de bons résultats (figure 3.4).

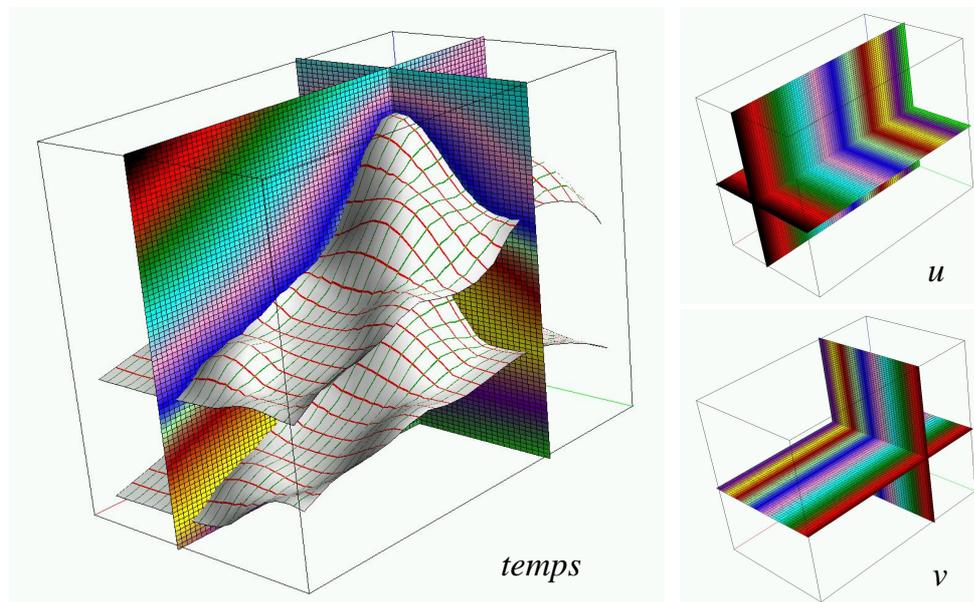


FIG. 3.4 : Paramétrisation GeoChron calculée sur une grille structurée régulière (« Voxet »). À gauche, les deux horizons initiaux, peints avec des lignes d'isovaleur de  $u$  et  $v$  ainsi que le paramètre  $t$  dans le volume; à droite, les composantes  $u$  et  $v$  dans le volume. L'horizon du bas a été utilisé comme horizon de référence pour  $(u, v)$ .

La figure 3.5 illustre les résultats d'un modèle de propriété calculé dans une telle grille (voir paragraphe 5.1 pour les détails de la méthode utilisée).

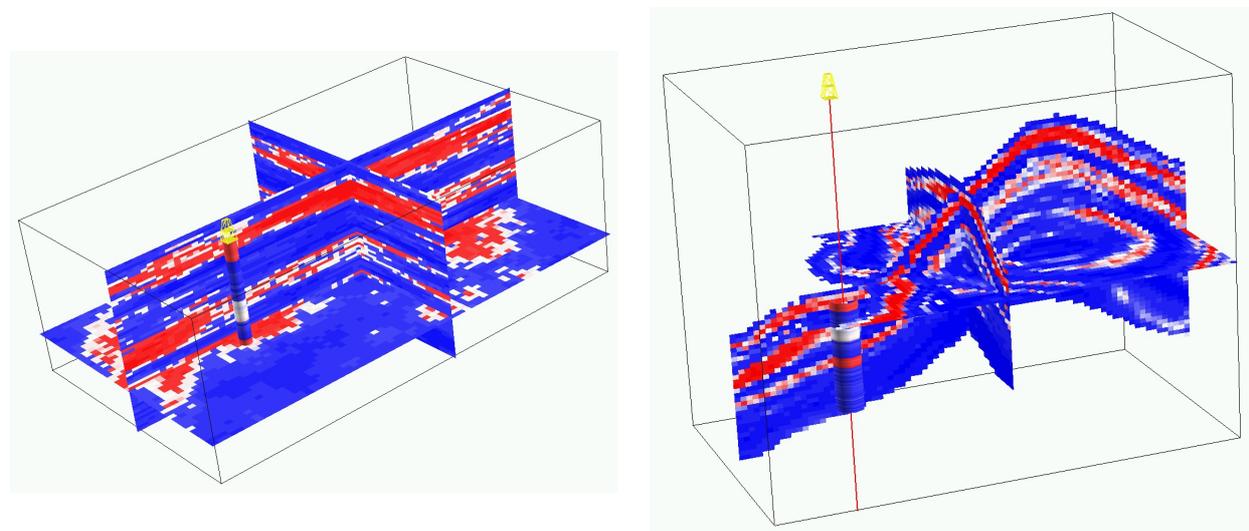


FIG. 3.5 : Visualisation d'une simulation de propriété pétrophysique sur une grille structurée régulière de l'espace géologique. À gauche, une propriété a été simulée à partir des données de puits, dans l'espace paramétrique. À droite, cette propriété est peinte sur le modèle géologique, en respectant la paramétrisation (voir paragraphe 5.1 pour la méthode de transfert).

Ces grilles posent toutefois des problèmes de résolution (figure 3.6) : si on veut obtenir une bonne résolution sur un modèle donné, il est nécessaire d'utiliser des cellules de petite taille, qui seront donc très nombreuses. En outre, la différence d'échelle suivant les différentes directions du modèle (dans un modèle de réservoir, l'extension horizontale peut être de plusieurs kilomètres, contre quelques mètres verticalement) amène à utiliser des cellules fortement aplaties, pour limiter le nombre de cellules horizontales. Ces distorsions de cellules se répercutent sur la paramétrisation calculée.

De plus, dès que nous avons essayé de modéliser des réseaux de failles, de nombreux problèmes sont apparus, dûs à l'absence de représentation particulière des failles dans ces grilles. Il est certes possible de déconnecter les nœuds de part et d'autre d'une faille, mais de nombreux artefacts de calcul apparaissent alors. Une solution serait d'utiliser une grille hybride au niveau des failles :

- Il est possible de déformer les cellules pour qu'elles soient plaquées aux surfaces de faille. On obtient alors une grille irrégulière à côté des failles, très proche des grilles stratigraphiques et avec les mêmes défauts. Pour cette raison, nous avons éliminé cette solution.
- Il est aussi possible d'envisager de diviser les cellules découpées par les failles en plusieurs cellules polyédriques, de telle sorte que la faille apparaisse dans la grille comme un ensemble de nœuds découplés et qu'aucune cellule ne soit traversée par une faille. La grille obtenue est un maillage partiellement non-structuré.

La deuxième solution semblait la plus prometteuse, mais ce type de maillage n'étant pas implémenté dans le géomodeleur GOCAD, il nous a paru plus simple d'utiliser dans

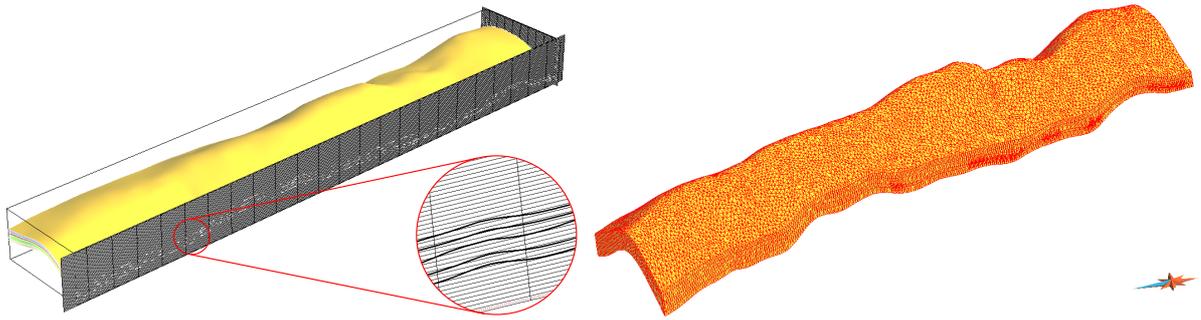


FIG. 3.6 : Modèle surfacique englobé dans un maillage structuré régulier (un « Voxel », à gauche) et dans un maillage tétraédrique non structuré (à droite). Pour garder un faible nombre de cellules au total (ici, de l'ordre de 10 000), les cellules de la grille régulière sont d'amplitudes très différentes verticalement et horizontalement, comme le montre le détail du cercle rouge, ce qui induit des erreurs de calcul. Un modèle aussi fin dans toutes les directions aurait de l'ordre du million de cellules. En comparaison, le modèle tétraédrique permet d'obtenir la même résolution, en se limitant au strict volume d'intérêt, avec environ 12 000 tétraèdres (données Chevron).

tout le volume un maillage non-structuré, qui n'a pas les problèmes de résolution d'une grille régulière.

Au final, il apparaît donc que, lorsqu'on utilise des grilles structurées, la correction des différentes erreurs ou imperfections amène progressivement à se rapprocher soit d'une grille stratigraphique, sans pour autant corriger ses défauts, soit d'un maillage non-structuré, qui est le plus à même de respecter les contraintes géométriques du modèle.

On pourra toutefois noter que l'utilisation de grilles stratigraphiques n'est pas totalement à exclure. En effet, dans le cas où une telle grille a déjà été construite pour d'autres raisons (par exemple lorsqu'une étude précédente est reprise), on peut l'utiliser comme support pour construire une paramétrisation GeoChron. En allant plus loin, on peut même utiliser la paramétrisation implicite  $(i, j, k)$  de ces grilles comme valeurs initiales de la paramétrisation GeoChron. Il est alors possible d'obtenir très rapidement un premier modèle GeoChron, qui se contentera de corriger les erreurs de la grille au voisinage des failles.

Si le point de départ de l'étude est un modèle surfacique et non pas une grille déjà construite, il est cependant préférable de construire directement un maillage volumique qui respecte parfaitement les interfaces, plutôt que de passer par une grille structurée imparfaite.

### Les maillages non-structurés

L'utilisation de maillages non-structurés répond au mieux à l'objectif initial que nous nous étions fixé, à savoir séparer sur trois objets différents d'une part la géométrie du réseau de failles, d'autre part la paramétrisation stratigraphique et enfin le modèle de propriété.

En effet, étant libéré de toute contrainte sur les relations entre cellules, il est possible d'adapter le maillage à n'importe quelle géométrie discrète, aussi fine soit-elle.

Nous nous sommes orientés vers des maillages non-structurés irréguliers formés de tétraèdres. De même que les triangles sont les plus simples des éléments de dimension topologique 2 et sont donc utilisés pour modéliser les surfaces, les tétraèdres sont les éléments de dimension topologique 3 les plus simples.

La création d'un volume tétraédrisé respectant certaines surfaces (principalement les failles) est un problème complexe, que nous ne détaillerons pas ici. Le travail de F. Lepage ([Lepage, 2003]), dont un aspect (le modèle macro-topologique *Soft Frame Model*) a été présenté dans la première partie (paragraphe 1.2.1), contient aussi un code que nous avons réutilisé, permettant la génération d'un volume tétraédrisé sous contraintes.

Enfin, la visualisation de ces maillages est plus complexe que celle des maillages structurés. En effet, sur une grille structurée, il est facile de représenter, par exemple, tous les points situés sur une section d'isovaleur de  $i$ ,  $j$  ou  $k$  (les coordonnées de la paramétrisation implicite). On obtient alors une image de la répartition de n'importe quelle propriété au cœur du domaine d'étude, le long de surfaces géologiquement significatives. Comme il n'existe pas de paramétrisation implicite dans les maillages non-structurés, cette approche n'est pas possible. Nous avons donc utilisé des outils de visualisation en cours de développement par T. Frank ([Frank, 2004], [Frank, 2005]), spécifiquement adaptés à ces volumes et au modèle GeoChron.

Par la suite, lorsque nous parlerons de la paramétrisation GeoChron, nous considérerons uniquement les maillages tétraédriques construits et explorés grâce à ces outils.

## 3.2 Approche locale : les lignes Iso-Paléo-Géographiques

La première méthode de construction d'une paramétrisation GeoChron que nous avons implémentée se base sur les lignes Iso-Paléo-Géographiques ou lignes IPG, qui ont été introduites dans le chapitre précédent (paragraphe 2.1.1 et figure 2.5). Après une présentation du principe de la méthode, quelques indications sur l'implémentation informatique et ses difficultés seront données. Enfin, nous montrerons l'intérêt des lignes IPG dans le cadre de l'édition d'une paramétrisation GeoChron.

### 3.2.1 Principe de la méthode

Nous avons vu que les lignes IPG se définissent comme des lignes le long desquelles les coordonnées paléo-géographiques  $(u, v)$  sont constantes et seul  $t$  varie. L'image de ces lignes dans l'espace paramétrique  $\overline{G}$  est donc une ligne verticale.

Si on arrive à modéliser ces lignes, en tout point du domaine, alors il suffit de connaître les valeurs de  $(u, v)$  en un point de chaque ligne, pour avoir des valeurs dans tout le domaine. Ainsi, si les coordonnées paléo-géographiques  $(u, v)$  sont connues sur un horizon suffisamment étendu, des valeurs de  $u$  et  $v$  pourront être propagées en un grand nombre de points du volume. Ensuite, une interpolation en utilisant ces valeurs comme points de contrôle permettra d'obtenir une solution globale.

En résumé, la méthode se présente ainsi :

1. choisir un horizon de référence, qui couvre au mieux le domaine d'étude (en particulier cet horizon devrait être présent dans chaque bloc de faille) ;
2. construire une paramétrisation 2D  $(u, v)$  sur cet horizon, en utilisant par exemple la *conformal mapping* décrit plus haut ;
3. construire dans le volume un champ de lignes IPG qui croisent l'horizon ;
4. affecter en tout point de chaque ligne IPG les valeurs de  $u$  et  $v$  au point d'intersection de cette ligne avec l'horizon ;
5. utiliser l'ensemble des valeurs affectées sur les lignes IPG comme points de contrôle DSI ;
6. interpoler la solution finale en tout point du domaine d'étude.

La principale difficulté de cet algorithme réside dans le troisième point, la construction du champ de lignes IPG.

### Approximation des lignes IPG

À priori, la géométrie de chaque ligne IPG est inconnue. Cependant, on peut remarquer que ces lignes semblent en général être perpendiculaires aux différents horizons  $H_t$  qu'elles croisent. Nous avons vu dans le paragraphe 2.3.2 que cela n'était pas vrai tout le temps, en particulier lorsque le style de déformation incluait du *flexural slip*.

Cependant, la figure 2.7 illustre un cas extrême, où la déformation est très importante et uniquement accommodée par un glissement banc sur banc. De plus, un des côtés du volume étant fixé, toute la déformation s'est concentrée sur l'autre côté. Dans les cas réels, les différents effets ont une influence bien plus faible et, en première approximation, on peut négliger ce glissement.

Il est toutefois important de conserver cette hypothèse en mémoire, car elle n'est pas forcément valide dans tous les cas. Nous proposons donc aussi quelques pistes qui permettront d'altérer le champ de lignes créées, de telle sorte qu'une composante de *flexural slip* soit ajoutée.

Par la suite, nous désignerons les lignes tracées suivant cette hypothèse comme des *fibres*, pour les différencier des véritables lignes IPG. Cette dénomination met aussi en évidence le parallèle faisable entre ces lignes et les fibres utilisées dans la construction des grilles régulières (paragraphe 1.2.2).

Les fibres que nous allons suivre sont donc perpendiculaires, en tout point du volume, à l'horizon passant par ce point. Or cet horizon est défini comme une surface d'isovaleur du paramètre  $t$ , déjà calculé. En conséquence, les fibres doivent, en tout point, être colinéaires au vecteur  $\mathbf{grad} t$ .

Le problème de la construction d'un champ de fibres se ramène donc au problème du suivi de lignes dans un champ de vecteurs.

### Tracé de lignes dans un champ de vecteurs par la méthode semi-analytique de Pollock

Ce problème est assez courant en physique, puisque le *temps* peut être considéré comme un potentiel et les fibres comme des lignes de courant de ce potentiel. Dans le domaine géologique, cela concerne par exemple le suivi de lignes de courant ou *streamlines*, dans un champ de pression (un état des lieux sur la simulation par lignes de courant se trouve dans [Thiele, 2001] et on pourra se reporter à [Voillemont, 2001] pour un exemple d'utilisation).

Plusieurs méthodes numériques existent pour résoudre ce problème. Dans le cadre des *streamlines*, une des plus utilisées est la méthode de Pollock ([Pollock, 1988]). Elle consiste à considérer que, dans une cellule cubique, le champ de vitesse en tout point peut se décomposer indépendamment en fonction des vitesses aux deux faces opposées, sur les trois axes (figure 3.7). Ainsi, le vecteur vitesse  $\mathbf{v} = (v_x, v_y, v_z)$  au point  $(x, y, z)$  à l'instant  $\tau_1$  peut s'exprimer comme :

$$\begin{aligned} v_x(\tau_1) &= A_x \cdot (x - x_1) + v_{x_1} \\ v_y(\tau_1) &= A_y \cdot (y - y_1) + v_{y_1} \\ v_z(\tau_1) &= A_z \cdot (z - z_1) + v_{z_1} \end{aligned}$$

où  $x_1, y_1$  et  $z_1$  représentent les coordonnées des faces de la cellule et  $A_x, A_y$  et  $A_z$  les gradients de vitesse entre les faces :

$$\begin{aligned} A_x &= \frac{v_{x_2} - v_{x_1}}{\Delta x} \\ A_y &= \frac{v_{y_2} - v_{y_1}}{\Delta y} \\ A_z &= \frac{v_{z_2} - v_{z_1}}{\Delta z} \end{aligned}$$

En utilisant une méthode d'intégration directe pour calculer le mouvement d'une particule entre les temps  $\tau_1$  et  $\tau_2$ , Pollock a montré que la position d'une particule au temps  $\tau_2$  est :

$$\begin{aligned} x(\tau_2) &= x_1 + \frac{v_x(\tau_1) \cdot e^{A_x \cdot (\tau_2 - \tau_1)} - v_{x_1}}{A_x} \\ y(\tau_2) &= y_1 + \frac{v_y(\tau_1) \cdot e^{A_y \cdot (\tau_2 - \tau_1)} - v_{y_1}}{A_y} \\ z(\tau_2) &= z_1 + \frac{v_z(\tau_1) \cdot e^{A_z \cdot (\tau_2 - \tau_1)} - v_{z_1}}{A_z} \end{aligned}$$

En choisissant un pas de temps donné  $\Delta\tau = \tau_2 - \tau_1$ , on peut donc construire la trajectoire d'une particule donnée. Dans les problèmes d'écoulements, les vitesses sur les différentes faces sont calculées à partir de la loi de Darcy (seule  $v_{x_1}$  est donnée ici, les

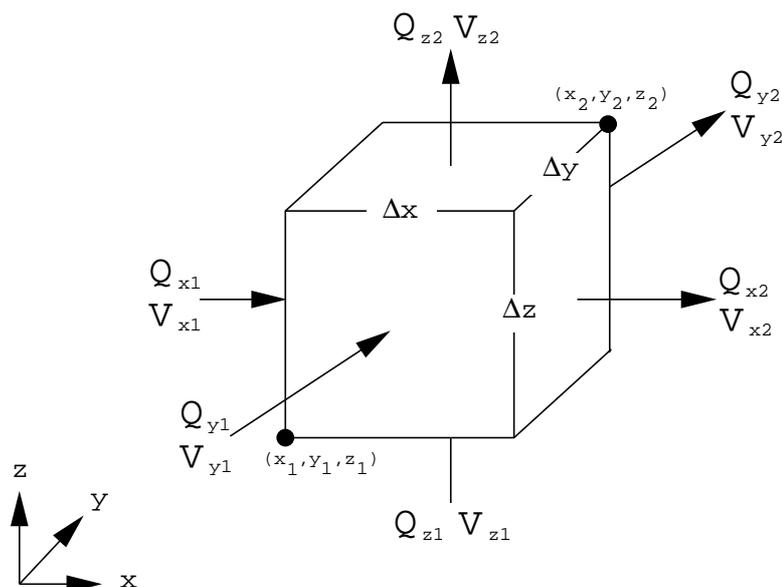


FIG. 3.7 : Notations utilisées dans le cadre de la méthode de tracé de particules de Pollock. Les  $Q_{x1} \dots$  se réfèrent au débit au travers des faces ([Voillemont, 2001], p. 36).

autres ont des expressions similaires) en utilisant une porosité  $\phi$  ainsi que le débit  $Q_{x1}$  (voir figure 3.7 pour les notations) :

$$v_{x1} = \frac{Q_{x1}}{\phi \cdot \Delta y \cdot \Delta z}$$

Dans le cas qui nous intéresse, c'est-à-dire le suivi du gradient du *temps*, on peut considérer que le vecteur **grad**  $t$  représente directement le vecteur vitesse. On a donc directement, en faisant la moyenne sur les faces considérées, les valeurs de  $v_{x1} = \mathbf{grad} t(x1)$ ,  $v_{x2} = \mathbf{grad} t(x2)$ ,  $v_{y1} = \mathbf{grad} t(y1) \dots$  et donc les expressions des coefficients  $A_i$ .

Cette méthode est numériquement très stable, mais elle fonctionne principalement sur des grilles régulières structurées, comme l'illustre la figure 3.7. Des extensions ([Prévoist *et al.*, 2001]) ont permis de l'adapter à des grilles non régulières, de type grilles stratigraphiques, en utilisant la paramétrisation de ces grilles, et même sur des grilles non structurées, à base de tétraèdres. Cependant, nous avons préféré des méthodes numériques plus simples et plus adaptées à ce type de support.

### Tracé de lignes de courant par la méthode numérique de Runge-Kutta

Plusieurs méthodes numériques permettent de reconstituer la trajectoire d'une particule dans un champ de vitesse défini aux nœuds d'un maillage quelconque. La plus simple est la méthode dite d'Euler, où le point suivant  $\mathbf{x}(\alpha + 1)$  d'une ligne de courant est déterminé à partir du vecteur vitesse  $\mathbf{v}(\alpha)$  au point précédent  $\mathbf{x}(\alpha)$  et d'un pas d'échantillonnage  $dl$  :

$$\mathbf{x}(\alpha + 1) = \mathbf{x}(\alpha) + \frac{\mathbf{v}(\alpha)}{\|\mathbf{v}(\alpha)\|} \cdot dl$$

Le pas d'échantillonnage  $dl$  peut être constant, pour avoir des points équidistants, ou variable, par exemple en se basant sur un pas de temps constant  $d\tau$  et en calculant le pas d'échantillonnage  $dl$  en fonction de la vitesse en chaque point :  $dl = \|\mathbf{v}(\alpha)\|.d\tau$ . On obtient alors des points espacés régulièrement dans le temps. Dans notre cas, la notion de temps n'a pas de sens et nous utiliserons des pas d'espace constants.

Cette méthode est très simple et très rapide, mais elle est peu stable numériquement, en particulier lorsque le champ de vitesse (ou, dans notre cas, le champ de  $\mathbf{grad} t$ ) varie très fortement localement. En effet, comme l'estimation de la trajectoire ne se fait qu'au point de départ, les résultats ne tiennent pas compte de l'évolution du champ de vitesse entre deux points successifs.

On préfère donc utiliser la méthode de Runge-Kutta (voir par exemple [Enright *et al.*, 1994] ou [Press *et al.*, 1992] pour des détails sur l'implémentation numérique) qui peut être vue comme un perfectionnement de la méthode d'Euler par itérations (figure 3.8).

Si on considère deux points successifs le long de la fibre  $\mathbf{x}(s)$  et  $\mathbf{x}(s+h)$ , la forme d'Euler indique :

$$\mathbf{x}(s+h) = \mathbf{x}(s) + h.\mathbf{T}(\mathbf{x})$$

où  $\mathbf{T}(\mathbf{x})$  représente la dérivée au point  $\mathbf{x}$ , c'est-à-dire le vecteur tangent à la fibre en  $\mathbf{x}$  :

$$\mathbf{T}(\mathbf{x}) = \frac{d\mathbf{x}}{ds}$$

La forme classiquement utilisée, à l'ordre quatre, de la méthode de Runge-Kutta passe par une suite d'estimations de la dérivée à des points médians dans l'intervalle considéré. Cette estimation s'écrit :

$$\begin{aligned} \mathbf{K}_1(\mathbf{x}, h) &= h.\mathbf{T}(\mathbf{x}) \\ \mathbf{K}_2(\mathbf{x}, h) &= h.\mathbf{T}(\mathbf{x} + 1/2.\mathbf{K}_1(\mathbf{x}, h)) \\ \mathbf{K}_3(\mathbf{x}, h) &= h.\mathbf{T}(\mathbf{x} + 1/2.\mathbf{K}_2(\mathbf{x}, h)) \\ \mathbf{K}_4(\mathbf{x}, h) &= h.\mathbf{T}(\mathbf{x} + \mathbf{K}_3(\mathbf{x}, h)) \\ \mathbf{x}(s+h) &= \mathbf{x}(s) + 1/6\left(\mathbf{K}_1(\mathbf{x}, h) + 2.\mathbf{K}_2(\mathbf{x}, h) + 2.\mathbf{K}_3(\mathbf{x}, h) + 6.\mathbf{K}_4(\mathbf{x}, h)\right) \end{aligned}$$

Comme cette méthode prend en compte les valeurs intermédiaires du champ de vitesses pour déterminer le point suivant, elle est nettement plus stable que la méthode d'Euler, en particulier lorsque l'orientation du champ de vitesse varie rapidement. En contre-partie, elle est plus lente, car elle nécessite de calculer les dérivées intermédiaires. Cependant, nous ne construirons qu'un nombre limité de fibres et le gain de stabilité est tel que cette méthode est de très loin préférable à la méthode d'Euler.

La méthode de Runge-Kutta fournit un échantillonnage à pas d'espace constant le long des fibres. Nous avons donc uniquement à déterminer ce pas d'échantillonnage. Comme les points des fibres servent ensuite de points de contrôle dans une interpolation DSI, il faut essayer d'avoir au moins un point de contrôle par tétraèdre traversé, pour contraindre au mieux l'interpolation. En revanche, bien qu'il ne soit pas utile d'avoir plus d'un point par tétraèdre, car cela ralentit inutilement l'interpolation, cela n'est pas très gênant, en

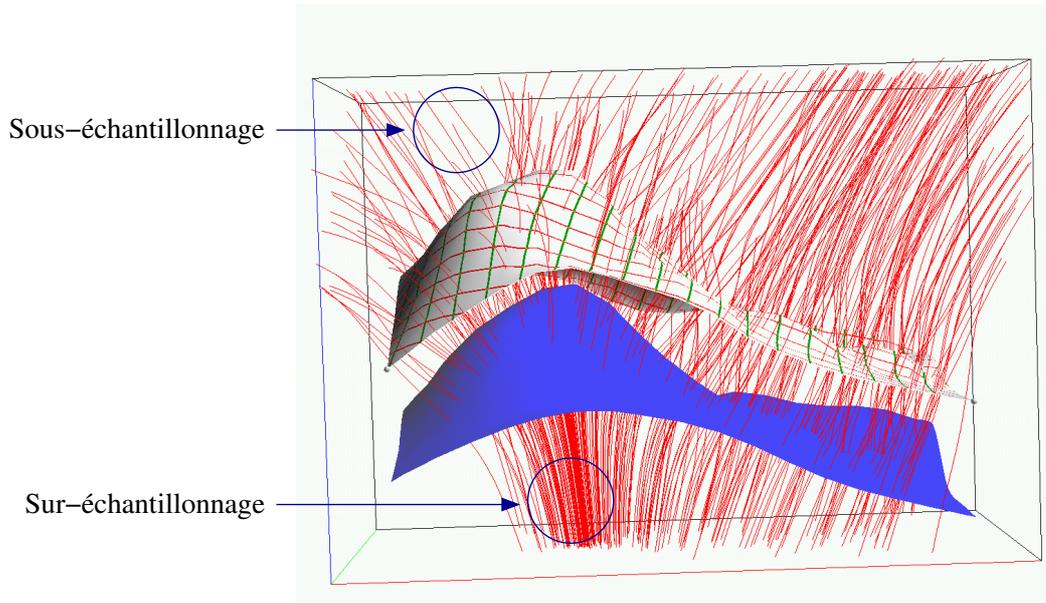


FIG. 3.8 : Champ de fibres extraites depuis un horizon de référence par la méthode de Runge-Kutta. Une paramétrisation par conformal mapping (représentée par les lignes d'isovaleur de  $u$  et  $v$  rouges et vertes) est calculée sur l'horizon de référence (en haut) et sera propagée dans le volume le long des fibres. La densité des fibres est très variable suivant la position dans le volume et dépend de la courbure de l'horizon de référence.

particulier lorsque ces points sont cohérents, ce qui est le cas ici. En conséquence, on choisira comme distance d'échantillonnage  $h$  une distance telle qu'on trouve au moins un point par tétraèdre.

Dans un maillage tétraédrique, les tétraèdres peuvent avoir des tailles très différentes suivant les endroits, il n'est donc pas possible de calculer rapidement une longueur  $h$  qui soit satisfaisante en tout point du volume. On peut cependant considérer qu'en prenant une distance  $h$  de l'ordre de 1% de l'extension verticale totale du modèle, on aura un résultat satisfaisant, car il est rare que les modèles aient plus de quelques dizaines de tétraèdres d'épaisseur. En pratique, l'utilisateur peut définir le nombre approximatif de points souhaités le long de la fibre, et le pas  $h$  en est déduit à partir de l'extension verticale totale du modèle.

### Le problème des failles

La méthode de construction de fibres en suivant le gradient de  $t$  est très similaire aux méthodes d'extrusion de fibres utilisées dans la construction des grilles stratigraphiques (paragraphe 1.2.2 et [Souche, 2005]). Cependant, nous avons à traiter ici un problème supplémentaire, concernant les failles.

En effet, les fibres utilisées pour construire des grilles structurées ne croisent jamais les failles, par définition, puisque les empilements de cellules s'appuient dessus. À l'inverse, dans notre cas, la paramétrisation GeoChron n'est pas construite à partir des failles,

puisque celles-ci sont postérieures à la formation des terrains qu'elles affectent. En conséquence les lignes IPG, et donc les fibres qui les modélisent, peuvent croiser les failles.

Si on considère une ligne IPG coupant une faille (figure 3.9) :

- dans l'espace paramétrique  $\overline{G}$ , cette ligne est verticale et la faille n'existe pas encore (on peut toutefois représenter la trace future de la faille) ;
- dans l'espace géologique  $G$ , la faille a décalé les terrains et la ligne IPG est maintenant discontinue, les deux extrémités étant décalées suivant le vecteur rejet.

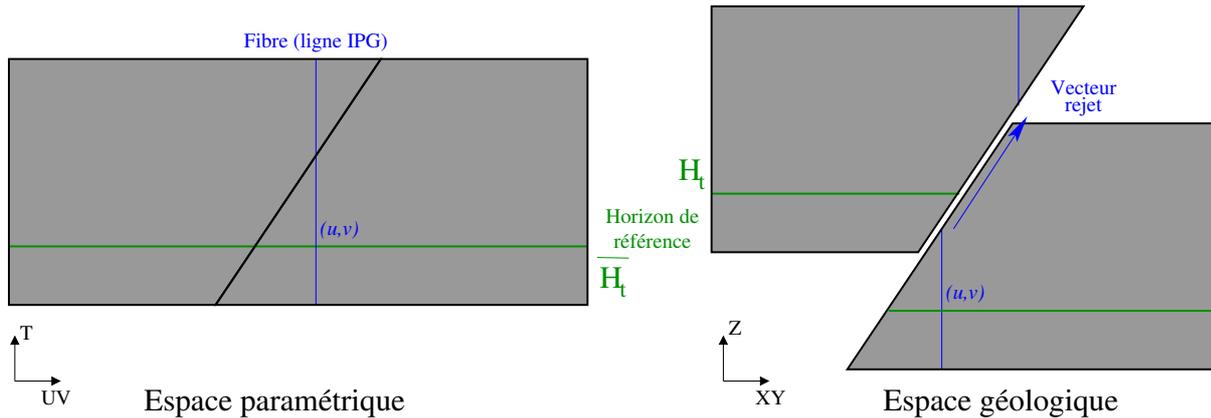


FIG. 3.9 : Principe de tracé des fibres au travers d'une faille. À gauche, une fibre (en bleu) extraite en un point  $(u, v)$  sur l'horizon de référence  $H_t$  ( $\overline{H}_t$  dans l'espace paramétrique, en vert) apparaît dans l'espace paramétrique comme une ligne verticale, car c'est une ligne IPG. Son image dans l'espace géologique (à droite) est quant à elle décalée suivant le rejet de la faille.

Il est donc nécessaire, pour suivre une ligne IPG ou une fibre au travers d'une faille (figure 3.10), de connaître le rejet de cette faille. Plusieurs méthodes existent. Dans un premier temps, nous avons utilisé une méthode développée par L. Souche ([Souche, 2005]) qui se base sur la géométrie de la trace des horizons sur les failles. Dans un deuxième temps, nous avons développé une méthode plus générale et plus exacte, qui fera l'objet du prochain chapitre. Cette méthode se base sur le paramètre *temps* et permet de calculer géométriquement un vecteur rejet en tout point d'une faille.

### 3.2.2 Mise en place de l'algorithme et implémentation

En résumant les étapes décrites précédemment, on obtient l'algorithme d'extraction de failles suivant :

1. Modéliser le *temps*, en utilisant par exemple la méthode décrite dans le paragraphe 3.1.1 (ou toute autre méthode).
2. Calculer, sur chaque tétraèdre, le vecteur gradient du *temps*  $\mathbf{grad} t$ . Comme ce vecteur sera utilisé de nombreuses fois, il est plus efficace de le calculer une seule fois au début et de le stocker en mémoire.
3. Choisir un horizon de référence (soit un horizon du modèle surfacique initial, soit une surface d'isovaleur de  $t$ ) de telle sorte qu'il couvre au mieux le domaine d'étude.

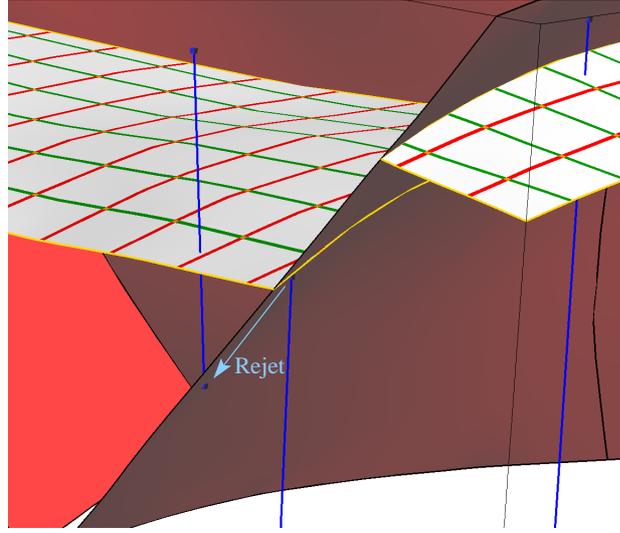


FIG. 3.10 : Fibre intersectant une faille et décalée suivant le vecteur rejet. Les fibres bleues sont extraites depuis l'horizon de référence sur lequel une paramétrisation initiale a été définie (lignes rouges et vertes). À droite, une autre fibre, non affectée par les failles.

4. Construire une paramétrisation 2D  $(u^*, v^*)$  sur cet horizon, par exemple par la méthode de *conformal mapping* décrite plus haut.
5. Choisir un ensemble de points d'extraction des fibres sur cet horizon, ainsi qu'un pas d'échantillonnage  $h$ .
6. Pour chaque point d'extraction  $P_i$  de coordonnées paramétriques sur la surface  $(u_i^*, v_i^*)$ , construire la fibre correspondante en utilisant la méthode de Runge-Kutta en suivant le champ de **grad**  $t$ . Affecter la valeur  $(u_i^*, v_i^*)$  à tous les points de la fibre.
7. Utiliser toutes les valeurs de  $(u^*, v^*)$  définies sur les fibres pour interpoler la solution finale  $(u, v)$  sur tous les nœuds.

### Découpage des fibres par les failles

Dans le cas où le volume est faillé, les fibres doivent prendre en compte ces failles, grâce au vecteur rejet. En pratique, l'algorithme suivant est utilisé :

- 1:  $\mathcal{D} \leftarrow$  domaine d'étude tétraédrisé
- 2:  $h \leftarrow$  pas d'échantillonnage des fibres
- 3:  $\{\mathcal{F}_k\} \leftarrow$  ensemble des surfaces de faille du modèle
- 4:  $P_0 \leftarrow$  point de départ de la fibre sur l'horizon de référence
- 5:  $\mathcal{L}(P_0) \leftarrow$  fibre extraite depuis le point  $P_0$
- 6:
- 7: Label *extraction*
- 8: **tant que**  $P_i \in \mathcal{D}$  **faire**
- 9:   //. `runge_kutta(P, h)` est une fonction permettant de calculer
- 10:   //. le successeur de  $P$ , le long d'une fibre, à un pas de  $h$ .
- 11:    $P_{i+1} \leftarrow$  `runge_kutta(P_i, h)`

```

12: Ajouter  $P_{i+1}$  à la fin de  $\mathcal{L}(P_0)$ 
13: fin tant que
14:
15: // Vérifier l'intersection avec des failles
16: si  $\mathcal{L}(P_0) \cap \{\mathcal{F}_k\} \neq \emptyset$  alors
17:    $P_+ \leftarrow \mathcal{L}(P_0) \cap \{\mathcal{F}_k\}$ 
18:   Insérer  $P_+$  dans  $\mathcal{L}(P_0)$ 
19:   Supprimer tous les points  $P$  de  $\mathcal{L}(P_0)$  tels que  $\text{distance}(P_0, P) > \text{distance}(P_0, P_+)$ 
20:   // rejet( $P$ ) est une fonction calculant le vecteur rejet en un point.
21:    $P_- \leftarrow \text{rejet}(P_+)$ 
22:    $P_0 \leftarrow P_-$ 
23:   Revenir au label extraction
24: fin si

```

Cet algorithme revient à extraire la totalité d'une fibre comme si il n'y avait pas de failles, puis à découper les morceaux au-delà des failles, calculer le point de sortie des fibres grâce au vecteur rejet, puis reconstruire la fibre depuis ce point de sortie.

On notera que de grandes parties de chaque fibre sont calculées inutilement. Cependant, cette solution est bien plus facile à implémenter, et le code résultant à maintenir, pour une perte de temps peu sensible, car le tracé d'une fibre est quasiment instantané : sur un volume de l'ordre de 100 000 tétraèdres avec une dizaine de failles, le tracé d'un millier de fibres prend moins de 5 secondes sur un Pentium III 1,2 GHz.

Enfin, cet algorithme n'extrait les fibres que dans une direction, en suivant la direction de **grad**  $t$ . De ce fait, les fibres ne sont calculées que d'un côté de l'horizon de référence. Pour avoir la totalité de la fibre, il suffit de refaire le même calcul en considérant cette fois **-grad**  $t$ .

### Contrôle de la densité des fibres

Pour que les fibres contraignent correctement le volume, il est important qu'elles couvrent la totalité du domaine. À l'inverse, il ne sert à rien d'avoir plusieurs fibres cohérentes qui contraignent le même tétraèdre. Nous avons vu comment la densité de points le long d'une fibre peut être ajustée, mais il est aussi possible de contrôler la densité de fibres dans le volume de deux manières différentes.

Le plus simple est de contrôler la densité des points d'extraction. Les fibres sont extraites depuis un ensemble de points sur l'horizon de référence et cet ensemble peut être adapté au volume. C'est particulièrement utile lorsque la résolution du maillage de l'horizon est très différente de la résolution du volume tétraédrique (par exemple, si l'horizon a été extrait automatiquement depuis un cube sismique, sa résolution est celle du cube sismique, qui est bien plus fin que les tétraèdres).

Pour cela, un algorithme de rééchantillonnage a été implémenté, avec l'aide de R. Coignet. Cet algorithme détermine une densité optimale de fibres sur la surface en fonction du maillage volumique. Il est à noter que dans les premiers essais avec un maillage structuré régulier, ce rééchantillonnage était simple, puisque la résolution est constante. Sur des

maillages non-structurés, nous avons choisi d'imposer une résolution constante, fonction de la taille moyenne des tétraèdres.

Mais ce rééchantillonnage ne résoud pas tous les problèmes et certains tétraèdres du volume ne sont pas pour autant traversés par des fibres (voir figure 3.8). On peut distinguer deux cas :

- Si l'horizon de référence est fortement plissé, les fibres partant des convexités seront très divergentes et, lorsqu'on s'éloigne beaucoup de l'horizon, ne traverseront pas tous les tétraèdres. Ce problème peut être résolu en augmentant la densité de points d'échantillonnage sur la surface, mais d'une part il est difficile de prédire la densité qui sera nécessaire et d'autre part la densité sur l'horizon devient parfois trop importante, et des erreurs numériques empêchent de contrôler correctement la trajectoire des fibres.
- D'autre part, il existe des points tels que les fibres les traversant ne croisent pas l'horizon de référence. Pour ces tétraèdres, il n'est pas possible de jouer uniquement sur la densité d'échantillonnage sur l'horizon de référence.

Nous proposons donc un algorithme par étapes, mis au point avec R. Cognot, qui permet de contourner ces deux problèmes. Il s'agit principalement de construire les fibres jusqu'à un horizon intermédiaire, sur lequel la paramétrisation  $(u, v)$  est propagée, avant d'extraire un nouvel ensemble de fibres :

1. Un horizon de référence  $H_0$  est sélectionné, avec une paramétrisation  $(u_0, v_0)$ .
2. Un ensemble d'horizons intermédiaires  $\{H_i\}$  est sélectionné. Ces horizons peuvent être soit des horizons déjà connus, soit des surfaces d'isovaleur de  $t$  harmonieusement réparties dans le volume.
3. Une résolution initiale est choisie sur  $H_0$ , en fonction de la taille des tétraèdres alentour.
4. Un ensemble de fibres est extrait depuis  $H_0$ . Chaque fibre est interrompue lorsqu'elle croise un horizon intermédiaire. La paramétrisation  $(u_0, v_0)$  est propagée le long de ces fibres.
5. Des points de contrôle avec les valeurs  $(u_0, v_0)$  sont installés sur l'horizon  $H_i$  et sont utilisés pour contraindre le calcul d'une nouvelle paramétrisation 2D  $(u_i, v_i)$  sur  $H_i$ .
6. Le processus d'extraction est recommencé depuis l'étape 3, à partir de l'horizon  $H_i$ .

Comme le montre la figure 3.11, le champ de fibres ainsi obtenu est bien plus régulier et couvre mieux le volume, la paramétrisation résultante est donc plus stable (figure 3.12). De plus, il n'est plus nécessaire d'avoir une très forte concentration de fibres sur l'horizon de référence, puisque les fibres sont rééchantillonnées sur chaque horizon intermédiaire.

Enfin il est possible, à posteriori, de repérer les tétraèdres qui ne sont traversés par aucune fibre et d'en extraire une. Si elle croise un horizon (l'horizon de référence ou un horizon intermédiaire), alors des valeurs de  $(u, v)$  peuvent y être affectées. Même si elle ne croise pas d'horizon, une contrainte de type *contrainte delta* (voir annexe B, paragraphe B.2.2), avec  $\Delta = 0$ , peut être utilisée entre les différents points de la fibre, pour imposer que  $(u, v)$  soient constants le long de la fibre.

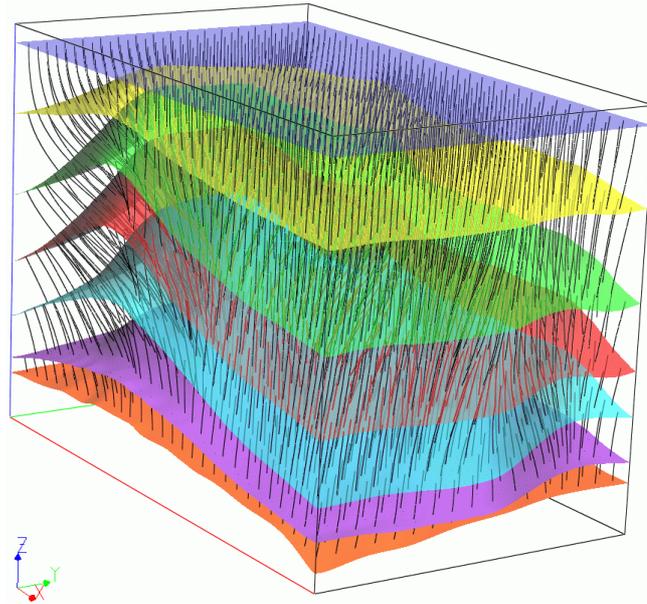


FIG. 3.11 : Contrôle de la densité du champ de fibres par réinitialisation de l'extraction sur chaque horizon. Depuis l'horizon de référence porteur de la paramétrisation 2D (en rouge, au milieu), des fibres sont extraites dans chaque direction jusqu'à l'horizon suivant. La paramétrisation  $(u, v)$  est alors interpolée sur cet horizon, avant d'en extraire un nouveau champ de fibres. Le volume est ainsi couvert par une densité de fibres constante (comparer avec la figure 3.8).

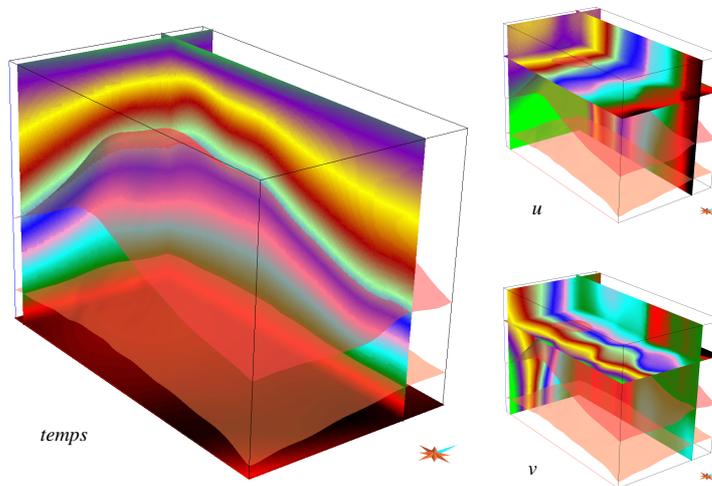


FIG. 3.12 : Paramétrisation GeoChron calculée à partir d'un champ de fibres, sur le jeu de données de la figure 3.11.

### 3.2.3 Édition d'une paramétrisation GeoChron existante

Le concept de lignes iso-paléo-géographiques trouve d'autres applications, en dehors du cadre strict de la construction d'une paramétrisation GeoChron. En effet, à tout mo-

ment, ces lignes forment un outil de visualisation de la paramétrisation, et permettent de regrouper, sur un objet facile à observer, des informations à la fois sur  $u$  et sur  $v$ , qui sont difficiles à visualiser simultanément.

### Construction d'une ligne IPG

Les méthodes présentées précédemment permettent de construire des fibres, basées sur le vecteur **grad**  $t$ , en supposant que ce vecteur n'est pas trop différent des lignes IPG, ce qui n'est vrai que dans certains contextes de déformation.

Une fois une paramétrisation construite, il est possible de remonter à la définition initiale de ces lignes, afin de construire les lignes réelles, et non plus des approximations : les lignes IPG sont définies comme des lignes le long desquelles  $(u, v)$  reste constant, et seul  $t$  varie. On peut donc, dans un volume paramétrisé, suivre ces lignes. Pour cela, nous proposons l'algorithme suivant :

1. Un point initial  $P_0$  est sélectionné (par exemple, désigné par l'utilisateur). Soient  $(u_0, v_0)$  les coordonnées paléo-géographiques de ce point, et  $\mathcal{T}$  le tétraèdre le contenant.
2. À partir des valeurs de  $(u, v)$  aux nœuds de  $\mathcal{T}$ , on détermine par quelle face la ligne IPG  $\mathcal{L}$  va sortir.
3. Le point de sortie exact de  $\mathcal{L}$  dans cette face est déterminé en utilisant les valeurs de  $(u, v)$  à ses sommets.
4. L'algorithme est répété depuis l'étape 2 dans le tétraèdre voisin.
5. Si, à un moment, une des faces est sur une faille, alors le point de sortie et le tétraèdre voisin sont déterminés en utilisant le vecteur rejet.

L'étape la plus complexe est la troisième (le calcul des rejets sera détaillé dans le chapitre suivant). Il s'agit de retrouver les coordonnées  $\mathbf{x}(\alpha) = (x, y, z)$  d'un point  $\alpha$  pour lequel on connaît  $(u, v)$ , définies par ailleurs aux sommets  $\alpha_1, \alpha_2$  et  $\alpha_3$  d'un triangle.

Soient  $\omega_1$  et  $\omega_2$  les coordonnées barycentriques de  $\alpha$  dans le triangle ([Mallet, 2002]) :

$$\mathbf{x}(\alpha) = (1 - \omega_1 - \omega_2) \cdot \mathbf{x}(\alpha_0) + \omega_1 \cdot \mathbf{x}(\alpha_1) + \omega_2 \cdot \mathbf{x}(\alpha_2)$$

De même, les valeurs  $u$  et  $v$  en  $\alpha$  s'expriment en fonction de  $\omega_1$  et  $\omega_2$  :

$$\begin{aligned} u(\alpha) &= (1 - \omega_1 - \omega_2) \cdot u(\alpha_0) + \omega_1 \cdot u(\alpha_1) + \omega_2 \cdot u(\alpha_2) \\ v(\alpha) &= (1 - \omega_1 - \omega_2) \cdot v(\alpha_0) + \omega_1 \cdot v(\alpha_1) + \omega_2 \cdot v(\alpha_2) \end{aligned}$$

Soit :

$$\begin{aligned} u(\alpha) - u(\alpha_0) &= \omega_1 \cdot (u(\alpha_1) - u(\alpha_0)) + \omega_2 \cdot (u(\alpha_2) - u(\alpha_0)) \\ v(\alpha) - v(\alpha_0) &= \omega_1 \cdot (v(\alpha_1) - v(\alpha_0)) + \omega_2 \cdot (v(\alpha_2) - v(\alpha_0)) \end{aligned}$$

Trouver les coordonnées barycentriques  $(\omega_1, \omega_2)$  de  $\alpha$  dans le triangle revient donc à résoudre le système linéaire  $2 \times 2$  suivant :

$$\begin{bmatrix} u(\alpha_1) - u(\alpha_0) & u(\alpha_2) - u(\alpha_0) \\ v(\alpha_1) - v(\alpha_0) & v(\alpha_2) - v(\alpha_0) \end{bmatrix} \cdot \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} u(\alpha) - u(\alpha_0) \\ v(\alpha) - v(\alpha_0) \end{bmatrix}$$

Ce processus est relativement coûteux puisque le système ci-dessus doit être résolu pour chaque tétraèdre traversé par chaque ligne IPG extraite mais, en pratique, cet algorithme est utilisé le plus souvent pour visualiser un très petit nombre de lignes IPG particulières (figure 3.13), et le calcul est alors instantané, car chaque ligne ne recoupe que peu de tétraèdres.

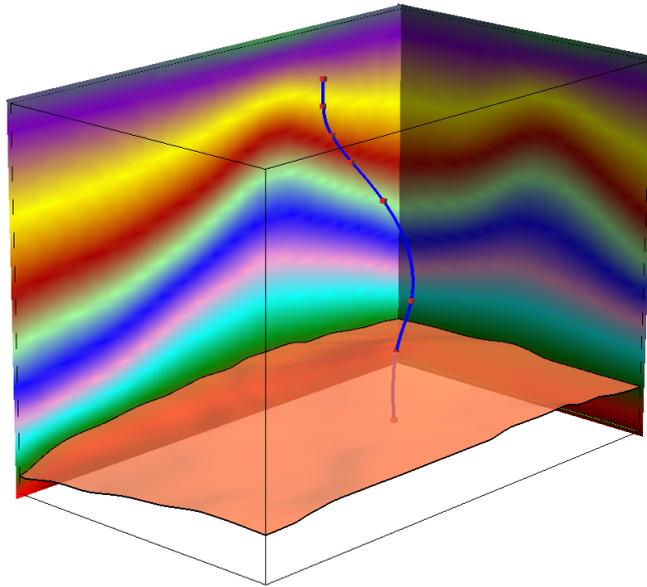


FIG. 3.13 : Ligne IPG (en bleu) extraite dans un volume paramétrisé depuis un horizon donné. Le paramètre temps est affiché sur le volume correspondant au modèle de la figure 3.11. Les points rouges le long de la ligne IPG représentent les intersections avec les horizons qui ont servi à construire le modèle, dont un seul est représenté ici (en transparence).

### Modification de la forme d'une ligne IPG

Les lignes IPG ne sont pas limitées à la visualisation de la paramétrisation, elles peuvent aussi être utilisées pour éditer une paramétrisation pré-existante, par exemple pour mieux s'adapter à un contexte local.

Ainsi, on peut imaginer qu'une fois extraite avec la méthode décrite ci-dessus, une ligne IPG peut être modifiée, avec des outils interactifs comme ceux présents dans le géomodeleur GOCAD (voir par exemple [Caumon, 2003] où de tels outils, appliqués à des surfaces, sont décrits). Un utilisateur pourrait de la sorte forcer une ligne à passer par un point donné, pour respecter des données connues par ailleurs, ou pour introduire un style de déformation particulier dans une zone du modèle.

Une fois une ou plusieurs lignes éditées de la sorte, une nouvelle interpolation de  $(u, v)$  est nécessaire<sup>1</sup>, afin de produire une nouvelle paramétrisation GeoChron qui prenne en compte ces modifications. Comme ces lignes éditées sont probablement conflictuelles avec les lignes voisines existantes, il est nécessaire de définir un voisinage autour de ces lignes, dans lequel toutes les autres contraintes existantes sur la paramétrisation sont désactivées, ou affaiblies.

Enfin, il est à noter que cette édition peut avoir lieu quelle que soit la méthode qui a été utilisée pour construire la paramétrisation GeoChron, que ce soit la méthode locale décrite ici, la méthode globale de la partie suivante, ou toute autre méthode envisageable.

Il est aussi possible d'envisager d'altérer globalement la totalité de la paramétrisation. Si on considère les lignes IPG comme caractéristiques de la paramétrisation, on peut par exemple appliquer à chacune de ces lignes une inclinaison, définie par deux angles  $\alpha$  et  $\beta$  (azimut et pendage) par rapport à un horizon de référence.

### Conservation de la paramétrisation indépendamment de tout maillage

Les lignes IPG peuvent aussi servir à conserver une paramétrisation, quelle que soit la manière dont elle a été calculée, indépendamment de tout maillage. Ainsi, il devient possible de modifier la grille, tout en conservant la même paramétrisation, c'est-à-dire la même correspondance entre un point de l'espace géologique et de l'espace de dépôt.

En particulier, dans le cadre des simulations d'écoulements, les méthodes de résolution numérique des équations de mécanique des fluides sont grandement facilitées si les interfaces entre cellules sont orthogonales aux directions d'écoulement. En conséquence, les maillages sur lesquels ces simulations sont calculées sont fortement dépendants de la nature des écoulements ([Heinemann et Heinemann, 2003]).

Durant l'exploitation d'un réservoir, les différents paramètres (différences de pression, saturation en huile, etc.) qui régissent les écoulements varient au cours du temps, et les directions d'écoulement aussi. Certains auteurs ont donc proposé de faire varier la géométrie des grilles d'écoulement au cours du temps ([Mlacnik *et al.*, 2004]).

Cependant, si une paramétrisation GeoChron est bâtie sur une grille d'écoulement, afin de faire correspondre au mieux celle-ci avec le modèle pétrophysique, il est nécessaire de conserver la paramétrisation lorsque la géométrie de la grille est modifiée.

Pour cela, on peut procéder comme suit :

1. Sur la grille initiale, construire une paramétrisation GeoChron, en utilisant n'importe quelle méthode.
2. Lorsqu'il est nécessaire de modifier la grille, construire un ensemble de lignes IPG, à une résolution cohérente avec celle de la grille. En tout point de ces lignes, on conserve les valeurs de  $(u, v, t)$ .
3. Reconstruire la grille en tenant compte des nouveaux paramètres d'écoulement (voir [Mlacnik *et al.*, 2004]).

---

<sup>1</sup>Cette modification peut aussi être faite sur les fibres construites depuis le champ du gradient de  $t$ , et donc avant toute interpolation de  $(u, v)$ .

4. Utiliser les lignes IPG pour réinterpoler  $(u, v, t)$  aux nœuds de la nouvelle grille.

Aussi longtemps que la géométrie du modèle initial, c'est-à-dire des failles et des horizons, ne change pas, il est donc possible d'utiliser les lignes IPG pour mémoriser une paramétrisation GeoChron indépendamment de tout maillage. Cette paramétrisation peut être à tout moment reproduite sur n'importe quel maillage cohérent avec le modèle géométrique initial.

### 3.3 Approche globale par contraintes

Dans la première approche, on considérait un ensemble de lignes, assimilées à des lignes iso-paléo-géographiques. Cette méthode est en quelque sorte un échantillonnage du domaine d'étude par les lignes, suivi d'une interpolation globale. Chaque ligne étant calculée indépendamment de ses voisines, la méthode en elle-même n'impose pas de cohérence globale, celle-ci vient uniquement du champ de *temps* sous-jacent.

Dans un deuxième temps, nous avons donc cherché une méthode alternative, qui permettrait de calculer les composantes  $(u, v)$  de la paramétrisation GeoChron en une seule fois, en assurant une cohérence globale. Pour cela, nous nous sommes basés sur les formules initiales du modèle, et nous avons cherché des contraintes DSI susceptibles d'exprimer au mieux les relations globales entre les différentes composantes de la paramétrisation. Ces relations sont complexes, et dépendent fortement du style de déformation, et nous avons envisagé deux approches différentes.

On rappelle que, comme dans le cadre de la construction par le biais des fibres, le paramètre *temps* est d'abord calculé, puis il est considéré comme fixé pour le calcul de  $(u, v)$ . Par conséquent, dans les équations suivantes, on considère  $t(\mathbf{x})$ , ou le vecteur  $\mathbf{grad} t(\mathbf{x})$ , comme fixes et connus.

#### 3.3.1 La contrainte d'orthogonalité

Le paragraphe 2.3.3 indique que, dans le cas d'une déformation par flexure simple d'une plaque mince, la déformation est telle que les trois vecteurs gradients de  $u(\mathbf{x})$ ,  $v(\mathbf{x})$  et  $t(\mathbf{x})$  sont orthogonaux deux à deux.

Par conséquent, si on est capable, en tout point du volume, d'imposer cette relation entre les valeurs de  $u(\mathbf{x})$ ,  $v(\mathbf{x})$  et  $t(\mathbf{x})$ , la paramétrisation résultante modélisera une déformation continue du domaine, de type *pure bending*.

La contrainte à respecter, appelée *contrainte d'orthogonalité 3D*, est la suivante :

$$\left\{ \begin{array}{l} \mathbf{grad} u \perp \mathbf{grad} v \\ \mathbf{grad} v \perp \mathbf{grad} t \\ \mathbf{grad} t \perp \mathbf{grad} u \\ \|\mathbf{grad} u\| = \|\mathbf{grad} v\| \end{array} \right. \quad (3.1)$$

Voyons maintenant comment cette contrainte peut s'exprimer dans le formalisme des contraintes acceptables par l'interpolateur DSI. Cette contrainte a été établie en collaboration avec J.-L. Mallet.

### Contrainte DSI

Considérons un tétraèdre  $T(\alpha_0, \alpha_1, \alpha_2, \alpha_3)$ , sur lequel une propriété  $t$  est connue aux sommets, et deux propriétés  $u$  et  $v$  sont à interpoler. On définit le vecteur  $\mathbf{N}$ , de norme 1, comme :

$$\mathbf{N} = \frac{\mathbf{grad} t}{\|\mathbf{grad} t\|}$$

Dans un premier temps, on peut remarquer que :

$$\mathbf{grad} v = \mathbf{N} \times \mathbf{grad} u \iff \begin{cases} \mathbf{grad} v \perp \mathbf{grad} u \\ \mathbf{grad} v \perp \mathbf{grad} t \\ \|\mathbf{grad} u\| = \|\mathbf{grad} v\| \end{cases}$$

D'autre part, on a aussi :

$$\mathbf{grad} u \cdot \mathbf{N} = 0 \iff \mathbf{grad} u \perp \mathbf{grad} t$$

En combinant les deux équations précédentes, on obtient deux équations linéaires qui permettent de définir toutes les relations cherchées :

$$\begin{cases} \mathbf{grad} v = \mathbf{N} \times \mathbf{grad} u \\ \mathbf{grad} u \cdot \mathbf{N} = 0 \end{cases} \iff \begin{cases} \mathbf{grad} u \perp \mathbf{grad} v \\ \mathbf{grad} v \perp \mathbf{grad} t \\ \mathbf{grad} t \perp \mathbf{grad} u \\ \|\mathbf{grad} u\| = \|\mathbf{grad} v\| \end{cases}$$

D'après l'équation B.4 (annexe B), et la définition des coefficients  $\mathbf{D}(\alpha_i)$  (équation B.3), on peut écrire le produit vectoriel de  $\mathbf{N}$  par  $\mathbf{grad} u$  en fonction des  $\mathbf{D}(\alpha_i)$  et  $u(\alpha_i)$  :

$$\mathbf{N} \times \mathbf{grad} u = \sum_{i=0}^3 \underbrace{\mathbf{N} \times \mathbf{D}(\alpha_i)}_{\mathbf{R}(\alpha_i)} \cdot u(\alpha_i)$$

La première équation de cette contrainte peut donc se reformuler ainsi :

$$\begin{aligned} \mathbf{grad} v = \mathbf{N} \times \mathbf{grad} u &\iff \sum_{i=0}^3 \mathbf{D}(\alpha_i) \cdot v(\alpha_i) = \sum_{i=0}^3 \mathbf{R}(\alpha_i) \cdot u(\alpha_i) \\ &\iff \sum_{i=0}^3 \mathbf{R}(\alpha_i) \cdot u(\alpha_i) - \sum_{i=0}^3 \mathbf{D}(\alpha_i) \cdot v(\alpha_i) = 0 \end{aligned}$$

D'autre part, la deuxième équation de cette contrainte s'exprime (d'après la définition de  $D(\alpha_i|\mathbf{N})$  de l'équation B.4, annexe B) :

$$\mathbf{grad} u \cdot \mathbf{N} = 0 \iff \sum_{i=0}^3 D(\alpha_i|\mathbf{N}) \cdot u(\alpha_i) = 0$$

On obtient alors les quatre contraintes DSI  $c_x(T, \mathbf{N})$ ,  $c_y(T, \mathbf{N})$ ,  $c_z(T, \mathbf{N})$  et  $c_N(T, \mathbf{N})$  suivantes :

$$c_x : \begin{cases} A_{c_x}^u(\alpha) = \mathbf{R}_x(\alpha) & \text{si } \alpha \in \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\} \\ A_{c_x}^v(\alpha) = -\mathbf{D}_x(\alpha) & \text{si } \alpha \in \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\} \\ A_{c_x}^u(\alpha) = 0 & \text{sinon} \\ A_{c_x}^v(\alpha) = 0 & \text{sinon} \\ b_{c_x} = 0 \end{cases}$$

$$c_y : \begin{cases} A_{c_y}^u(\alpha) = \mathbf{R}_y(\alpha) & \text{si } \alpha \in \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\} \\ A_{c_y}^v(\alpha) = -\mathbf{D}_y(\alpha) & \text{si } \alpha \in \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\} \\ A_{c_y}^u(\alpha) = 0 & \text{sinon} \\ A_{c_y}^v(\alpha) = 0 & \text{sinon} \\ b_{c_y} = 0 \end{cases}$$

$$c_z : \begin{cases} A_{c_z}^u(\alpha) = \mathbf{R}_z(\alpha) & \text{si } \alpha \in \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\} \\ A_{c_z}^v(\alpha) = -\mathbf{D}_z(\alpha) & \text{si } \alpha \in \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\} \\ A_{c_z}^u(\alpha) = 0 & \text{sinon} \\ A_{c_z}^v(\alpha) = 0 & \text{sinon} \\ b_{c_z} = 0 \end{cases}$$

$$c_N : \begin{cases} A_{c_N}^u(\alpha) = D(\alpha|\mathbf{N}) & \text{si } \alpha \in \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\} \\ A_{c_N}^u(\alpha) = 0 & \text{sinon} \\ b_{c_N} = 0 \end{cases}$$

Il est à noter que ces deux contraintes agissent simultanément sur  $u$  et sur  $v$ , d'où les coefficients  $A^u$  et  $A^v$ , qui se réfèrent respectivement à  $u$  et  $v$ .

### 3.3.2 Le conformal mapping étendu

L'approche précédente reproduit une déformation continue, de type *pure bending*. Nous avons cependant vu qu'il existait un autre style de déformation, le *flexural slip* ou glissement banc sur banc. De plus, nous avons vu dans le paragraphe 2.3.4 que ce type de déformation pouvait être formulé dans le volume.

Pour cela, il faut que la projection de la paramétrisation GeoChron sur chaque horizon, c'est-à-dire sur chaque surface d'isovaleur du *temps*, forme une paramétrisation de type *conformal mapping 2D* de  $H_t$ . Nous avons en effet indiqué, au paragraphe 3.1.2, que cette paramétrisation est celle qui minimise les déformations nécessaires pour transformer la surface en un plan. Comme on admet généralement que la solution la plus proche de la réalité pour déformer un objet naturel de son état initial à son état actuel est celle qui minimise les déformations, la paramétrisation par *conformal mapping* est donc appropriée.

En termes mathématiques, si on note  $\mathbf{grad}_H u$  et  $\mathbf{grad}_H v$  les projections de  $\mathbf{grad} u$  et  $\mathbf{grad} v$  sur un horizon quelconque  $H$ , on cherche donc à avoir :

$$\left| \begin{array}{l} \mathbf{grad}_H u \perp \mathbf{grad}_H v \\ \|\mathbf{grad}_H u\| = \|\mathbf{grad}_H v\| \end{array} \right. \quad (3.2)$$

la première relation pouvant aussi se formuler comme  $\mathbf{grad}_H u \cdot \mathbf{grad}_H v = 0$ .

Par analogie avec la paramétrisation des surfaces, nous appelons cette contrainte volumique *conformal mapping étendu*<sup>2</sup>.

### Contrainte DSI

Nous allons maintenant exprimer la relation ci-dessus sous forme d'une contrainte DSI dans un tétraèdre  $T(\alpha_0, \alpha_1, \alpha_2, \alpha_3)$ , dépendante des valeurs de  $u$  et de  $v$  uniquement. Cette contrainte a été établie par J.-L. Mallet.

Dans un premier temps, décomposons  $\mathbf{grad}_H u$  et  $\mathbf{grad}_H v$  en fonctions linéaires de  $\mathbf{grad} u$  et  $\mathbf{grad} v$ . Pour cela, on peut choisir une paire quelconque de vecteurs unitaires  $\mathbf{U}$  et  $\mathbf{V}$  orthogonaux, et tangents à l'horizon  $H$  dans  $T$ . Les vecteurs  $\mathbf{grad}_H u$  et  $\mathbf{grad}_H v$  se décomposent alors :

$$\left| \begin{array}{l} \mathbf{grad}_H u = (\mathbf{grad} u \cdot \mathbf{U}) \cdot \mathbf{U} + (\mathbf{grad} u \cdot \mathbf{V}) \cdot \mathbf{V} \\ \mathbf{grad}_H v = (\mathbf{grad} v \cdot \mathbf{U}) \cdot \mathbf{U} + (\mathbf{grad} v \cdot \mathbf{V}) \cdot \mathbf{V} \end{array} \right.$$

Les vecteurs  $\mathbf{U}$  et  $\mathbf{V}$  peuvent être calculés, par exemple, à partir d'un vecteur unitaire  $\mathbf{N}$  normal à l'horizon  $H$  :

$$\mathbf{N} = \frac{\mathbf{grad} t}{\|\mathbf{grad} t\|} = \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix} \Rightarrow \left| \begin{array}{l} \mathbf{U} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} \text{ avec : } \begin{cases} i \in \{1, 2, 3\} \\ U_i = 0 \\ U_{(i+1) \bmod 3} = a, a \in \mathbb{R} \\ U_{(i+2) \bmod 3} = -\frac{N_{(i+1) \bmod 3}}{N_{(i+2) \bmod 3}} \cdot U_{(i+1) \bmod 3} \end{cases} \\ \mathbf{V} = \mathbf{N} \times \mathbf{U} \end{array} \right.$$

Les équations 3.2 peuvent alors se formuler :

$$\left| \begin{array}{l} \mathbf{grad}_H u \cdot \mathbf{grad}_H v = 0 \\ \|\mathbf{grad}_H u\| = \|\mathbf{grad}_H v\| \end{array} \right. \iff \left| \begin{array}{l} \mathbf{grad} u \cdot \mathbf{U} = \mathbf{grad} v \cdot \mathbf{V} \\ \mathbf{grad} u \cdot \mathbf{V} = -\mathbf{grad} v \cdot \mathbf{U} \end{array} \right.$$

Ces relations peuvent à leur tour être transformées de façon à faire apparaître les valeurs de  $u$  et  $v$  aux nœuds du tétraèdre  $T$  :

$$\begin{aligned} \sum_{i=0}^3 D(\alpha_i | \mathbf{U}) \cdot u(\alpha_i) - D(\alpha_i | \mathbf{V}) \cdot v(\alpha_i) &= 0 \\ \sum_{i=0}^3 D(\alpha_i | \mathbf{V}) \cdot u(\alpha_i) + D(\alpha_i | \mathbf{U}) \cdot v(\alpha_i) &= 0 \end{aligned}$$

<sup>2</sup>Notons toutefois qu'il ne s'agit pas d'une paramétrisation conforme en trois dimensions, donc d'un *conformal mapping 3D*, mais plutôt de l'extension en trois dimensions d'une paramétrisation conforme en deux dimensions.

ou encore, sous la forme de deux contraintes DSI  $c(T, \mathbf{U}, \mathbf{V})$  et  $\bar{c}((T, \mathbf{U}, \mathbf{V}))$  :

$$\begin{array}{l}
 c : \left\{ \begin{array}{ll}
 A_c^u(\alpha) = D(\alpha|\mathbf{U}) & \text{si } \alpha \in \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\} \\
 A_c^v(\alpha) = -D(\alpha|\mathbf{V}) & \text{si } \alpha \in \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\} \\
 A_c^u(\alpha) = 0 & \text{sinon} \\
 A_c^v(\alpha) = 0 & \text{sinon} \\
 b_c = 0 & \\
 \end{array} \right. \\
 \\
 \bar{c} : \left\{ \begin{array}{ll}
 A_{\bar{c}}^u(\alpha) = D(\alpha|\mathbf{V}) & \text{si } \alpha \in \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\} \\
 A_{\bar{c}}^v(\alpha) = D(\alpha|\mathbf{U}) & \text{si } \alpha \in \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\} \\
 A_{\bar{c}}^u(\alpha) = 0 & \text{sinon} \\
 A_{\bar{c}}^v(\alpha) = 0 & \text{sinon} \\
 b_{\bar{c}} = 0 & \\
 \end{array} \right.
 \end{array}$$

Comme pour l'orthogonalité 3D, ces deux contraintes agissent simultanément sur  $u$  et sur  $v$ , d'où les coefficients  $A^u$  et  $A^v$ .

### 3.3.3 Résultats

Nous présentons ici quelques résultats de l'utilisation des deux contraintes d'orthogonalité 3D, et de *conformal mapping* étendu. Le jeu de données choisi pour illustrer ces contraintes est géologiquement simple, puisqu'il s'agit d'un anticlinal (données fournies par l'Institut Français du Pétrole, dans le cadre du projet CARFRAC) présenté sur la figure 3.14. L'anticlinal de Split Mountain est situé dans le Dinosaur National Monument, Utah. Il s'intègre dans le cadre d'un épisode compressif dans la région de la Cordillère, daté aux environs de la fin du Mésozoïque et du début du Tertiaire. L'anticlinal de Split Mountain lui-même s'est formé à l'occasion du soulèvement des Eastern Unita Mountains, à l'Éocène, et se compose principalement d'un axe est-ouest, avec vers l'est un axe secondaire vers le nord. Il a été particulièrement étudié en tant qu'analogie de terrain à des réservoirs fortement affectés par la fracturation (voir [Silliphant *et al.*, 2002], [Wilkins et Gross, 2002], [Guiton, 2001]).

L'intérêt d'un tel jeu de données simple est qu'il est plus facile de vérifier la cohérence des résultats. Toutefois, les chapitres suivants présenteront d'autres exemples plus complexes, et en particulier intégrant des failles (chapitre 4).

La procédure suivante a été utilisée pour calculer le modèle GeoChron sur ces données :

1. À partir des surfaces initiales, un volume tétraédrique les englobant a été construit, ainsi que le modèle topologique associé (figure 3.14).
2. À chaque horizon du modèle initial a été affecté une valeur de *temps* choisie arbitrairement (figure 3.15).
3. Le paramètre *temps* a été interpolé dans le volume par DSI, en utilisant des contraintes de type *point de contrôle* (voir annexe B.2.1) pour tous les nœuds des surfaces, ainsi qu'une contrainte de *gradient constant* (voir annexe B, paragraphe B.3.2) dans le volume (figure 3.15).

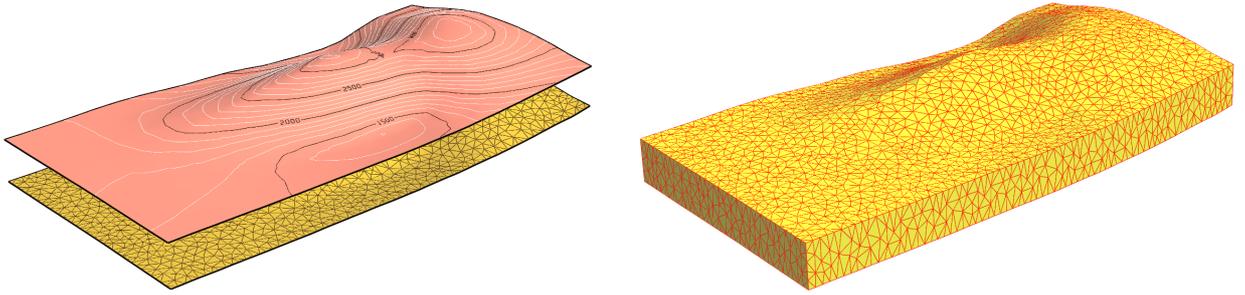


FIG. 3.14 : Modèle initial de l'anticlinal de Split Mountain, Utah (données IFP), exagération verticale  $\times 2$ . À gauche, les deux horizons triangulés reconstitués à partir des données de terrain, avec des courbes de niveau sur l'horizon supérieur et le maillage sur l'horizon inférieur. À droite, le volume tétraédrisé correspondant.

4. Un des horizons, celui du milieu, a été choisi comme horizon de référence, et une paramétrisation par *conformal mapping* a été calculée dessus (figure 3.16).
5. Enfin,  $u(\mathbf{x})$  et  $v(\mathbf{x})$  ont été interpolés par DSI dans le volume, avec les contraintes suivantes :
  - des points de contrôle sont installés le long de l'horizon de référence ;
  - une contrainte de gradient constant assure la continuité de la solution ;
  - pour reproduire un style de déformation donné, la contrainte correspondante (orthogonalité 3D ou *conformal mapping* étendu) est ajoutée.

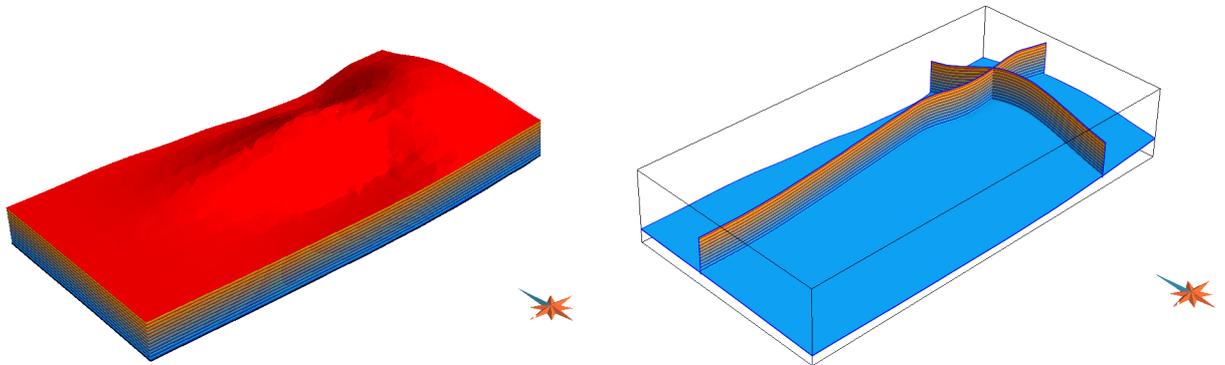


FIG. 3.15 : Calcul du paramètre  $t$  sur le modèle de Split Mountain, Utah (exagération verticale  $\times 2$ ). Un temps constant a été affecté à chacun des horizons de la figure 3.14. À gauche, le volume tétraédrisé peint avec le temps, à droite, deux coupes verticales dans le modèle et une surface d'isovaleur de  $t$  mettant en évidence la courbure du modèle.

Notons que, numériquement, les deux contraintes nécessitent d'interpoler simultanément  $u(\mathbf{x})$  et  $v(\mathbf{x})$ , c'est-à-dire deux propriétés différentes. Cependant, comme expliqué dans l'annexe A, l'interpolateur DSI est parfaitement adapté à des propriétés vectorielles à plusieurs champs. En pratique, on utilisera donc une propriété vectorielle à deux champs, le premier représentant  $u$  et le second  $v$ . Tous les résultats présentés dans cette section, et par la suite, représentent toutefois séparément  $u$  et  $v$ .

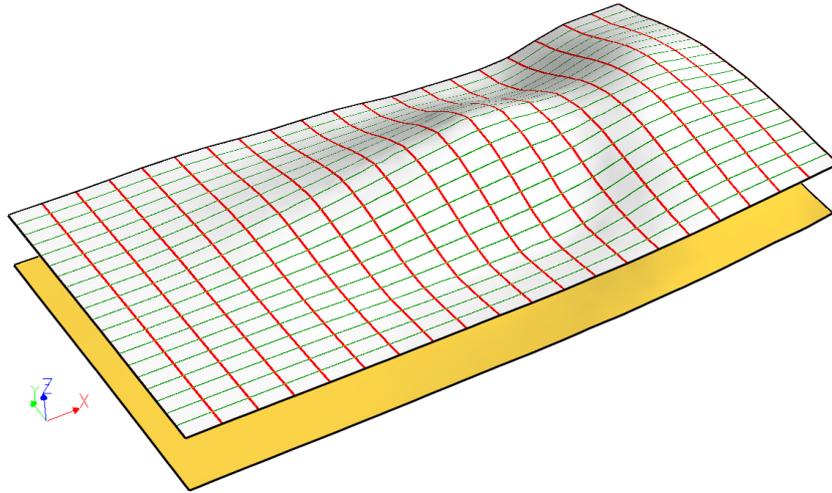


FIG. 3.16 : Paramétrisation de l'horizon de référence du modèle de Split Mountain, Utah. Les lignes rouges et vertes figurent des isovaleurs de  $u$  et  $v$  (exagération verticale  $\times 2$ ).

Les figures 3.17 et 3.18 présentent les résultats des contraintes d'orthogonalité 3D, et de *conformal mapping* étendu. Ces résultats semblent cohérents avec les données, et ne montrent pas d'erreur majeure.

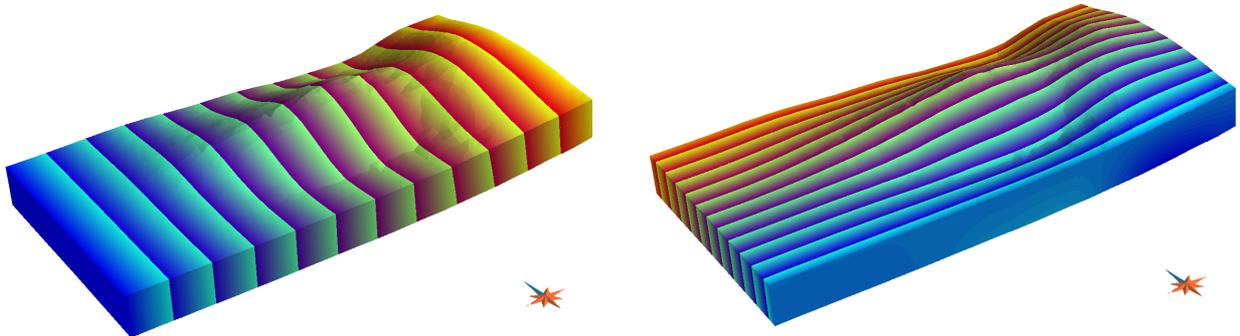


FIG. 3.17 : Paramétrisation  $(u, v)$  obtenue par la contrainte d'orthogonalité 3D (à gauche,  $u(\mathbf{x})$ , à droite  $v(\mathbf{x})$ ), exagération verticale  $\times 2$ . Les lignes de couleur représentent des lignes d'isovaleurs. Ces lignes sont équidistantes sur l'horizon de référence (figure 3.16).

Si on compare avec les résultats obtenus par la première méthode, utilisant un champ de fibres (figure 3.19), la paramétrisation obtenue est très similaire à celle obtenue par la contrainte d'orthogonalité. En effet, l'hypothèse de base du tracé de fibre est que les lignes IPG sont orthogonales en tout point aux horizons, ce qui correspond à l'hypothèse d'une déformation de type *pure bending*.

De plus, la figure 3.20 montre une section particulière du volume, perpendiculairement à l'axe du pli. On y voit clairement la différence entre les deux styles de déformations : les lignes de couleur figurent des surfaces d'isovaleur de  $u$  ou de  $v$ , elles permettent donc de mesurer des distances le long des surfaces, en sachant qu'avant déformation, l'écart entre chaque paire de ligne était le même partout.

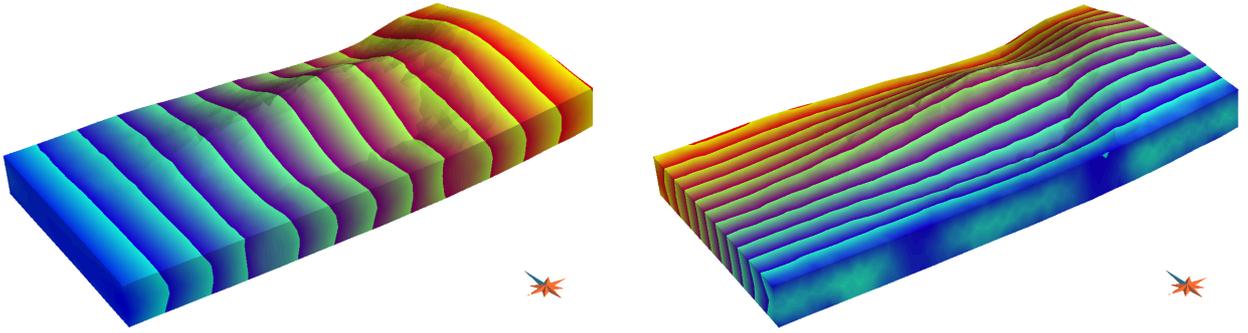


FIG. 3.18 : Paramétrisation  $(u, v)$  obtenue par la contrainte de conformal mapping étendu (à gauche,  $u(\mathbf{x})$ , à droite  $v(\mathbf{x})$ ), exagération verticale  $\times 2$ . Les lignes de couleur représentent des lignes d'isovaleurs. Ces lignes sont équidistantes sur l'horizon de référence (figure 3.16).

Dans le cas d'une déformation par *pure bending*, la distance varie suivant la position verticale de l'horizon dans le volume. À l'inverse, dans le cas du *flexural slip*, les distances le long de chaque horizon restent à peu près constantes, ce qui est conforme aux caractéristiques de la déformation.

Notons toutefois qu'un certain nombre d'artefacts apparaissent dans le cadre de la paramétrisation par *conformal mapping* étendu (figure 3.18, sur le flanc sud pour le paramètre  $v$ , ou encore figure 3.20, sur la flanc nord de la section). Ces instabilités peuvent être provoquées par des effets de bord : la contrainte de gradient constant appliquée sur les tétraèdres du bord du volume peut engendrer des déséquilibres. Plus globalement, la contrainte de *conformal mapping* étendu est numériquement plus instable, car elle met en jeu un plus grand nombre d'opérations numériques (produits scalaires et vectoriels), et elle se base sur les vecteurs  $\mathbf{U}$  et  $\mathbf{V}$ , dont le calcul peut être instable (en particulier si  $\text{grad } t = 0$ ).

## Conclusions et perspectives

Dans ce chapitre, nous avons présenté plusieurs méthodes permettant de construire concrètement une paramétrisation GeoChron, en respectant et en utilisant les définitions théoriques du chapitre précédent. Ces méthodes se décomposent en deux grandes familles, les méthodes locales et les méthodes globales.

Nous avons vu que l'efficacité de ces méthodes ne dépendait pas uniquement de la rigueur des formules mathématiques sur lesquelles elles reposent, mais aussi des différents choix faits lors de l'implémentation. Nous avons ainsi étudié l'influence des différents maillages possibles, avant de choisir les maillages non structurés tétraédriques. Nous avons aussi mis en place des algorithmes de densification ou de ré-échantillonnage, dans le cadre des fibres, qui permettent d'obtenir un résultat exploitable dans la plupart des cas.

Il est aussi à noter que ce chapitre ne prétend nullement être exhaustif sur les méthodes de construction d'une paramétrisation GeoChron. Bien d'autres solutions sont envisageables, soit en utilisant d'autres outils numériques, soit encore pour répondre à

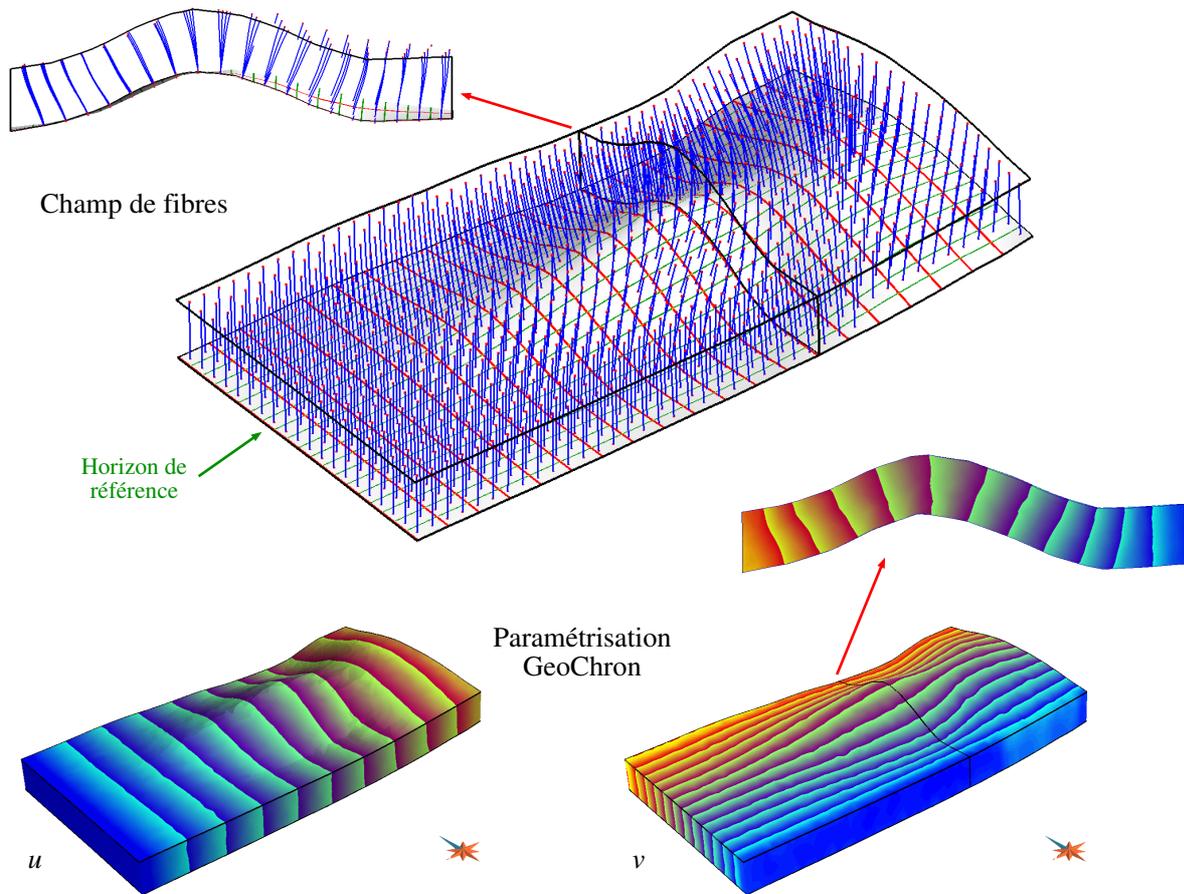


FIG. 3.19 : Paramétrisation du modèle de Split Mountain par la méthode des fibres (exagération verticale  $\times 2$ ). En haut, champ de fibres extraites depuis l'horizon de référence. La coupe montre la courbure des fibres au centre du pli. En bas, paramétrisation GeoChron correspondante (seuls  $(u, v)$  sont représentés,  $t$  étant calculé indépendamment de la méthode de construction). La coupe dans la paramétrisation indique un style en pure bending (voir figure 3.20).

des problèmes plus spécifiques que ceux abordés ici (par exemple, pour aligner les coordonnées paléo-géographiques  $(u, v)$  sur des directions sédimentologiques ou structurales particulières, etc.).

L'approche locale nous a permis de construire des approximations des lignes IPG, dans un contexte de déformation en *pure bending* uniquement. Bien que limitée à ce contexte, cette approche est intéressante, car elle permet d'introduire l'objet « lignes IPG », qui peut ensuite être utilisé pour affiner localement la paramétrisation. De plus, la plupart des opérations de cette méthode s'implémentent avec des outils classiques de modélisation numérique. Enfin, cette méthode permet d'obtenir rapidement une solution initiale valide dans la plupart des cas.

L'autre approche, plus globale, consiste à déterminer des contraintes DSI valables en tout point du domaine, en accord avec le style de déformation du volume. Cette méthode

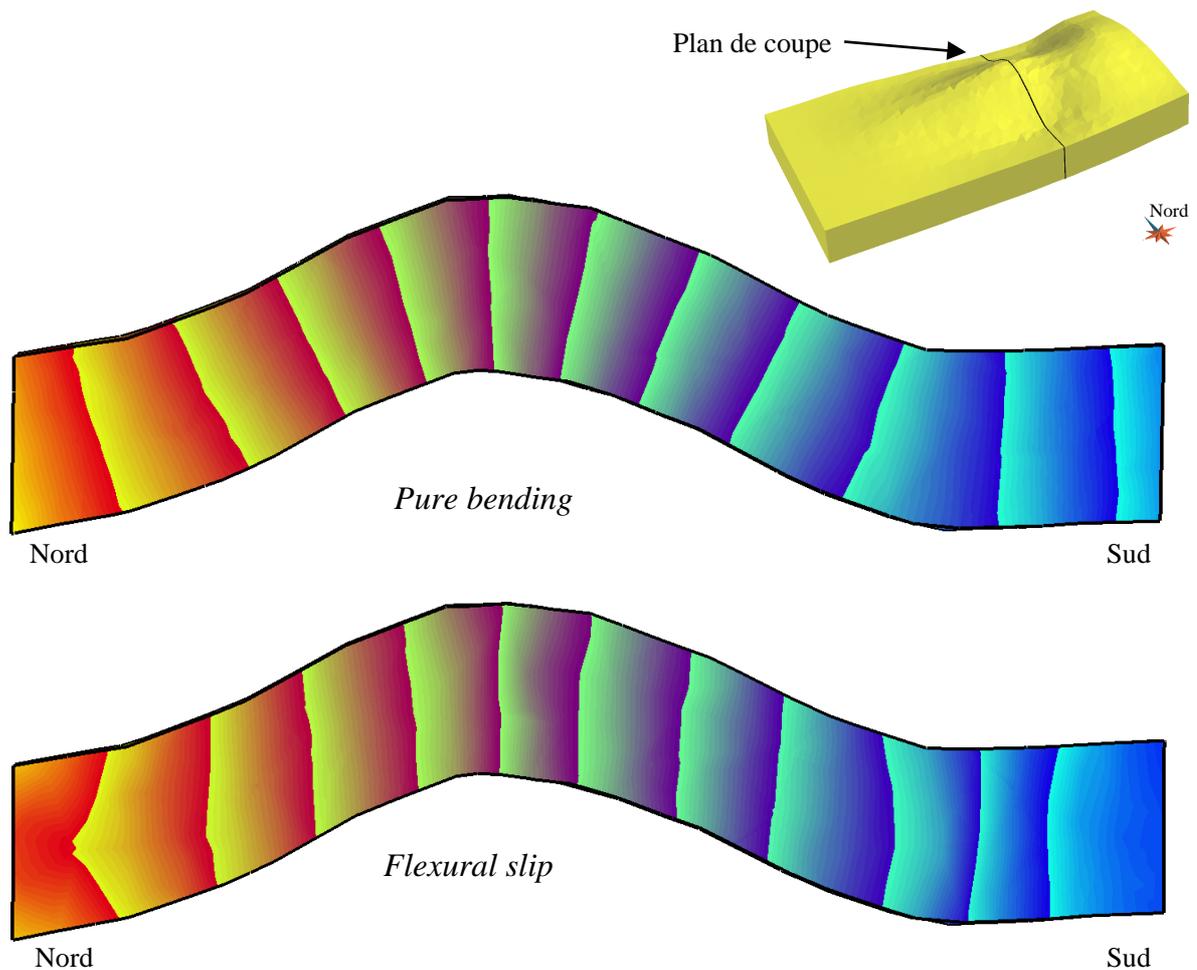


FIG. 3.20 : Comparaison en coupe (exagération verticale  $\times 2$ ) entre les deux contraintes de paramétrisation globale. En haut, la paramétrisation obtenue par orthogonalité 3D, en bas par conformal mapping étendu. Les lignes sont similaires à celles observées suivant les styles de déformation (figures 2.10 et 2.12).

est donc plus souple, mais elle nécessite d'être capable à priori de déterminer le style de déformation dominant dans les terrains étudiés.

### Styles de déformation intermédiaires

Quelle que soit la méthode utilisée, seuls les deux styles de déformation extrêmes opposés sont modélisés. Or, dans la nature, il est rare que les problèmes soient aussi simples. Dans certains cas, ces approximations sont valables, mais la plupart du temps, la réalité se situe entre les deux.

Il est toutefois possible d'affiner une paramétrisation GeoChron existante, afin de prendre en compte des styles de déformation intermédiaires. Pour cela, nous suggérons de calculer une paramétrisation correspondant à chaque extrême, et d'utiliser des coeffi-

cients de pondération pour combiner les deux. Une phase d'homogénéisation est ensuite nécessaire, car rien ne garantit à priori la cohérence des deux paramétrisations.

Cette voie permet non seulement d'obtenir une infinité de solutions intermédiaires, mais aussi de faire varier les fonctions de mélange au sein du domaine, permettant ainsi de privilégier un style de déformation en un point, et un autre ailleurs. De plus, certains critères géomécaniques issus de la paramétrisation, comme le tenseur de déformation ou les directions de fracturation, peuvent être calculés en temps réel. L'utilisateur peut alors interactivement explorer l'espace des fonctions de mélange jusqu'à trouver la solution qui correspond le mieux aux données (par exemple des directions de fracturation mesurées dans des puits ou observées en surface).

D'autre part, les outils d'édition locale des lignes IPG décrits précédemment permettent d'affiner encore plus la paramétrisation, et d'intégrer des informations plus précises sur les déformations.

### Intégration des deux méthodes de construction

Nous avons introduit dans ce chapitre deux méthodes distinctes. Cependant, elles reposent toutes les deux sur les mêmes bases théoriques, et ont un certain nombre de points communs. Il est donc tout à fait possible d'utiliser partiellement les deux voies dans un même processus.

Ainsi, dans des contextes de déformation complexe, on peut envisager le processus suivant :

- En utilisant une méthode globale correspondant au style dominant, une première paramétrisation est construite.
- Parallèlement, quelques fibres sont construites, en des points particuliers du domaine. Ces fibres permettent, en particulier, d'assurer la cohérence de la paramétrisation au travers des failles. En effet, le vecteur rejet tel qu'il sera détaillé dans le chapitre suivant n'est pas toujours suffisant pour assurer une bonne propagation des valeurs. Quelques fibres permettent alors de mieux contraindre l'interpolation.
- Optionnellement, des lignes IPG peuvent être extraites depuis la paramétrisation initiale. Ces lignes peuvent ensuite être modifiées par l'utilisateur, si besoin est, pour mieux s'ajuster aux données.
- Enfin, la paramétrisation est ré-interpolée dans le volume, en prenant en compte non seulement les contraintes globales, mais aussi les fibres extraites précédemment, et les lignes IPG modifiées.

Enfin, la plupart des exemples présentés ici ne montrent que des domaines non-faillés. Nous avons évoqué la question de l'intégration des failles dans la construction des fibres, mais sans détailler la méthode de calcul des vecteurs rejet utilisée. Le chapitre suivant est consacré à ce problème, et présente une méthode de calcul des rejets, basée sur le paramètre *temps*, ainsi que son intégration, aussi bien dans le cadre du calcul d'une paramétrisation GeoChron que plus généralement.