En tout premier lieu, je désire remercier Monsieur le professeur Mallet, directeur du Laboratoire Informatique et Analyse des Données de l'École de Géologie de Nancy et responsable de l'équipe Infographie du Centre de Recherche Informatique de Nancy, pour le temps qu'il m'a consacré et la confiance qu'il su m'accorder. Durant les deux années du cycle ingénieur de l'École de Géologie de Nancy, et les trois années de thèse il m'a orienté avec son enthousiasme, sa gentillesse et sa rigueur intellectuelle. Grâce au consortium GOCAD qu'il a mis en place, j'ai pu bénéficier d'un certain nombre d'avantages tant sur la qualité du matériel mis à la disposition de ses étudiants, que sur l'opportunité de parcourir le monde au gré des congrès internationaux.

Ma reconnaissance s'adresse tout particulièrement à Monsieur René SCHOTT, Professeur à l'Université Henri Poincaré (Nancy-I). Je lui dois de précieuses suggestions lors de la rédaction de ce mémoire, et je le remercie d'avoir accepté d'être rapporteur de cette thèse.

J'aimerais remercier Monsieur Nicolas CHEIMANOFF, Maître de Recherches à l'École des Mines de Paris, et Monsieur Hervé PERROUD, Professeur à l'Université de Pau et des Pays de l'Adour, qui ont gentiment accepté d'être les rapporteurs extérieurs de ce mémoire.

Je désire remercier Monsieur Jean-Claude PAUL, Professeur à l'École d'Architecture de Nancy, d'avoir accepté de juger ce travail.

Je tiens à remercier Monsieur Pierre BOUCHET, Maître de Conférences à l'Université Henri Poincaré (Nancy-I) pour ses nombreux conseils et ses nombreuses suggestions lors de la rédaction de ce mémoire, je le remercie aussi d'avoir accepté de juger ce mémoire.

Je remercie aussi Monsieur Naâmen KESKES, Expert Scientifique à Elf Aquitaine, qui a été le premier à m'avoir parlé des problèmes de modélisation géométrique des failles, il a aussi accepté de juger ce mémoire.

Dans le cadre de la thèse, j'ai passé un mois dans les locaux de la compagnie pétrolière **UNOCAL 76** à *Anaheim* en *Californie*. Là-bas j'ai fait la connaissance de Monsieur Fred AMINZADEH qui m'a gentiment accueilli dans son service, j'ai aussi fait la connaissance de Monsieur F. V. CORONA qui m'a parlé des problèmes qu'il a rencontrés lors de la modélisation géomé-

triques des failles. A eux deux je voudrais dire Merci. Monsieur J. C. DULAC, Ingénieur de Recherches dans la même compagnie, m'a conseillé lors de la conception du prototype de génération automatique de modèles géologiques, je lui présente mes remerciements, ainsi qu'à sa famille (Nathalie, Guillaume, Camille et Clément) qui m'a invité à plusieurs reprises pendant mon séjour aux états-unis.

Ce mémoire n'aurait pas vu le jour sans l'aide de Y. CHIPOT, M. CUGURNO, C. FAY, M. LECLAIRE, M. AYT OUGOUGDAL, R. COGNOT, J. CONRAUD, B. GERARD, R. HAKKOU, P. LAVEST, P. JACQUEMIN, P. SAMSON. Je les remercie tous d'avoir accepté de lire et relire cette thèse.

Mes remerciements vont aussi à tous mes amis, je sais que leur gentillesse, leur sympathie et leur soutien ont permis l'aboutissement de ce travail de thèse. Il sont tellement nombreux, et cela me réjouis, qu'une page ne suffira pas pour les citer tous.

Enfin, je tiens à remercier les membres du consortium GOCAD, qui supportent ce projet avec enthousiasme et intelligence :

Universités

Bonn University
Colorado School of Mines
Harc
Freiburg University
Institut de Physique du Globe
Imperial College London
Karlsruhe University
École de Géologie de Nancy
Osservatorio Geofisico Trieste
École des Mines de Paris
Politecnico Milano
Princeton University
Stanford University
T.N.O. Institut
Trieste University

Compagnies

Amoco
C.G.G.-Petrosystems
Chevron
Conoco
Elf-Aquitane
Exxon
Gaz de France
Institut Français du Pétrole
Landmark
Noranda
Philips Petroleum
Shell Research
Statoil
Tensor P.G.S.
Total
Unocal

Résumé

La modélisation géométrique de surfaces géologiques est un domaine en plein essor dont l'objectif est de représenter, de façon précise, des surfaces géologiques complexes. Les approches basées sur les méthodes de $C.A.O.$ classiques n'étant pas suffisamment adaptées pour la modélisation de telles surfaces, le Professeur Mallet a proposé une nouvelle approche de la modélisation des surfaces naturelles dans le cadre du projet GOCAD. Le noyau du projet réside dans un nouveau principe d'interpolation en 3 dimensions appelé $D.S.I.$ (Discrete Smooth Interpolation).

Les travaux présentés dans ce mémoire furent menés au sein de ce projet, et consistaient à développer un ensemble d'outils géométriques pour la construction automatique de modèles géologiques. Parmi ces outils, on trouve le calcul des courbes d'isovaleurs sur des surfaces complexes et la modélisation des failles. La nouvelle approche, pour le calcul des courbes d'isovaleurs sur des surfaces, se distingue par sa rapidité et sa précision même si les surfaces sont complexes (par exemple présentant des trous). La modélisation d'une faille passe par la construction d'une surface et par la caractérisation de la relation qu'elle entretient avec un horizon. Cette dernière opération a été rendue possible par la mise en place de trois contraintes géométriques $D.S.I.: OnTsurf$, $VecLink$ et $OnBorder$.
Enfin, les outils de la modélisation géométrique, développés dans le cadre du projet GOCAD , seront utilisés pour mettre au point un prototype d'un générateur automatique de modèles géologiques.

Abstract

Geometrical modeling of geological surfaces is a promising research area, aimed to represent complex geological surfaces. The classical $C.A.D.$ methods are not well adapted to model those complex surfaces, for that reason Professor Mallet introduced a new approach for modeling *natural surfaces* in the frame of the GOCAD project. This project is based on a new interpolation method called $D.S.I.$ (Discrete Smooth Interpolation).

My work consisted of developing a set of geometrical tools for the automatic building of geological model, in the frame of the GOCAD projet. Among those tools, I will present the computation of the *isovalue lines*, and the geometrical modeling of the faults. The new approach for the computation of the *isovalue lines* is fast and accurate even if the surface is complex (for example presenting holes). The modeling of a fault consists in building a surface and characterization of the fault-horizon relationship, thanks to three geometrical $D.S.I.$ constraints : $OnTsurf$, $VecLink$ and $OnBorder$.
The tools provided by the GOCAD project will be used for creating a prototype of the automatic builder of geological models.

Introduction

Les gisements pétroliers et miniers, économiquement exploitables, sont actuellement recherchés à des profondeurs importantes et dans des zones géologiquement compliquées. Les forages de reconnaissance de ces gisements sont de plus en plus rares en raison de leur coût et il est nécessaire de connaître avec précision et en un temps minimum la disposition des couches géologiques avant le début de l'exploitation d'un gisement. Par exemple, en raison du coût élevé d'un forage de pompage de pétrole ou d'extraction des minerais, une erreur de positionnement d'un tel forage peut s'avérer catastrophique. Depuis les années 1970, des générateurs automatiques de modèles géologiques, rapides et précis, ont été développés dans le cadre de la cartographie automatique. Malheureusement, ce type de modèle $2D\frac{1}{2}$ n'est pas adapté à la modélisation 3D de surfaces géologiques complexes. Une nouvelle approche a donc été choisie.

Le travail de cette thèse s'inscrit dans le cadre des recherches entreprises pour la mise en place d'un système semi-automatique, de type $C.A.O.$, permettant de générer des modèles géologiques. La mise en place d'un tel générateur a nécessité le développement de plusieurs outils de modélisation et de représentation $3D$. Ainsi, le chapitre 1 présente l'un des outils les plus utilisés par les géologues pour décrire la morphologie des surfaces géologiques complexes et la variation des propriétés physiques, il s'agit des courbes d'isovaleurs. L'approche que nous proposons permet d'extraire des courbes d'isovaleurs à partir de surfaces géologiques de forme géométrique et de topologie complètement générales. Ces courbes d'isovaleurs seront utilisées comme moyen de contrôle des résultats obtenus dans les chapitre 4 et 5. Après avoir présenté, dans les chapitres 2 et 3, la méthode d'interpolation qui sera utilisée pour modéliser les surfaces naturelles ainsi que quelques notions de géologie, nous allons présenter dans le chapitre 4 une nouvelle approche pour la modélisation des failles géologiques. Une faille est une surface de rupture pouvant

couper les autres surfaces géologiques. L'importance de la faille réside dans ses caractéristiques intrinsèques qui lui permettent de modifier le comportement des surfaces géologiques coupées. Trois types de contraintes DSI sont alors mis en place pour modéliser le comportement d'une surface géologique coupée par une faille. Il s'agit de la contrainte $OnTsurf$ qui permet aux bords d'une surface de glisser le long d'une autre surface, de la contrainte $VecLink$ qui permet de contrôler le déplacement engendré par une faille sur les bords des surfaces qu'elle coupe et de la contrainte $OnBorder$ qui permet à un nœud d'une surface de glisser le long du bord d'une autre surface. Dans le chapitre 5, nous proposons un prototype de générateur automatique de modèles géologiques, utilisant un ensemble d'outils de modélisations géométriques de surfaces naturelles, dont ceux présentés dans les chapitres 1 et 4, ainsi que l'interpolateur DSI présenté dans le chapitre 2. Un test de ce prototype, sur des données réelles, permet d'apprécier l'intérêt de l'approche proposée.

Table des matières

Chapitre 1

Calcul et tracé de courbes d'isovaleur sur une surface triangulée

Ce mémoire étant à priori dédié à la construction automatique de modèles géologiques, on peut se demander quel est l'intérêt des courbes d'isovaleur dans un tel contexte? La réponse est simple, la notion de courbes d'isovaleur d'une propriété φ attachée à une surface géologique S est le moyen le plus précis que nous ayons trouvé pour visualiser un certain nombre de paramètres intervenant au cours de cette modélisation. Parmi ces paramètres, on peut citer :

- une estimation locale de l'erreur d'ajustement d'une surface à des données, un exemple de cette utilisation sera présenté dans le chapitre 2,
- l'intensité du rejet d'une faille (cf chapitre 4),
- ...

Comme le montre la figure 1.1, les courbes d'isovaleur, selon les trois directions (x, y, z) de l'espace, sont également très utiles pour mettre en évidence la forme géométrique d'une surface complexe. Par exemple, les courbes de niveau se rapprochent aux endroits où la surface présente un fort gradient

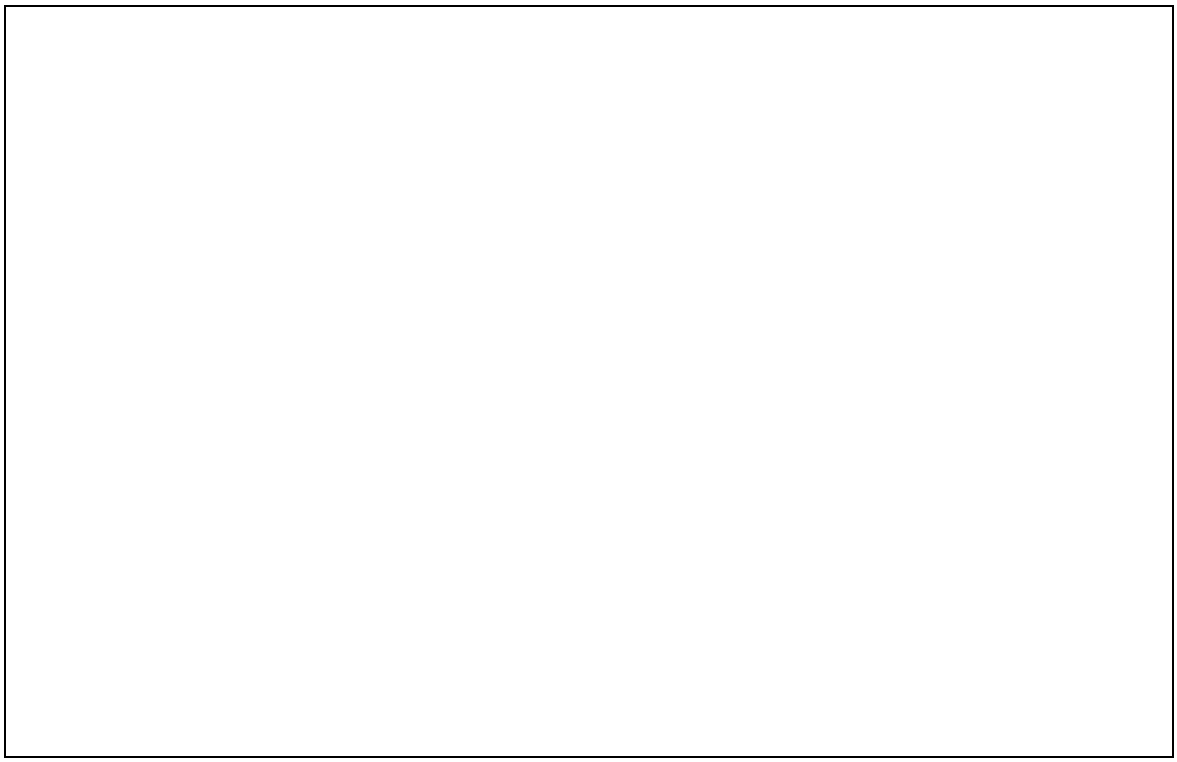

FIG. 1.1 – *L'espacement des courbes de niveau est fonction de la géométrie de la surface*

vertical (falaise par exemple), dans les zones planes les courbes de niveau sont espacées.
Ce chapitre se décompose en deux paragraphes :

- Rappel du principe et des limitations des méthodes classiques de calcul des courbes d'isovaleur dans le cadre de la cartographie numérique.
- Présentation de la nouvelle approche adoptée lors du calcul des courbes d'isovaleur sur des surfaces de forme géométrique générale.

1.1 Courbes d'isovaleur en cartographie numérique

La cartographie numérique est l'ensemble des techniques permettant de visualiser les variations d'une fonction continue $\phi = \varphi(u, v)$ définie sur une

partie compacte D de l'espace euclidien bidimensionnel R^2([38])

Parmi les méthodes de cartographie numérique les plus utilisées dans le domaine des sciences de la terre on trouve *les courbes d'isovaleur*. Ces courbes permettent de représenter les variations d'un paramètre $\phi = \varphi(u, v)$ dépendant des deux paramètres u et v. Les algorithmes classiquement utilisés, en cartographie numérique, pour résoudre ce problème ([38]), interprètent (u, v, ϕ) comme étant les coordonnées (x, y, z) d'un point dans l'espace $3D$ et supposent que l'on ait :

$$\begin{array}{rcccl} x & = & x(u,v) & = & u \\ y & = & y(u,v) & = & v \\ z & = & z(u,v) & = & \phi \end{array}$$

Dans cet espace $3D$, le graphe de la fonction $\varphi(x, y)$ est alors une surface S dite "univoque", c'est à dire que toute droite parallèle à l'axe Oz ne coupe cette surface qu'en, au plus, un seul point. Si l'on suppose que cette surface ne présente aucun plateau d'altitude ϕ_0, alors l'intersection $C(\phi_0)$ du plan horizontal d'altitude ϕ_0 avec la surface S est une courbe dont la projection $C_\varphi(\phi_0)$, dans le plan $(x, y) \equiv (u, v)$ est appelée courbe "isovaleur" de ϕ.

1.1.1 Interprétation physique

Il est facile de donner une interprétation physique au modèle des courbes de niveau ; par exemple, supposons que :

- le plan $(x, y) \equiv (u, v)$ est une feuille de métal parfaitement plane conductrice de l'électricité,
- en un certain nombre N de points du plan $(x, y) \equiv (u, v)$, on place des électrodes ayant des potentiels $\{\phi_1, \phi_2, \cdots, \phi_N\}$.

Les électrodes ainsi disposées engendrent dans le plan $(x, y) \equiv (u, v)$ un potentiel $\varphi(x, y) \equiv \varphi(u, v)$, et pour tout potentiel ϕ_0 fixé, la courbe d'isovaleur $C_\varphi(\phi_0)$ apparaît comme le lieu des points du plan pour lesquels le potentiel est égal à ϕ_0 (cf figure 1.2).

1.1.2 Limitations de la cartographie numérique

Reprenons l'exemple précédent et supposons que :

- le plan (u, v) est une couche géologique très mince,

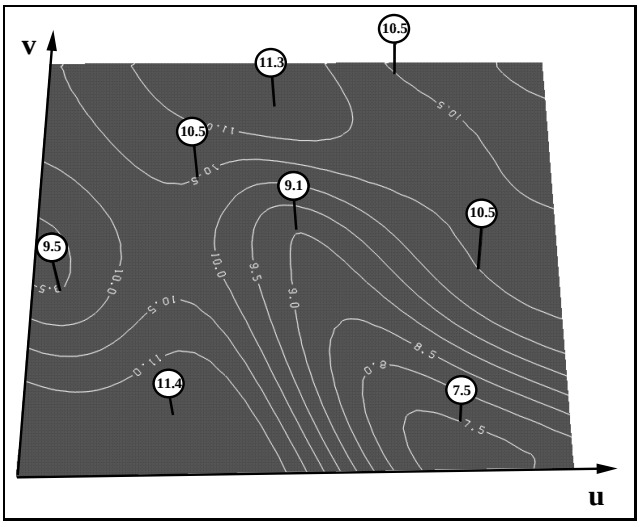

FIG. 1.2 – *Interprétation physique des courbes d'isovaleur, les lignes verticales correspondent aux électrodes et le chiffre attaché à chaque électrode correspond à la valeur du potentiel. Les courbes blanches correspondent aux lignes d'isopotentiel*

- le potentiel $\varphi(u, v)$ représente un paramètre physique attaché à la couche, comme par exemple :
  - Le potentiel hydraulique du pétrole contenu dans la couche,
  - la perméabilité de la couche,
  - une incertitude sur la position de la couche,
  - ...

Comme le suggère la figure 1.3, les surfaces géologiques sont rarement assimilables à un plan S, tant en ce qui concerne leur géométrie que leur topologie :

- du point de vue géométrique, les surfaces géologiques ont été plissées, ceci implique que S n'est pas une surface plane,
- du point de vue topologique, elles ont été découpées par un réseau de failles, ceci implique que S peut avoir des trous ([39][48]).

Il s'ensuit que les méthodes de cartographie numérique ne sont pas bien adaptées à ce type de problème. Au début des années 80, un certain nombre

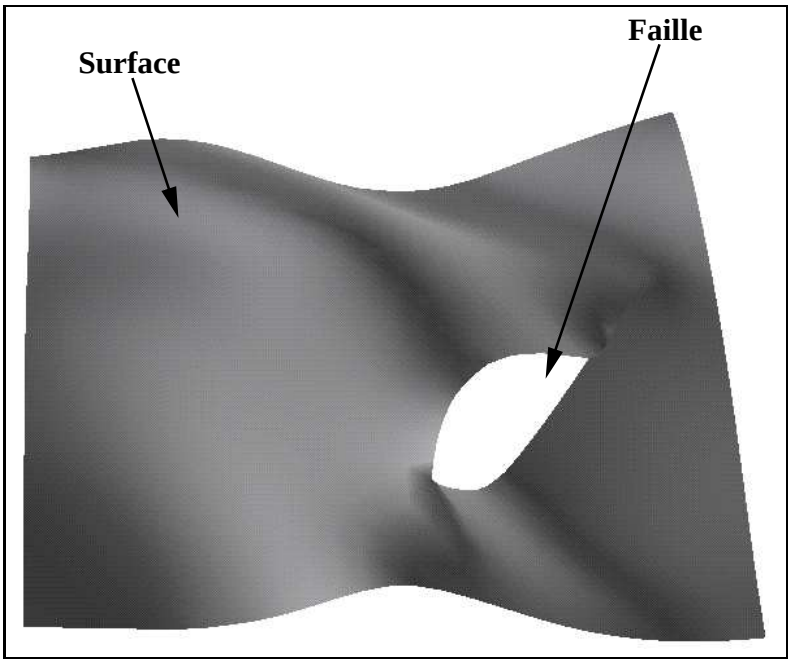

FIG. 1.3 – *Exemple d'une surface plissée découpée par une faille*

de méthodes, en général très compliquées, ont été proposées pour essayer de tenir compte des discontinuités engendrées par les failles. Ces méthodes ont toujours été basées sur des modèles de type $\phi = \varphi(x, y)$, ce qui interdit de modéliser un paramètre $\varphi(u, v)$ attaché à une surface gauche paramétrée par des coordonnées curvilignes (u, v).
Pour résoudre ces problèmes nous avons développé une nouvelle approche qui sera présentée dans le paragraphe suivant.

1.2 Présentation de la nouvelle approche

Il s'agit d'une approche très générale et relativement simple permettant de modéliser et cartographier les variations d'un paramètre $\varphi(u, v)$ attaché à une surface géologique S plissée et découpée par un réseau de failles. Comme le suggère la figure 1.4, nous supposons que la surface S est une surface "triangulée", c'est à dire constituée par un ensemble de triangles adjacents,

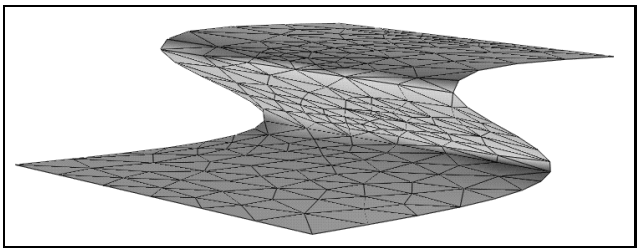

FIG. 1.4 – *Exemple d'une surface triangulée*

et nous supposons de plus que :

- la géométrie de S n'est connue que par la position dans l'espace $3D$ des sommets $\mathbf{p}_i$ des triangles qui la composent,
- le paramètre φ défini sur S n'est connu que par ses valeurs φ_i en chaque sommet $\mathbf{p}_i$ des triangles de S.

Il est facile de voir que, malgré sa simplicité, un tel modèle permet de tenir compte de n'importe quelle géométrie ou topologie d'une surface. Du point de vue pratique, il faut noter que ce modèle suppose implicitement que la taille des triangles a été localement ajustée à la complexité de la géométrie et aux variations de φ ; ainsi des petits triangles sont utilisés :

- aux endroits où la surface présente une géométrie complexe,
- aux endroits où φ varie rapidement (même si à ces endroits la géométrie est assimilable à un plan).

1.2.1 Les surfaces triangulées

Une surface triangulée est constituée d'un ensemble de triangles. Pour chaque triangle $t = T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$, l'information concernant la géométrie et la propriété φ est concentrée aux sommets $(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ et nous n'avons aucune indication concernant la géométrie à l'intérieur de $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ et les variations de φ à l'intérieur de ce même triangle. En conséquence, il nous faut, à partir de ces seules données ponctuelles, "reconstituer" une représentation continue de ce qui se passe à l'intérieur de chaque triangle $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$.
Dans le paragraphe suivant nous présenteront quelques modèles de représentation des triangles.

1.2.2 Notion de patch triangulaire

Dans la suite de ce chapitre le mot *patch*, couramment utilisé dans le domaine de la modélisation géométrique, désigne une facette.
Considérons le repère orthonormé $(O|u,v)$ défini dans $\mathbf{R}^2$, on désigne par "triangle paramétrique unitaire" l'ensemble Θ des points de $\mathbf{R}^2$ dont les coordonnées (u,v) vérifient la condition suivante :

$$(u,v) \in \Theta \quad \Longleftrightarrow \quad \begin{cases} u \in [0,1] \\ v \in [0,1] \\ u+v \leq 1 \end{cases}$$

Comme l'illustre la figure 1.5(a), Θ est un triangle dont les sommets sont respectivement $(0,0)$, $(1,0)$ et $(0,1)$.

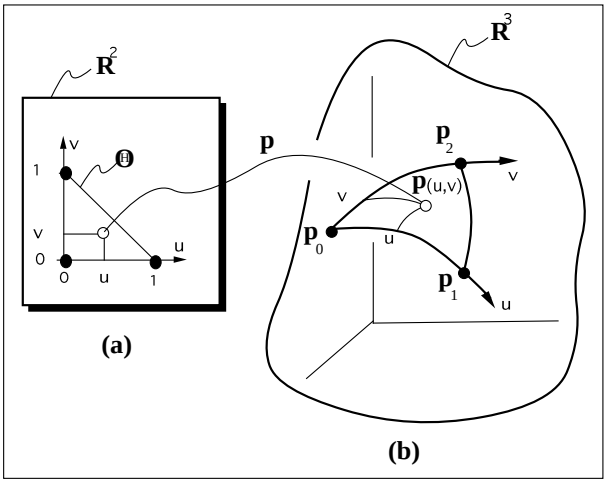

FIG. 1.5 – *Cartographie du triangle paramétrique unitaire Θ de R^2 (figure a) sur le patch triangulaire $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ de R^3 (figure b).*

Comme l'illustre la figure 1.5 (b), si $\mathbf{p}_0$, $\mathbf{p}_1$ et $\mathbf{p}_2$ sont trois points de $\mathbf{R}^3$ et si $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ est l'image du triangle paramétrique unitaire Θ par une application $p(u,v)$ de R^2 vers R^3 telle que :

$$\begin{aligned} p(0,0) &= \mathbf{p}_0 \\ p(1,0) &= \mathbf{p}_1 \\ p(0,1) &= \mathbf{p}_2 \end{aligned}$$

On dit que $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ est un patch triangulaire **régulier** de R^3, ou simplement un **patch triangulaire** ou un **patch**, si et seulement si :

$$\left| \begin{array}{l} \left\{ \begin{array}{ll} p(u,v) & \text{est continue sur } \Theta \\ \frac{\partial p(u,v)}{\partial u} & \text{est continue sur } \Theta \\ \frac{\partial p(u,v)}{\partial v} & \text{est continue sur } \Theta \end{array} \right. \\ \\ \frac{\partial p(u,v)}{\partial u} \times \frac{\partial p(u,v)}{\partial v} \neq \mathbf{0} \quad \text{sur } \Theta \end{array} \right.$$

La continuité de $p(u,v)$ garantit que les points $\mathbf{p}_0$, $\mathbf{p}_1$ et $\mathbf{p}_2$ appartiennent au bord de la facette, ces points seront appelés **vertices** de $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$.

Étude du bord d'un patch triangulaire

Le bord d'un patch triangulaire $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ est composé de trois courbes continues $E(\mathbf{p}_0, \mathbf{p}_1)$, $E(\mathbf{p}_0, \mathbf{p}_2)$ et $E(\mathbf{p}_1, \mathbf{p}_2)$ telles que :

$$E(\mathbf{p}_i, \mathbf{p}_j) = \text{ courbe qui joint } \mathbf{p}_i \text{ à } \mathbf{p}_j$$

On peut vérifier que :

$$\left| \begin{array}{lcl} E(\mathbf{p}_0, \mathbf{p}_1) & = & \text{image du côté } \{(0,0),(1,0)\} \text{ de } \Theta \\ E(\mathbf{p}_0, \mathbf{p}_2) & = & \text{image du côté } \{(0,0),(0,1)\} \text{ de } \Theta \\ E(\mathbf{p}_1, \mathbf{p}_2) & = & \text{image du côté } \{(1,0),(0,1)\} \text{ de } \Theta \end{array} \right.$$

Coordonnées curvilignes

Comme le montre la figure 1.5, les images des axes $(O|u)$ et $(O|v)$ de R^2 sont deux courbes orientées de R^3 correspondant aux côtés $E(\mathbf{p}_0, \mathbf{p}_1)$ et $E(\mathbf{p}_0, \mathbf{p}_2)$. Ces courbes sont donc appelées "axes curvilignes " et les paramètres (u,v) correspondants sont appelés "coordonnées curvilignes" de $p(u,v)$.

Tangentes p^u et p^v

Lors de la définition du patch curviligne nous avons supposé que $p(u,v)$ est dérivable par rapport à u et v. Dans la suite, p^u et p^v correspondent à ces deux dérivées :

$$\left| \begin{array}{l} p^u(u,v) = \dfrac{\partial p(u,v)}{\partial u} \\ \\ p^v(u,v) = \dfrac{\partial p(u,v)}{\partial v} \end{array} \right.$$

Ces deux vecteurs sont tangents au patch curviligne $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ au point $p(u, v)$ (cf figure 1.6), on dira alors que :

- $p^u(u, v)$ est tangent à $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ selon la direction u,
- $p^v(u, v)$ est tangent à $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ selon la direction v

Normale $\mathbf{N}(u, v)$

La normale $N(u, v)$ (cf figure 1.6) en tout point $p(u, v)$ du patch curviligne $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ est donnée par la formule suivante :

$$N(u, v) = \frac{p^u(u, v) \times p^v(u, v)}{||p^u(u, v) \times p^v(u, v)||} \tag{1.1}$$

Cette normale existe si et seulement si les deux vecteurs $p^u(u, v)$ et $p^v(u, v)$ ne sont pas colinéaires. Comme le montre la figure 1.6, les normales aux nœuds $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ de la facette curviligne sont les suivantes :

$$\left| \begin{array}{lcl} \mathbf{N}_0 & = & N(0,0) \\ \mathbf{N}_1 & = & N(1,0) \\ \mathbf{N}_2 & = & N(0,1) \end{array} \right.$$

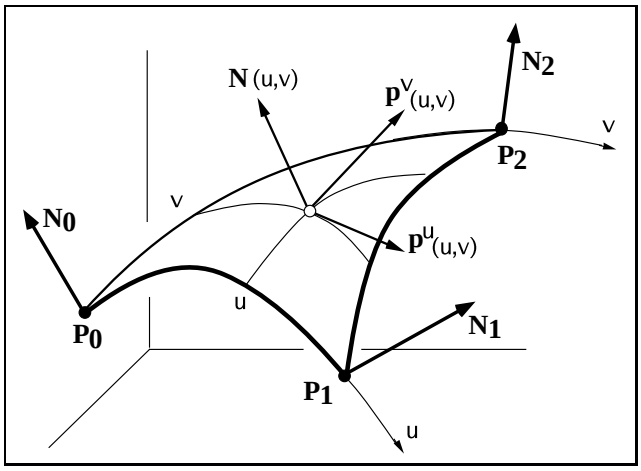

FIG. 1.6 – *Tangentes et normales à la facette curviligne* $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ *de* $\mathbf{R}^3$.

Calcul de la tangente au bord du patch

Supposons que T_{ij} soit la tangente en $\mathbf{p}_i$ au côté $E(\mathbf{p}_i, \mathbf{p}_j)$, orientée positivement de $\mathbf{p}_i$ vers $\mathbf{p}_j$. Il existe une infinité de tangentes T_{ij} qui respectent la condition suivante :

$$T_{ij} \cdot \mathbf{N}_i = 0$$

En 1987 Nielson ([47]) propose la formule suivante pour le calcul de T_{ij} :

$$T_{ij} = \frac{\{N_i \times (\mathbf{p}_j - \mathbf{p}_i)\} \times \mathbf{N}_i}{||\{\mathbf{N}_i \times (\mathbf{p}_j - \mathbf{p}_i)\} \times \mathbf{N}_i||}$$

Cette formule suppose que $(\mathbf{p}_j - \mathbf{p}_i)$ et N_i ne sont pas colinéaires. En pratique, cette condition est toujours vérifiée.

La formulation de la fonction $p(u, v)$ permet de définir plusieurs types de patch, nous nous proposons d'en étudier deux :

- patch linéaire,
- patch de Bézier.

1.2.2.1 Patch linéaire

Il s'agit de la représentation la plus simple des patchs, la fonction $p(u, v)$ associée à ce type de patch est définie par la formule suivante :

$$\left| \begin{array}{l} p(u,v) \quad = \quad \mathbf{p}_0 + u \cdot \mathbf{U} + v \cdot \mathbf{V} \\ \\ \text{avec : } \left\{ \begin{array}{rcl} \mathbf{U} & = & \mathbf{p}_1 - \mathbf{p}_0 \\ \mathbf{V} & = & \mathbf{p}_2 - \mathbf{p}_0 \end{array} \right. \end{array} \right.$$

Comme le montre la figure 1.7, le patch défini par $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ correspond à une facette plane et le côté $E(\mathbf{p}_i, \mathbf{p}_j)$ correspond au segment joignant les deux nœuds $\mathbf{p}_i$ et $\mathbf{p}_j$. Les vecteurs tangents $p^u(u, v)$, $p^v(u, v)$ et la normale $N(u, v)$ sont définis par :

$$\begin{array}{rcl} p^u(u,v) & = & \mathbf{U} \\ p^v(u,v) & = & \mathbf{V} \\ \mathbf{N}(u,v) & = & \frac{\mathbf{U} \times \mathbf{V}}{||\mathbf{U} \times \mathbf{V}||} \end{array}$$

$T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ sera appelé **patch triangulaire plan**.

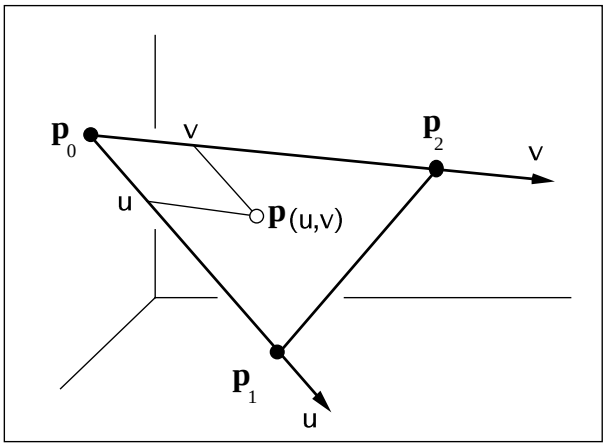

FIG. 1.7 – *Exemple d'une facette triangulaire plane* $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$.

Le patch plan peut aussi être décrit de la manière suivante :

$$p(u,v) = (1-u-v)\cdot \mathbf{p}_0 + u \cdot \mathbf{p}_1 + v \cdot \mathbf{p}_2$$

Si on suppose que

$$\left\{ \begin{array}{lcl} w_0(u,v) & = & 1-u-v \\ w_1(u,v) & = & u \\ w_2(u,v) & = & v \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{lcl} b_0 & = & \mathbf{p}_0 \\ b_1 & = & \mathbf{p}_1 \\ b_2 & = & \mathbf{p}_2 \end{array} \right.$$

La fonction $p(u,v)$ sera donc définie par :

$$p(u,v) = \sum_i^2 w_i(u,v) \cdot b_i$$

Le réseau de contrôle correspond aux trois sommets de la facette, et pour tout $(u,v) \in \Theta$, les coefficients pondérateurs $\{w_i(u,v)\}$ sont non négatifs et leur somme est égale à 1. Le patch triangulaire plan possède donc la propriété "enveloppe convexe"([46]).

Remarques

1. Il est évident que le patch plan interpole les trois sommets :

$$\begin{array}{lcl} p(0,0) & = & \mathbf{p}_0 \\ p(1,0) & = & \mathbf{p}_1 \\ p(0,1) & = & \mathbf{p}_2 \end{array}$$

On conclue que le patch triangulaire plan est une interpolation valable du patch $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ si et seulement si les trois normales $\mathbf{N}_0$, $\mathbf{N}_1$ et $\mathbf{N}_2$ sont égales à $\dfrac{U \times V}{\| U \times V \|}$.

2. Si l'on désigne par $\{\varphi_0, \varphi_1, \varphi_2\}$ les valeurs d'un paramètre φ aux sommets du triangle $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$, alors il est possible d'utiliser le même modèle linéaire pour modéliser les variations de φ en tout point (u, v) à l'intérieur de $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$:

$$\varphi(u, v) = (1 - u - v) \cdot \varphi_0 + u \cdot \varphi_1 + v \cdot \varphi_2$$

Il est important de remarquer que le modèle linéaire présenté ci-dessus, ne permet d'assurer qu'une continuité C^0 lorsqu'on passe d'une facette triangulaire $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ à sa voisine $T(\mathbf{p}_0^{'}, \mathbf{p}_1, \mathbf{p}_2)$ en traversant le côté $E(\mathbf{p}_1, \mathbf{p}_2)$.

1.2.2.2 Patch cubique de Bézier

Le réseau de contrôle associé à un patch cubique de Bézier $\mathcal{B}_3$ est défini par l'ensemble des points $\{ \mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_{01}, \mathbf{p}_{10}, \mathbf{p}_{02}, \mathbf{p}_{20}, \mathbf{p}_{12}, \mathbf{p}_{21}, \mathbf{q} \}$, illustré par la figure 1.8, tel que :

$$\left| \begin{array}{rcl} (\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2) & = & \text{sommets de } T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2) \\ \mathbf{p}_{ij} & = & \mathbf{p}_i + \beta_{ij} \cdot T_{ij} \quad \forall\ i = (0, 1, 2) \\ \mathbf{q} & = & \psi(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_{01}, \mathbf{p}_{10}, \mathbf{p}_{02}, \mathbf{p}_{20}, \mathbf{p}_{12}, \mathbf{p}_{21}) \end{array} \right.$$

Les coefficients β_{ij} sont supposés être définis par :

$$\beta_{ij} = \beta \cdot ||\mathbf{p}_j - \mathbf{p}_i|| \quad \text{avec : } \beta > 0$$

Le choix des coefficients positifs β et de la fonction $\psi()$ a fait l'objet de plusieurs études ([46][43]).

Définition de $p(u, v)$

Par définition, on appelle "patch cubique de Bézier" la facette curviligne $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$, basée sur le réseau de contrôle de Bézier $\mathcal{B}_3$ et dont l'équation

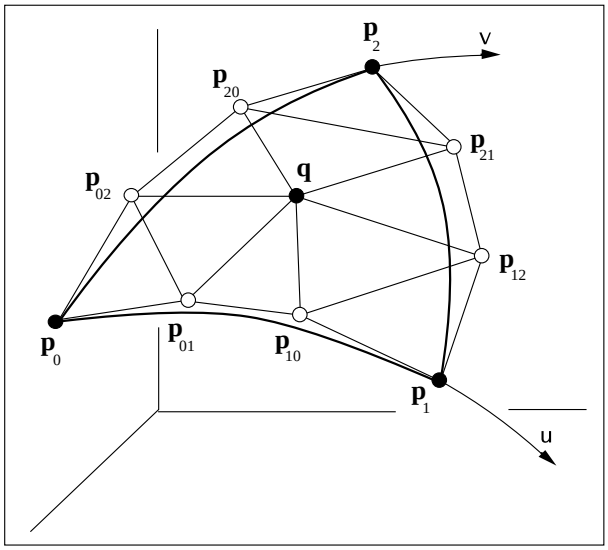

FIG. 1.8 – *Exemple d'un patch triangulaire curviligne* $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ *et du réseau de contrôle de Bézier associé.*

$p(u,v)$ peut s'écrire de la manière suivante, où $w = (1-u-v)$:

$$\begin{aligned} p(u,v) &= w^3 \cdot \mathbf{p}_0 + u^3 \cdot \mathbf{p}_1 + v^3 \cdot \mathbf{p}_2 \\ &+ 3 \cdot (\ uw^2 \cdot \mathbf{p}_{01} + u^2 w \cdot \mathbf{p}_{10}) \\ &+ 3 \cdot (\ vw^2 \cdot \mathbf{p}_{02} + v^2 w \cdot \mathbf{p}_{20}) \\ &+ 3 \cdot (\ vu^2 \cdot \mathbf{p}_{12} + v^2 u \cdot \mathbf{p}_{21}) \\ &+ 6 \cdot uvw \cdot \mathbf{q} \end{aligned}$$

Il est facile de vérifier que le patch de Bézier interpole les trois sommets :

$$\begin{aligned} p(0,0) &= \mathbf{p}_0 \\ p(1,0) &= \mathbf{p}_1 \\ p(0,1) &= \mathbf{p}_2 \end{aligned}$$

$p(u,v)$ peut aussi s'écrire d'une autre façon, de manière à montrer le contrôle de chaque couple de points $(\mathbf{p}_{ij}, \mathbf{p}_{ji})$:

$$\begin{aligned} p(u,v) &= w^3 \cdot \mathbf{p}_0 + u^3 \cdot \mathbf{p}_1 + v^3 \cdot \mathbf{p}_2 \\ &+ 3 \cdot wu \cdot (\ w \cdot \mathbf{p}_{01} + u \cdot \mathbf{p}_{10}\ ) \\ &+ 3 \cdot uv \cdot (\ u \cdot \mathbf{p}_{12} + v \cdot \mathbf{p}_{21}\ ) \\ &+ 3 \cdot vw \cdot (\ w \cdot \mathbf{p}_{02} + v \cdot \mathbf{p}_{20}\ ) \\ &+ 6 \cdot uvw \cdot \mathbf{q} \end{aligned}$$

Calcul des tangentes au patch de Bézier

Les tangentes suivant les directions u et v peuvent être obtenues en dérivant l'équation $p(u,v)$ du patch cubique de Bézier :

$$\begin{aligned} p^u(u,v) = & - 3\cdot w^2 \cdot (\mathbf{p}_0 - \mathbf{p}_{01}) \\ & + 3\cdot u^2 \cdot (\mathbf{p}_1 - \mathbf{p}_{10}) \\ & + 3\cdot v^2 \cdot (\mathbf{p}_{21} - \mathbf{p}_{20}) \\ & + 6\cdot uv \cdot (\mathbf{p}_{12} - \mathbf{q}) \\ & - 6\cdot uw \cdot (\mathbf{p}_{01} - \mathbf{p}_{10}) \\ & - 3\cdot vw \cdot (\mathbf{p}_{02} - \mathbf{q}) \end{aligned}$$

$$\begin{aligned} p^v(u,v) = & - 3\cdot w^2 \cdot (\mathbf{p}_0 - \mathbf{p}_{02}) \\ & + 3\cdot u^2 \cdot (\mathbf{p}_{12} - \mathbf{p}_{10}) \\ & + 3\cdot v^2 \cdot (\mathbf{p}_2 - \mathbf{p}_{20}) \\ & + 6\cdot uv \cdot (\mathbf{p}_{21} - \mathbf{q}) \\ & - 6\cdot uw \cdot (\mathbf{p}_{01} - \mathbf{q}) \\ & - 6\cdot vw \cdot (\mathbf{p}_{02} - \mathbf{p}_{20}) \end{aligned}$$

On peut vérifier que les tangentes aux sommets du patch du Bézier sont données par les relations suivantes :

$$\mathbf{p}_0 \longleftrightarrow \left| \begin{array}{lcl} p^u(0,0) & = & 3\cdot \beta_{01}\cdot T_{01} \\ p^v(0,0) & = & 3\cdot \beta_{02}\cdot T_{02} \end{array} \right.$$

$$\mathbf{p}_1 \longleftrightarrow \left| \begin{array}{lcl} p^u(1,0) & = & -\ 3\cdot \beta_{10}\cdot T_{10} \\ p^v(1,0) & = & 3\cdot(\beta_{12}\cdot T_{12} - \beta_{10}\cdot T_{10}) \end{array} \right.$$

$$\mathbf{p}_2 \longleftrightarrow \left| \begin{array}{lcl} p^u(0,1) & = & 3\cdot(\beta_{21}\cdot T_{21} - \beta_{20}\cdot T_{20}) \\ p^v(0,1) & = & -\ 3\cdot \beta_{20}\cdot T_{20} \end{array} \right.$$

Normales aux nœuds $\mathbf{p}_0$, $\mathbf{p}_1$ et $\mathbf{p}_2$

A partir des relations précédentes, nous pouvons vérifier que le vecteur normale $\mathbf{N}(u,v)$ interpole les normales $\mathbf{N}_0$, $\mathbf{N}_1$ et $\mathbf{N}_2$ aux trois sommets du patch (voir figure 1.9) :

$$\begin{aligned} N(0,0) &= \mathbf{N}_0 \\ N(1,0) &= \mathbf{N}_1 \\ N(0,1) &= \mathbf{N}_2 \end{aligned}$$

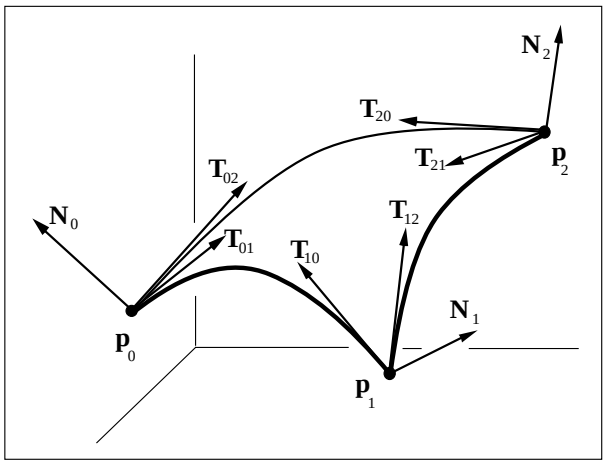

FIG. 1.9 – *Les normales et les tangentes au niveau des sommets des patchs de Bézier*

On en déduit que l'ensemble des patchs de Bézier partageant le même sommet $\mathbf{p}_i$ se raccordent en ce point suivant une continuité de type C^1 ou plus exactement G^1

1.2.2.3 Nature du côté $E(\mathbf{p}_0, \mathbf{p}_1)$

Le côté $E(\mathbf{p}_0, \mathbf{p}_1)$ correspondant au bord du patch entre les points $\mathbf{p}_0$ et $\mathbf{p}_1$, correspond aux points $p(u, v)$ pour lesquels le paramètres v est égal à zéro :

$$p(u,v) \in E(\mathbf{p}_0, \mathbf{p}_1) \quad \Longleftrightarrow \quad \left\{ \begin{array}{lcl} u & \in & [0,1] \\ v & = & 0 \end{array} \right.$$

Le côté $E(\mathbf{p}_0, \mathbf{p}_1)$ sera donc décrit par l'équation $p(u)$ telle que :

$$p(u) = p(u,0) = w^3 \cdot \mathbf{p}_0 + u^3 \cdot \mathbf{p}_1 + 3 \cdot (\, uw^2 \cdot \mathbf{p}_{01} + u^2 w \cdot \mathbf{p}_{10})$$

En plus, les dérivées le long de ce côté sont :

$$\begin{array}{rcl} p^u(u,0) & = & - \; 3 \cdot w^2 \cdot (\mathbf{p}_0 - \mathbf{p}_{01}) \\ & & + \; 3 \cdot u^2 \cdot (\mathbf{p}_1 - \mathbf{p}_{10}) \\ & & - \; 6 \cdot uw \cdot (\mathbf{p}_{01} - \mathbf{p}_{10}) \end{array}$$

$$\begin{array}{rcl} p^v(u,0) & = & - \; 3 \cdot w^2 \cdot (\mathbf{p}_0 - \mathbf{p}_{02}) \\ & & + \; 3 \cdot u^2 \cdot (\mathbf{p}_{12} - \mathbf{p}_{10}) \\ & & - \; 6 \cdot uw \cdot (\mathbf{p}_{01} - \mathbf{q} \;\;) \end{array}$$

A partir de ces expressions on déduit que le côté $E(\mathbf{p}_0, \mathbf{p}_1)$ possède les propriétés suivantes :

- l'équation $p(u) = p(u, 0)$ dépend seulement de $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{N}_0$ et $\mathbf{N}_1$.
- la tangente $p^u(u, 0)$ au patch $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ est identique à la tangente $p^u(u)$ à la courbe $E(\mathbf{p}_0, \mathbf{p}_1)$. La tangente $p^u(u, 0)$ dépend seulement de $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{N}_0$ et $\mathbf{N}_1$,
- la tangente $p^v(u, 0)$ au patch $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ dépend non seulement de $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{N}_0$ et $\mathbf{N}_1$, mais aussi du point central $\mathbf{q}$.

La dernière propriété implique que la normale $N(u, 0)$ au bord $E(\mathbf{p}_0, \mathbf{p}_1)$ du patch $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ dépend non seulement de $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{N}_0$ et $\mathbf{N}_1$, mais aussi du point central $\mathbf{q}$. On en déduit que, d'un point de vue géométrique, deux patchs cubiques de Bézier $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ et $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2')$ partageant le même côté $E(\mathbf{p}_0, \mathbf{p}_1)$ ont un raccord possédant les propriétés suivantes :

- continuité C^0,
- continuité C^1 aux sommets $\mathbf{p}_0$ et $\mathbf{p}_1$,
- pas de continuité C^1 sur le côté $E(\mathbf{p}_0, \mathbf{p}_1)$.

On peut, toutefois, rétablir la continuité C^1 de deux manières différentes :

- soit en introduisant des patchs de degré 9 ([51]),
- soit en découpant chaque triangle en trois sous-triangles suivant la technique de Clough-Toucher ([18]).

Les modèles paramétriques du type "patch de Bézier" ont l'intérêt de proposer une équation $p(u, v)$ calculable à l'aide des opérations arithmétiques de base. Ils présentent toutefois des inconvénients majeurs pour la résolution de nos problèmes de calcul et tracé des courbes d'isovaleur :

- ils ne sont pas toujours parfaits en ce qui concerne les problèmes de continuité,
- ils conduisent à des résolutions d'équations non linéaires compliquées dès lors que l'on veut "extraire" des courbes d'isovaleur à partir de surfaces modélisées.

Il ne faut pas perdre de vue que les seules données de départ sont :

- la position dans l'espace $3D$ des sommets $(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ de chaque triangle $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$,
- les valeurs $(\varphi_0, \varphi_1, \varphi_2)$ de la propriété φ aux sommets de chaque triangle.

En conséquence, nous pensons qu'**il est dangereux de se laisser enfermer dans un modèle mathématique conduisant à des calculs complexes dont la seule justification est le modèle lui même et non pas les données de départ**. Pour cette raison, en nous inspirant de la notion de patch cubique de Bézier, nous proposons dans ce qui suit un modèle beaucoup plus simple pour lequel :

- nous ne cherchons plus à obtenir une équation $p(u, v)$ représentant la géométrie interne de chaque patch $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$,
- nous nous contentons de l'équation mathématique de la géométrie de la frontière de chaque patch $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$.

Réseau cubique triangulaire de Bézier

Nous appellerons **réseau cubique triangulaire de Bézier** ou plus simplement **réseau de Bézier**, l'ensemble des courbes correspondant aux côtés $E(\mathbf{p}_i, \mathbf{p}_j)$ des patchs triangulaires cubiques de Bézier.
On peut vérifier que l'équation paramétrique $p_{ij}(u)$ d'un côté $E(\mathbf{p}_i, \mathbf{p}_j)$ peut s'écrire sous la forme :

$$p_{ij} = (1-u)^3 \cdot \mathbf{p}_i \quad + \quad u^3 \cdot \mathbf{p}_j \quad + \quad 3 \cdot (u \cdot (1-u)^2 \cdot \mathbf{p}_{ij}) \quad + u^2 \cdot (1-u) \cdot \mathbf{p}_{ji}) \quad (1.2)$$

$$\text{avec :} \quad \begin{cases} \mathbf{p}_{ij} = \mathbf{p}_i + \frac{\|\mathbf{p}_j - \mathbf{p}_i\|}{3} \cdot T_{ij} \\ \mathbf{p}_{ji} = \mathbf{p}_j + \frac{\|\mathbf{p}_j - \mathbf{p}_i\|}{3} \cdot T_{ji} \end{cases}$$

Calcul des tangentes $p^u_{ij}(u)$ et $U_{ij}(u)$

Toujours en se référant à ce qui a été dit à propos des patchs cubiques de Bézier, on vérifie facilement que la tangente non unitaire $p^u_{ij}(u)$, à la courbe est telle que :

$$\begin{aligned} p^u_{ij}(u) \quad = \quad & (1-u)^2 \cdot \| \mathbf{p}_j - \mathbf{p}_i \| \cdot T_{ij} \quad - \quad u^2 \cdot \| \mathbf{p}_j - \mathbf{p}_i \| \cdot \quad T_{ji} \\ & -6 \cdot u \cdot (1-u) \cdot (\| \mathbf{p}_i - \mathbf{p}_j \| \quad + \quad \tfrac{\|\mathbf{p}_j - \mathbf{p}_i\|}{3} \cdot (T_{ij} - T_{ji})) \end{aligned}$$

Par définition, et comme le suggère la figure 1.10, nous noterons U_{ij} la tangente unitaire associée à $p^u_{ij}(u)$:

$$U_{ij}(u) = \frac{p^u_{ij}(u)}{\| p^u_{ij}(u) \|}$$

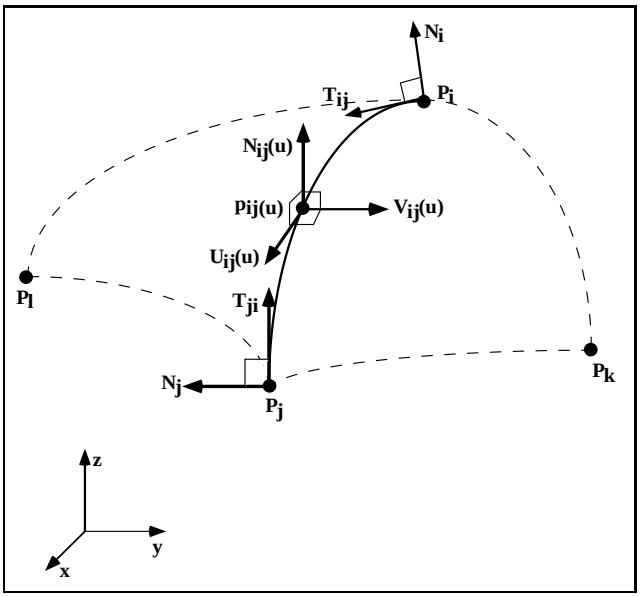

FIG. 1.10 – *Calcul de la tangente unitaire à un côté du patch de Bézier.*

Calcul de la normale $N_{ij}(u)$

Soit $n_{ij}(u)$ le vecteur correspondant à l'interpolation linéaire des normales N_i et N_j le long de l'axe $E(\mathbf{p}_i, \mathbf{p}_j)$:

$$n_{ij}(u) = (1-u) \cdot \mathbf{N}_i + u \cdot \mathbf{N}_j$$

Ce vecteur n'est en général ni unitaire ni orthogonal à $U_{ij}(u)$, c'est pourquoi nous proposons de le remplacer par un vecteur voisin $N_{ij}(u)$ défini par :

$$N_{ij}(u) = \frac{(U_{ij}(u) \times n_{ij}(u)) \times U_{ij}(u)}{\| (U_{ij}(u) \times n_{ij}(u)) \times U_{ij}(u) \|}$$

Dans ce qui suit, le vecteur $N_{ij}(u)$, ainsi défini, correspondra à la normale à la surface S à modéliser au point $p_{ij}(u)$. Comme le suggère la figure 1.10, le plan tangent à la surface S au point $p(u)$ est alors défini par les vecteurs unitaires $U_{ij}(u)$ et $V_{ij}(u)$ où $V_{ij}(u)$ est défini par :

$$V_{ij}(u) = N_{ij}(u) \times U_{ij}(u)$$

Remarque

A y regarder de plus près, nous pouvons constater que l'arc de courbe $p_{ij}(u)$ et le repère orthonormé associé $\{U_{ij}(u), V_{ij}, N_{ij}\}$ sont entièrement définis par les seules informations portées par les nœuds i et j, à savoir :

- les coordonnées des points $\mathbf{p}_i$ et $\mathbf{p}_j$,
- les composantes des vecteurs $\mathbf{N}_i$ et $\mathbf{N}_j$.

En prenant en compte cette remarque nous nous proposons d'étudier le calcul des courbes d'isovaleur sur des surfaces de géométrie et topologie générale. Dans un premier temps nous étudierons le cas où la propriété attachée aux courbes d'isovaleur correspond à l'altitude de la surface modélisée (on parle de courbes de niveau), ensuite nous étudierons le cas où la propriété attachée à la courbe d'isovaleur correspond à une propriété physique définie aux nœuds de la surface à modéliser.

1.3 Extraction des courbes de niveau

Dans ce paragraphe nous allons concentrer notre attention sur les courbes *iso_z* encore appelées "courbes de niveau" et correspondant à la fonction $\varphi(u, v)$ définie sur la surface S d'équation $p(u, v)$ telle que :

$$p(u,v) = \begin{bmatrix} p_x(u,v) \\ p_y(u,v) \\ p_z(u,v) \end{bmatrix} \Longrightarrow \varphi(u,v) = p_z(u,v)$$

Si l'on se fixe une valeur $\phi_0 \equiv z_0$, alors la courbe de niveau $C_0(\phi_0) \equiv C(z_0)$ correspond à l'intersection de S par un plan horizontal $H(z_0)$ d'altitude z_0. Cette dernière remarque sera particulièrement utile pour la suite.

1.3.1 Détection des triangles intersectés

La première opération à faire avant de pouvoir envisager de tracer la courbe $C(z_0)$ consiste à détecter tous les triangles traversés par cette courbe. Cette opération est particulièrement simple et rapide si, comme le suggère la figure 1.11, nous admettons les hypothèses simplificatrices suivantes :

- une courbe de niveau ne peut pas recouper deux fois le même côté $E(\mathbf{p}_i, \mathbf{p}_j)$ d'un triangle (figure 1.11a),
- une courbe de niveau fermée ne peut pas être entièrement contenue dans un triangle sans recouper ses côtés (figure 1.11b),
- une courbe de niveau ne peut pas passer strictement par un sommet de triangle (figure 1.11c).

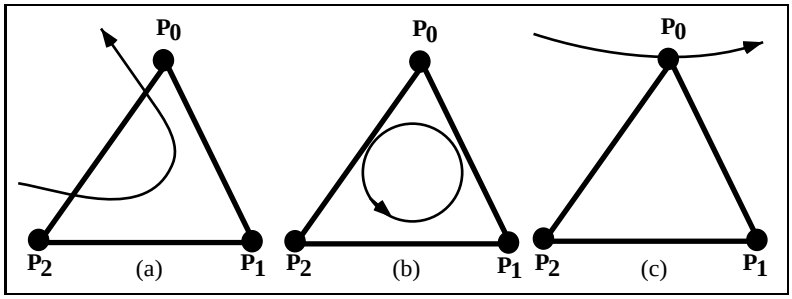

FIG. 1.11 – *Les configurations interdites par la nouvelle approche*

Si ces hypothèses sont respectées, alors un triangle $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ est considéré comme étant traversé par la courbe $C(z_0)$ si :

$$\exists\ (\mathbf{p}_i, \mathbf{p}_j) \in \{\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2\}\ : z_0 \in]\mathbf{p}_{iz}, \mathbf{p}_{jz}[$$

1.3.2 Intersection de $E(\mathbf{p}_i, \mathbf{p}_j)$ avec $C(z_0)$

Comme le suggère la figure 1.12, et compte tenu des hypothèses simplificatrices énoncées dans le paragraphe précédent, nous pouvons dire que si la courbe $C(z_0)$ intersecte un triangle $T(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k)$, alors elle ne peut qu'entrer par un côté, par exemple $E(\mathbf{p}_i, \mathbf{p}_j)$, et sortir par un autre côté, par exemple $E(\mathbf{p}_j, \mathbf{p}_k)$.

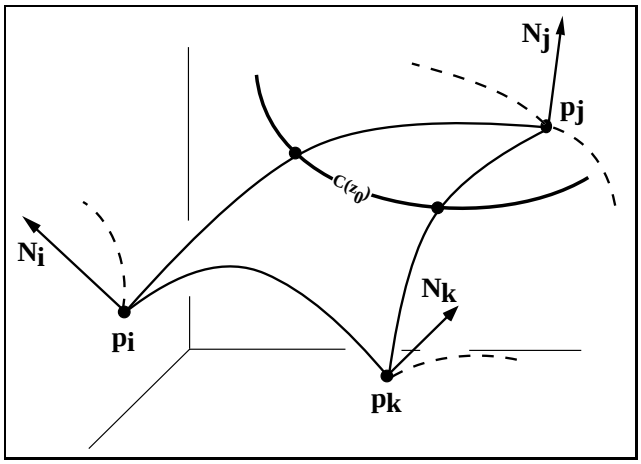

FIG. 1.12 – *Une courbe de niveau traversant le triangle* $T(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k)$

1.3.2.1 Calcul de l'abscisse û du point d'intersection

Soit $E(\mathbf{p}_i, \mathbf{p}_j)$ un côté du triangle $T(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k)$ traversé par la courbe $C(z_0)$. Compte tenu des hypothèses simplificatrices, cela implique que l'on soit dans l'une des situations suivantes :

$$\begin{cases} (a)\ p_{iz} < z_0 < p_{jz} \\ \text{ ou bien} \\ (b)\ p_{iz} > z_0 > p_{jz} \end{cases}$$

En se souvenant que le côté $E(\mathbf{p}_i, \mathbf{p}_j)$ est représenté par un polynôme $p_{ij}(u)$ de degré 3, on en déduit que dans le plan (u, z) le graphe de la composante $p_{ij}(u)$ de ce côté correspond à l'un des deux cas représentés par la figure 1.13. Comme on peut le voir, suivant l'orientation des tangentes en $u = 0$ et $u = 1$, l'équation $p_{ijz}(u) = z_0$ peut avoir une ou trois racines. Il faut toutefois noter que, dans la pratique, il n'y a généralement qu'une seule racine û qui vérifie l'hypothèse simplificatrice spécifiant que $C(z_0)$ ne peut pas couper plusieurs fois un même côté $E(p_i, p_j)$. Dans le cas où il y a 3 racines, nous choisirons par exemple celle qui est la plus proche du milieu du côté $E(\mathbf{p}_i, \mathbf{p}_j)$. En pratique, pour obtenir une telle solution unique û dans tous les cas de figure, il suffit d'utiliser une méthode de bipartition :

$u_0 = 0.0$
$u_1 = 1.0$

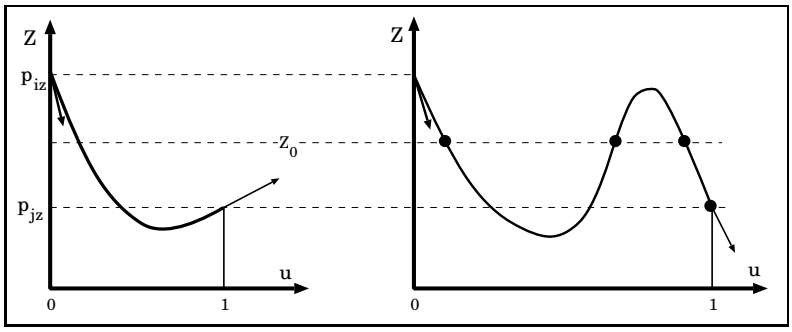

FIG. 1.13 – *Résolution de l'équation* $p_{ij\,z}(u) = z_0$

```
while(nombre maximum d'itérations non atteint)
  {
      û = (u0+u1)/2
      if(p_ij z(û) < z0) u1 = û ;
      else u0 = û;
  }
```

1.3.3 Calcul de la tangente $t_{ij}(\hat{u})$

Comme l'indique la figure 1.15, soit $t_{ij}(\hat{u})$ la tangente unitaire à la courbe $C(z_0)$ en son point d'intersection $p_{ij}(\hat{u})$ avec le côté $E(\mathbf{p}_i, \mathbf{p}_j)$. On peut remarquer, sur la figure 1.14, que cette tangente est située dans deux plans différents :

- puisque la courbe $C(z_0)$ est tracée sur la surface S, on en déduit que la tangente $t_{ij}(\hat{u})$ doit appartenir au plan tangent à la surface S, défini par la normale $N_{ij}(\hat{u})$,
- puisque $C(z_0)$ correspond à l'intersection de la surface S avec le plan horizontal $H(z_0)$, on en déduit que $t_{ij}(\hat{u})$ doit appartenir à ce plan.

Il est donc facile de construire le vecteur unitaire $t_{ij}(\hat{u})$ qui est nécessairement tel que :

$$t_{ij}(\hat{u}) = \frac{H \times N_{ij}(\hat{u})}{\| H \times N_{ij}(\hat{u}) \|}$$

où H n'est autre que le vecteur vertical :

$$H = [0\ 0\ 1]^t$$

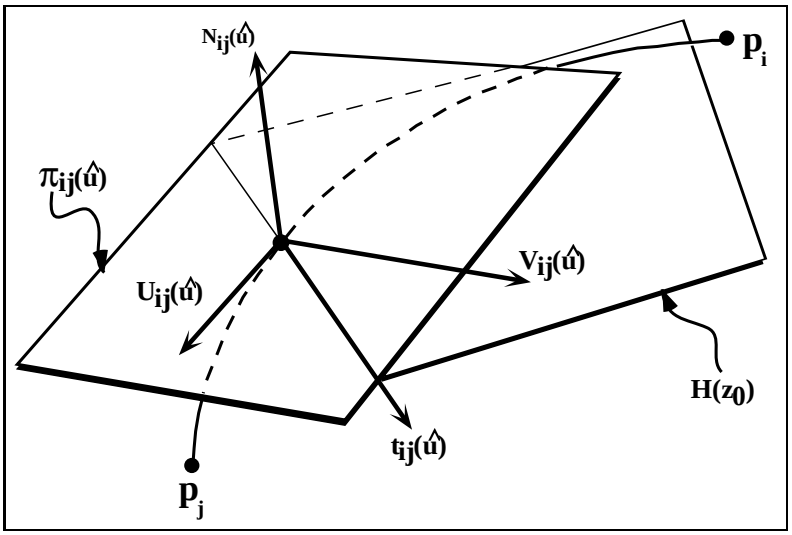

FIG. 1.14 – *Position de la tangente à la courbe de niveau*

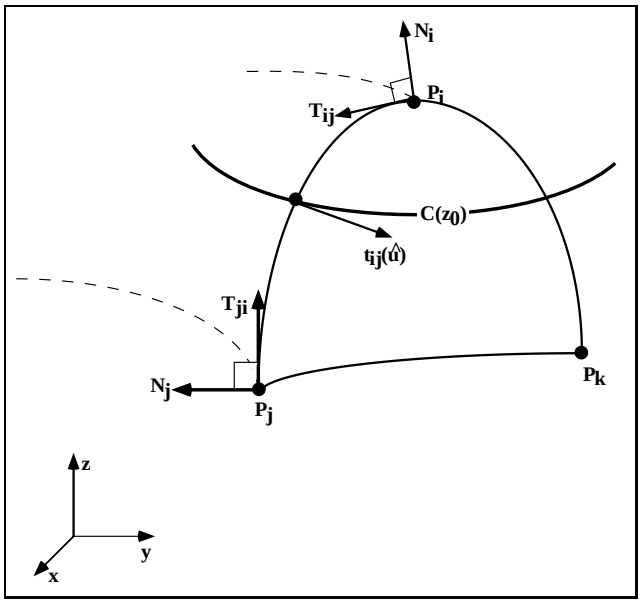

FIG. 1.15 – *Calcul de la tangente à la courbe de niveau*

1.3.4 Calcul de l'intersection de $C(z_0)$ avec $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$

Considérons un triangle $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ de la surface S intersecté par la courbe $C(z_0)$. Comme le montre la figure 1.16, supposons que $C(z_0)$ entre par le point $M_1 = p_{01}(\hat{u}_1)$ sur le côté $E(\mathbf{p}_0, \mathbf{p}_1)$ et sort par le point $M_2 = p_{02}(\hat{u}_2)$ sur le côté $E(\mathbf{p}_0, \mathbf{p}_2)$. Soient d'autre part $t_{01}(\hat{u}_1)$ et $t_{02}(\hat{u}_2)$ les directions définissant la droite tangente en M_1 et M_2 à $C(z_0)$ et soient $t^*_{01}(\hat{u}_1)$ et $t^*_{02}(\hat{u}_2)$ les vecteurs tels que :

$$t^*_{01}(\hat{u}_1) = \begin{cases} t_{01}(\hat{u}_1) \text{ si angle } (t_{01}(\hat{u}_1), M_1M_2) < \text{ angle } (-t_{01}(\hat{u}_1), M_1M_2) \\ -t_{01} \text{ sinon} \end{cases}$$

$$t^*_{02}(\hat{u}_2) = \begin{cases} t_{02}(\hat{u}_2) \text{ si angle } (t_{02}(\hat{u}_2), M_2M_3) < \text{ angle } (-t_{02}(\hat{u}_2), M_2M_3) \\ -t_{02} \text{ sinon} \end{cases}$$

Soient d'autre part deux vecteurs τ_1 et τ_2 tels que :

$$\tau_1 = \frac{\|M_1M_2\|}{3} \cdot \frac{t^*_{01}(\hat{u}_1)}{\|t^*01(\hat{u}_1)\|}$$

$$\tau_2 = \frac{\|M_2M_3\|}{3} \cdot \frac{t^*_{02}(\hat{u}_2)}{\|t^*02(\hat{u}_2)\|}$$

Comme le suggère la figure 1.16, nous proposons de représenter l'intersection de $C(z_0)$ avec $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ sous la forme d'un arc cubique de Bézier $q(t)$ interpolant les points M_1 et M_2 ainsi que les tangentes τ_1 et τ_2 associées :

$$\begin{array}{rcl} q(t) & = & (1-u)^3 \cdot M_1 + u^3 \cdot M_2 + 3 \cdot \{u \cdot (1-u)^2 \cdot (M_1 + \tau_1) \\ & + & \mathrm{u}^2 \cdot (1-u) \cdot (M_2 + \tau_2)\} \end{array} \tag{1.3}$$

Cette courbe a les propriétés suivantes :

$$\begin{cases} q(0) = M_1 \\ \\ q(1) = M_2 \end{cases} \quad \begin{cases} \frac{dq(0)}{dt} = \tau_1 \\ \\ \frac{dq(1)}{dt} = \tau_2 \end{cases}$$

De plus, lorsque M_1 et M_2 tendent vers $\mathbf{p}_1$ et $\mathbf{p}_2$ (cf figure 1.16), alors l'arc de courbe (M_1, M_2) représenté par $q(t)$ tend vers le côté $E(\mathbf{p}_1, \mathbf{p}_2)$.

Remarque

Il n'est pas inutile de remarquer que l'équation $q(t)$ de l'arc de courbe (M_1, M_2), définie précédemment, ne dépend que de z_0 et des informations portées par les trois sommets du triangle $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$, à savoir :

– la position des trois sommets $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ dans l'espace $3D$,

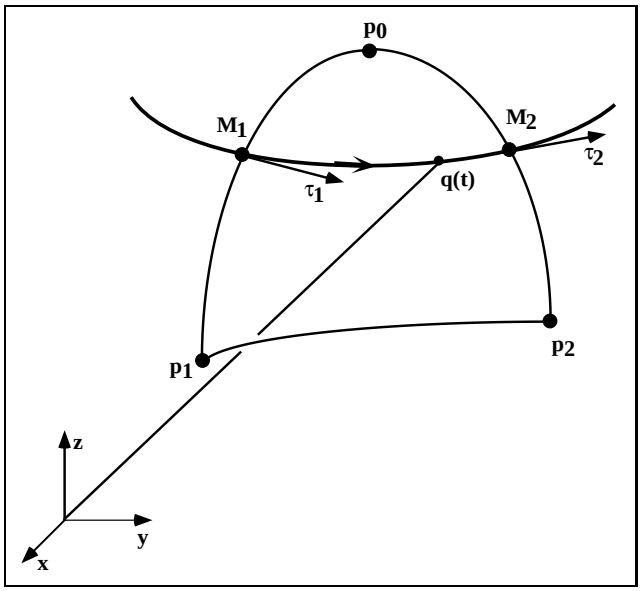

FIG. 1.16 – *Calcul de l'arc de courbe joignant le point d'entrée et le point de sortie*

– les composantes des trois normales $\mathbf{N}_0, \mathbf{N}_1, \mathbf{N}_2$

Cette remarque simplifie la programmation car elle rend possible la détermination de chaque arc (M_1, M_2) indépendamment de ce qui se passe sur le triangle voisin.

Les résultats obtenus par la nouvelle approche, proposée dans les paragraphes précédents, seront exposés à la fin de ce chapitre.

Après avoir présenté l'algorithme de construction d'une courbe de niveau sur une surface triangulée nous nous proposons d'étudier, dans le paragraphe suivant, le calcul d'une courbe iso_ϕ pour une propriété $\varphi(u, v)$ définie sur une surface triangulée.

1.4 Calcul des courbes iso_ϕ

Le problème du calcul d'une courbe iso_ϕ pour une propriété $\varphi(u, v)$, définie sur S, est très similaire à celui des courbes de niveau iso_z. Cependant,

deux différences existent :

- la détermination de l'abscisse $\hat{u}$ du point d'intersection avec un côté $E(\mathbf{p}_i, \mathbf{p}_j)$,
- la détermination de la tangente $t_{ij}(\hat{u})$ au point d'intersection $p_{ij}(\hat{u})$.

Comme nous l'avons fait pour la géométrie, nous supposons ici aussi que l'information concernant $\varphi(u,v)$ est conservée au niveau des sommets des triangles :

$$p(u,v) = \mathbf{p}_i \Longrightarrow \varphi(u,v) = \varphi_i$$

Le problème posé consiste à trouver l'ensemble $C_\varphi(\phi_0)$ des points $p(u,v)$ de la surface S pour lesquels nous avons :

$$p(u,v) \in C_\varphi(\phi) \Longleftrightarrow \varphi(u,v) = \phi$$

Si l'on reprend les hypothèses simplificatrices utilisées pour déterminer $C(z_0)$, alors $C_\varphi(\phi)$ est une courbe posée sur S appelée courbe *iso_ϕ* ou *courbe iso-valeur*.

1.4.1 Calcul du gradient

Soit $\mathbf{N}_i$ la normale unitaire en un sommet $\mathbf{p}_i$ d'un triangle de la surface S. Comme l'illustre la figure 1.17, il est toujours possible de trouver deux vecteurs unitaires U et V tels que (U, V, N_i) soit une base orthonormée. Les vecteurs U et V définissent le plan tangent π_i à la surface au nœud $\mathbf{p}_i$, et il est possible de projeter (suivant la direction N_i) tous les points $\mathbf{p}_k$ voisins de $\mathbf{p}_i$ sur ce plan.
Pour tout voisin $\mathbf{p}_k$ de $\mathbf{p}_i$, posons :

$$\left|\begin{array}{lcl} \mathbf{p}_k^* & = & \text{projection de } \mathbf{p}_k \text{ sur } \pi_i \\ (u_k, v_k) & = & \text{coordonnées de } \mathbf{p}_k^* \text{ dans le repère } (U,V) \text{ de } \pi_i \end{array}\right.$$

En supposant que les distances $\| \mathbf{p}_k^* - \mathbf{p}_k \|$ sont petites, nous pouvons poser :

$$\varphi_k^* \equiv \varphi_k = \text{ valeur de } \varphi \text{ au point } \varphi_k^*$$

ce qui revient à écrire :

$$\varphi^*(u_k, v_k) \simeq \varphi(u_k, v_k) = \varphi_k$$

Soit $\Lambda(i)$ l'ensemble des indices correspondant aux nœuds $\mathbf{p}_i$ directement

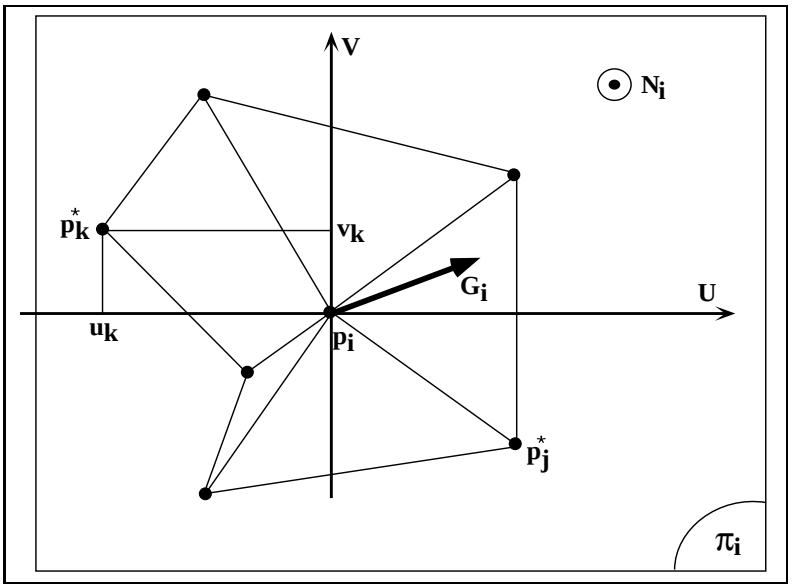

FIG. 1.17 – *Calcul du gradient de propriété*

connectés à $\mathbf{p}_i$:

$$j \in \Lambda(i) \Longleftrightarrow E(\mathbf{p}_i, \mathbf{p}_j) \text{ existe}$$

L'approximation linéaire de $\varphi^*(u,v)$ au voisinage du point $\mathbf{p}_i$ peut s'écrire sous la forme suivante, où G_i est le vecteur gradient de φ au point $\mathbf{p}_i$ dans le plan π_i :

$$\left|\begin{array}{ll} \varphi^*(u,v) & = \overline{\varphi_i} + \gamma_i^u \cdot u + \gamma_i^v \cdot v \\ \text{avec :} & \left\{\begin{array}{lll} \overline{\varphi}_i & = & \frac{\varphi_i + \sum_{k\in\Lambda(i)} \varphi_k}{1+|\Lambda(i)|} \\ G_i & = & \gamma_i^u \cdot U + \gamma_i^v \cdot V \\ |\Lambda(i)| \text{ est le nombre d'élements de l'ensemble } \Lambda(i) & & \end{array}\right. \end{array}\right.$$

Pour tout $k \in \Lambda(i)$, il existe un couple (u_k, v_k) et cela nous permet d'écrire le système d'équations linéaires suivant, aux inconnues (γ_i^u, γ_i^v) :

$$\left|\begin{array}{l} [W_k]^t \cdot [\Gamma_i] = (\varphi_k - \overline{\varphi_i}) \quad \forall k \in \Lambda(i) \\ \text{avec} \left\{\begin{array}{lll} [W_k] & = & \left[\begin{array}{c} u_k \\ v_k \end{array}\right] \\ [\Gamma] & = & \left[\begin{array}{c} \gamma_i^u \\ \gamma_i^v \end{array}\right] \end{array}\right. \end{array}\right.$$

On montre facilement que la solution de ce système au sens des moindres carrés est telle que :

$$[\Gamma] = (\sum_{k\in\Lambda(i)} [W_k]\cdot[W_k]^t)^{-1}\cdot(\sum_{k\in\wedge(i)} (\varphi_k - \overline{\varphi_i})\cdot[W_k])$$

Nous proposons d'utiliser les composantes (γ_i^u, γ_i^v) de cette solution pour estimer le gradient G_i au point $\mathbf{p}_i$:

$$G_i = \gamma_i^u \cdot U + \gamma_i^v \cdot V$$

Il est alors possible de calculer le gradient g_{ij} de φ au point $\mathbf{p}_i$ dans la direction $D_{ij} = (\mathbf{p}_j^* - \mathbf{p}_i)$ qui est défini par :

$$g_{ij} = G_i \cdot \frac{D_{ij}}{\| D_{ij} \|} = G_i \cdot T_{ij}$$

En d'autres termes, si nous désignons par s l'abscisse curviligne le long de l'axe $E(\mathbf{p}_0, \mathbf{p}_1)$, alors nous pouvons considérer qu'au voisinage de $\mathbf{p}_i$, on a :

$$\left.\frac{d\varphi}{ds}\right|_{s\simeq 0} = g_{ij}$$

Par ailleurs, si l'on dérive l'équation 1.2 de l'arc $E(\mathbf{p}_i, \mathbf{p}_j)$ relativement à l'abscisse curviligne, on obtient :

$$\frac{dp_{ij}(u)}{ds} = \frac{dp_{ij}(u)}{du}\cdot\frac{du}{ds} = p_{ij}^u \cdot \frac{du}{ds}$$

Compte tenu que :

$$\| \frac{dp_{ij}}{ds} \| = 1$$

On en déduit que :

$$\frac{ds}{du} = \| p_{ij}^u(u) \|$$

Compte tenu que sur l'arc $E(p_i, p_j)$ nous pouvons toujours écrire :

$$\frac{d\varphi}{du} = \frac{d\varphi}{ds}\cdot\frac{ds}{du}$$

On en conclut que :

$$\left.\frac{d\varphi}{du}\right|_{u=0} = g_{ij}\cdot \| p_{ij}^u(0) \|$$

1.4.1.1 Intersection de $E(\mathbf{p}_i, \mathbf{p}_j)$ avec $C_\varphi(\phi)$

Sur l'axe $E(\mathbf{p}_i, \mathbf{p}_j)$, la fonction $\varphi(u)$ est connue :

- par des valeurs (φ_i, φ_j) données aux deux extrémités,
- par ses dérivés $\left.\frac{d\varphi}{du}\right|_{u=0}$ et $\left.\frac{d\varphi}{du}\right|_{u=1}$ estimées à l'aide du procédé décrit dans paragraphe précédent.

Il est donc possible de calculer un polynôme du $3^{\text{ème}}$ degré $\varphi(u)$ respectant les valeurs et les dérivées de φ aux extrémités de $E(\mathbf{p}_i, \mathbf{p}_j)$. L'intersection de $C_\varphi(\phi)$ avec le coté $E(\mathbf{p}_i, \mathbf{p}_j)$ correspond alors au point $p_{ij}(\hat{u})$ dont l'abscisse $\hat{u}$ est solution de :

$$\varphi(\hat{u}) = \phi$$

De façon analogue à la technique employée dans le cas des courbes de niveau *iso_z*, la recherche de $\hat{u}$ est effectuée par bipartition entre u=0 et u=1.

1.4.1.2 Détermination de la tangente $t_{ij}(\hat{\mathbf{u}})$

Dès lors que l'on connaît les vecteurs G_i et G_j aux sommets respectifs $\mathbf{p}_i$ et $\mathbf{p}_j$ de l'arc $E(\mathbf{p}_i, \mathbf{p}_j)$, il est possible d'estimer une direction de tangente $t_{ij}(\hat{u})$ à la courbe $C_\varphi(\phi)$. Pour ce faire, nous proposons de procéder comme suit :

1. nous calculons $G_{ij}(\hat{u})$ interpolant linéairement G_i et G_j :

$$G_{ij}(\hat{u}) = (1 - \hat{u}) \cdot G_i + \hat{u} \cdot G_j$$

2. Nous posons :

$$t_{ij}(\hat{u}) = \frac{N_{ij}(\hat{u}) \times G_{ij}(\hat{u})}{\| N_{ij}(\hat{u}) \times G_{ij}(\hat{u}) \|}$$

Cela suppose que $\| N_{ij}(\hat{u}) \times G_{ij}(\hat{u}) \| \neq 0$, sinon :

$$t_{ij}(\hat{u}) = \frac{N_{ij}(\hat{u}) \times (\mathbf{p}_j - \mathbf{p}_i)}{\| N_{ij}(\hat{u}) \times (\mathbf{p}_j - \mathbf{p}_i) \|}$$

Le procédé de calcul de l'intersection de $C(\phi_0)$ avec $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$, sera identique à celui adopté pour la courbe $C(z_0)$.

1.5 Conclusion

L'approche que nous avons proposée, permet de calculer des courbes d'isovaleur sur une surface en ne considérant que les connexions entre les différents nœuds de la surface. Cette approche est assez générale, elle peut être utilisée dans différents domaines scientifiques pour rendre compte des variations d'un paramètre d'une surface. La rapidité de l'extraction des courbes d'isovaleur permet au géologue une utilisation très fréquente des courbes d'isovaleur. En effet, les courbes de niveau font partie des outils de contrôle de la géométrie des surfaces les plus utilisés au quotidien. Ces courbes de niveau seront utilisées lors de la modélisation de surfaces géologiques (voir chapitres 4 et 5). Les figures suivantes permettent d'apprécier les résultats obtenus par notre approche sur des surfaces de géométrie générale et pouvant comporter des trous.

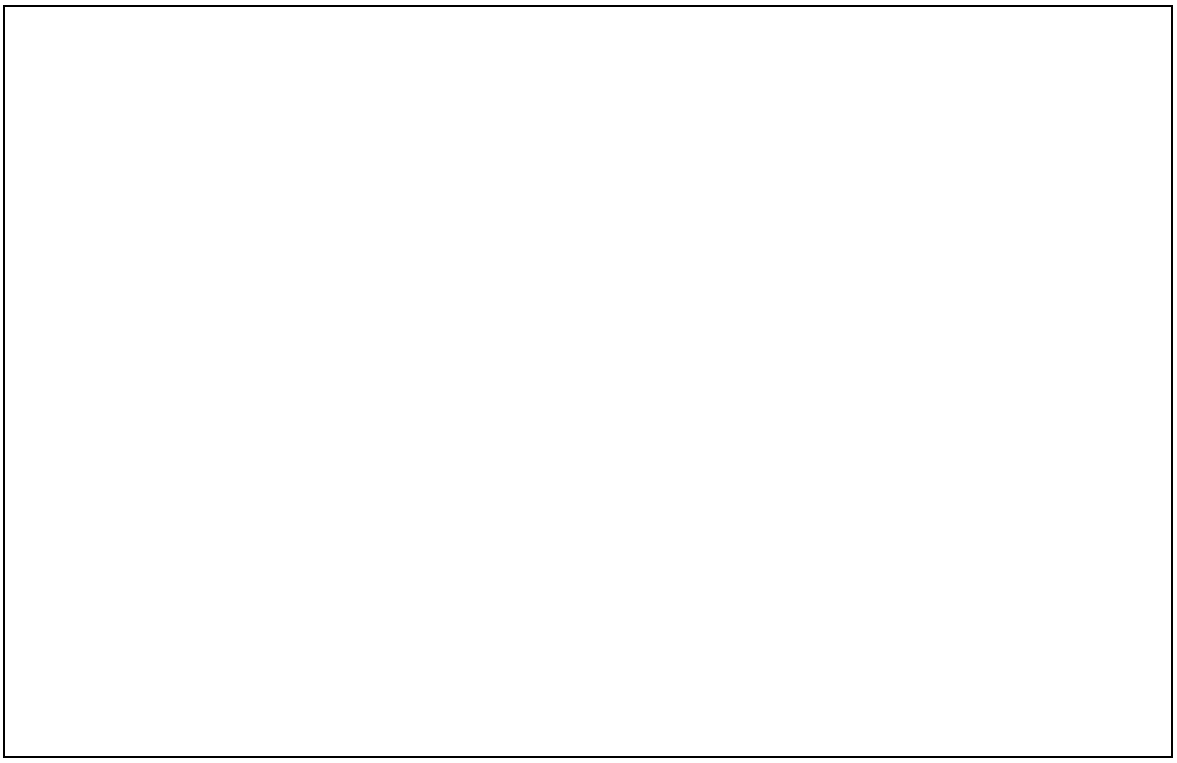

FIG. 1.18 – *Calcul des courbes de niveau sur un dôme de sel*

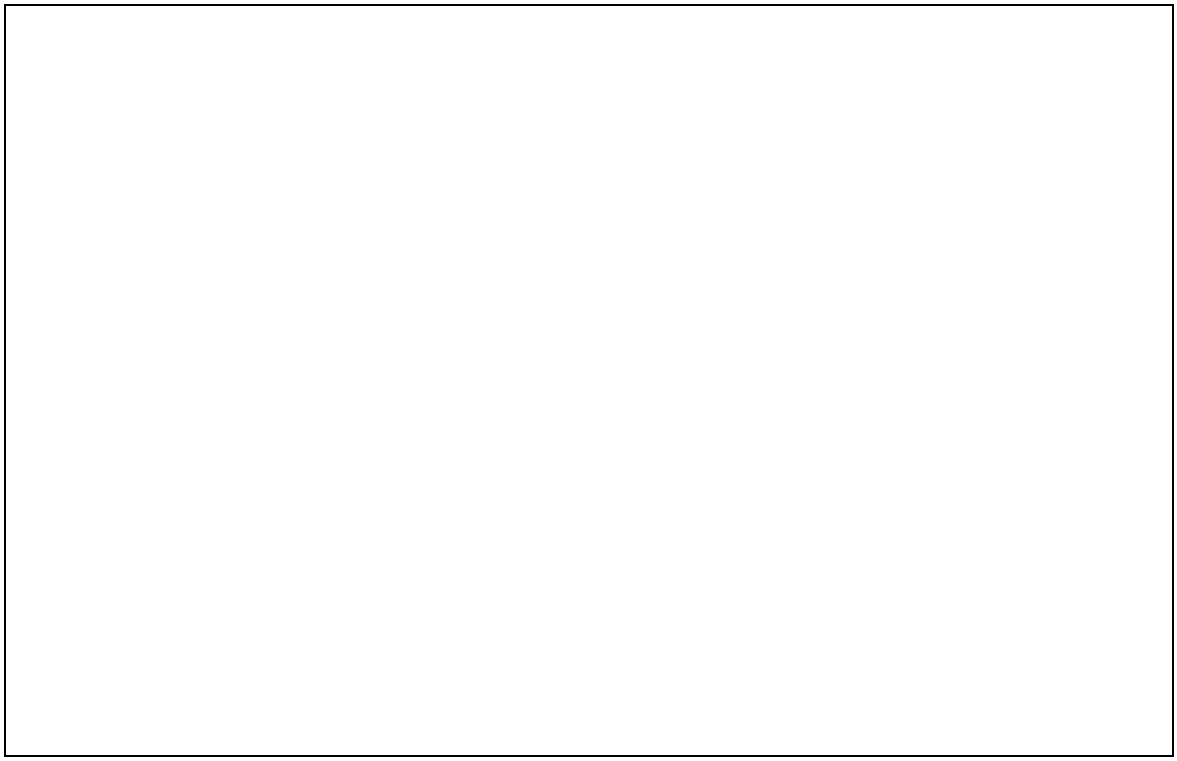

FIG. 1.19 – *Calcul des courbes de niveau sur une surface présentant plusieurs trous*

Chapitre 2

La méthode DSI

Le projet G◯CAD propose une méthode originale de modélisation, spécialement adaptée aux problèmes de l'industrie pétrolière et minière. Le cœur du système consiste en une nouvelle méthode d'interpolation appelée **Discrete Smooth Interpolation** ou plus simplement (DSI)([42]). Cette méthode d'interpolation suppose que les objets sont modélisés sous forme d'un ensemble de points connectés les uns aux autres de façon à constituer un graphe :

- la position des nœuds définit la géométrie de l'objet,
- la *connectivité* des nœuds définit la topologie.

Dans le système G◯CAD la structure de données définissant la notion de nœud du graphe est appelée **atome** (cf figure 2.1). L'atome est l'entité élémentaire du maillage d'un objet quelconque. La structure correspondant à un atome contient toutes les informations nécessaires à l'interpolateur DSI, à savoir, la position géométrique du nœud, sa connectivité avec les autres atomes (appelés *satellites*), les propriétés physiques et la liste des contraintes que l'interpolateur doit respecter localement (voir Annexe).

Bien que la méthode $D.S.I$ puisse être utilisée pour modéliser des objets quelconques (courbes, valeurs, $\cdots$), dans ce qui suit nous nous limiterons

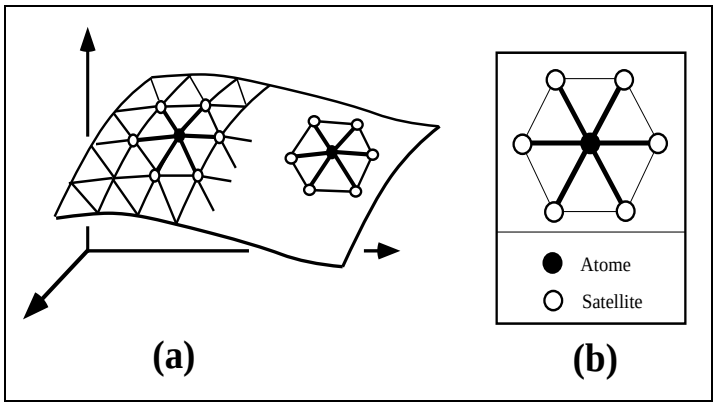

FIG. 2.1 – *Réseau atomique d'une surface*

au cas des surfaces composées de facettes triangulaires (cf figure 2.1) pour lesquelles :

– les sommets des triangles correspondent aux nœuds du graphe,

– les côtés des triangles correspondent aux axes du graphe.

2.1 Principe de la méthode DSI

Supposons que les sommets des triangles de la surface représentée sur la figure 2.1, soient numérotés de 1 à N, et soit Ω l'ensemble de ces sommets appelés "atomes" :

$$\Omega = \{1, 2, \cdots, N\}$$

Pour chaque nœud k on appelle voisinage de k et l'on note $N(k)$, l'ensemble des nœuds directement connectés à k.

Le vecteur joignant l'origine de l'espace $3D$ à un sommet donné k est noté $\varphi(k)$ et l'on note φ la collection de ces vecteurs pour tous les sommets de S.

$$\varphi = \{\varphi(1), \varphi(2), \cdots, \varphi(N)\}$$

Le critère de rugosité

La méthode DSI est basée sur un critère de rugosité locale $R(\varphi \mid k)$ définie en chaque nœud $k \in \Omega$ par :

$$R(\varphi \mid k) = \| \sum_{\alpha \in N(k)} \nu^{\alpha}(k) \cdot \varphi(\alpha) \|^2$$

Les coefficients $\{\nu^{\alpha}(k)\}$ sont des pondérateurs donnés ([42].
La somme des critères locaux $R(\varphi \mid k)$, représente une rugosité globale notée $R(\varphi)$:

$$R(\varphi) = \sum_{k \in \Omega} \mu(k) \cdot R(\varphi \mid k)$$

$\mu(k)$ est une fonction de pondération strictement positive donnée ([15]).

2.1.1 Les critères de contraintes

DSI suppose que chacune des contraintes à respecter peut être exprimée de la manière suivante :

$$A_i \cdot \varphi \equiv \sum_{\nu=(x,y,z)} \sum_{\alpha \in \Omega} A_i^{\nu}(\alpha) \cdot \varphi^{\nu}(\alpha) \simeq b_i$$

où :

- (x, y, z) sont les coordonnées $(\varphi^x(\alpha), \varphi^y(\alpha), \varphi^z(\alpha))$ du nœud α,
- $A_i^{\nu}(\alpha)$ et b_i sont des coefficients donnés définissant la $i^{\text{ème}}$ contrainte à respecter.

On introduit alors un critère de rugosité modifié en présence de contraintes :

$$R^*(\varphi) = \sum_{k \in \Omega} R(\varphi \mid k) + \sum_i \varpi_i^2 \cdot \mid A_i(\varphi) - b_i \mid^2$$

Les coefficients ϖ_i^2, appelés "coefficients de précision", modulent l'importance de chaque contrainte.
La méthode DSI prend en compte deux types de contraintes :

1. **les contraintes floues** (ou fuzzy constraints)
   Il s'agit de prendre en compte une information floue concernant une valeur inconnue $\{\varphi(\lambda) : \lambda \in \Omega\}$. A cette information floue est associé

un coefficient de précision positif ϖ_i^2. Si on désire, par exemple, que la position $\varphi(\lambda)$ du nœud λ soit proche d'une valeur donnée $\widetilde{\varphi_\lambda}$, avec un coefficient de précision ϖ_λ^2, la condition suivante doit être vérifiée :

$$\varphi(\lambda) \simeq \widetilde{\varphi}(\lambda) \tag{2.1}$$

Cela peut se se traduire par :

$$\varpi_\lambda^2 \cdot |\varphi(\lambda) - \widetilde{\varphi}(\lambda)|^2 \text{ est minimum} \tag{2.2}$$

Si on suppose que la relation 2.1 définit la $i^{\text{ème}}$ contrainte au niveau du nœud λ, cette même relation peut alors se traduire sous forme d'une contrainte DSI en posant :

$$\left| \begin{array}{l} A_i(\alpha) = \left\{ \begin{array}{ll} +1 & \text{si } \alpha = \lambda \\ 0 & \text{sinon} \end{array} \right. \\ b_i = \widetilde{\varphi}\ (\lambda) \\ \varpi_i^2 = \varpi_\lambda^2 \end{array} \right.$$

On définit deux coefficients $\Gamma_i(\alpha)$ et $\gamma_i(\alpha)$ tels que :

$$\Gamma_i(\alpha) = \left\{ \begin{array}{ll} -\varpi_\lambda^2\ \widetilde{\varphi}\ (\lambda) & \text{si } \alpha = \lambda \\ 0 & \text{sinon} \end{array} \right.$$

$$\gamma_i(\alpha) = \left\{ \begin{array}{ll} \varpi_\lambda^2 & \text{si } \alpha = \lambda \\ 0 & \text{sinon} \end{array} \right.$$

Grâce aux coefficients $\Gamma_i(\varphi)$ et $\gamma_i(\varphi)$, la $i^{\text{ème}}$ contrainte sera prise en compte au niveau de la forme locale de l'équation de $D.S.I.$

- **Remarque :**
  La dénomination *contraintes floues* vient du fait que ces contraintes agissent sur les nœuds de la surface, en autorisant un certain degré de liberté inversement proportionnel au coefficient de précision.

2. **les contraintes dures** (ou hard constraints)
   A l'inverse des contraintes floues, les contraintes dures désignent l'ensemble des informations précises concernant les nœuds d'une surface.

Reprenons l'exemple précédent, la contrainte consiste à modifier les coordonnées du nœud λ pour vérifier la relation suivante

$$\varphi(\lambda) = \widetilde{\varphi}(\lambda)$$

Les contraintes dures sont prises en compte, par DSI, après que les critères de minimisation de la rugosité et du degré de violation des contraintes floues aient été pris en compte.

Le résultat obtenu par la méthode DSI est un compromis entre la minimisation de la rugosité et la minimisation du degré de violation des contraintes floues. Cela revient à vérifier la relation suivante :

$$\frac{\partial R^*(\varphi)}{\partial \varphi(\alpha)} = 0, \forall \alpha \in \Omega \tag{2.3}$$

Une solution partielle au nœud α est recherchée pour valider la relation 2.3. Ce procédé sera répété, itérativement, pour tout nœud $\alpha \in \Omega$ jusqu'à ce que la solution soit proche du point où $\dfrac{\partial R^*(\varphi)}{\partial \varphi(\alpha)} = 0$.
En un nœud α, la relation 2.3 est vérifiée lorsque

$$\varphi(\alpha) = -\frac{1}{M(\alpha)} \cdot \left\{ \sum_{k \in N(\alpha)} \{\mu(k) \cdot \nu^{\alpha}(k) \cdot \sum_{\substack{\beta \in N(k) \\ \beta \neq \alpha}} \nu^{\beta}(k) \cdot \varphi(\beta)\} + \sum_i \Gamma_i(\alpha) \right\}$$

avec

$$\begin{cases} M(\alpha) = \displaystyle\sum_{k \in N(\alpha)} \mu(k) \cdot (\nu^{\alpha}(k))^2 + \sum_i \gamma_i(\alpha) \\ \gamma_i(\alpha) = \varpi_i^2 \cdot (A_i(\alpha))^2 \\ \Gamma_i(\alpha) = \varpi_i^2 \cdot A_i(\alpha) \cdot \{\displaystyle\sum_{\beta \neq \alpha} A_i(\beta)\varphi(\beta) - b_i\} \end{cases}$$

Cette équation est connue sous le nom de *forme locale de DSI*. On montre que l'application itérative de cette forme locale à chaque nœud, diminue la rugosité globale $R^*(\varphi)$, et que la méthode converge vers une solution unique indépendante de la solution initiale ([41]). Ce procédé itératif est résumé en pseudo-c :

```
 I est l'ensemble des nœuds où la fonction φ(α) est connue
φ est la solution initiale approximative
tant que( nécessité de plus d'itérations )
{
  pour( α ∈ I )
  {
      calcul de φ(α) qui minimise le critère de rugosité
      J est la liste des contraintes floues attachées au nœud α
      pour(τ ∈ J)
      {
        calcul de φ(α) qui minimise le degré de violation
        de la contrainte τ
      }
      K est la liste des contraintes dures attachées au nœud α
      tant que(τ ∈ K)
      {
        calcul de φ(α) qui respecte la contrainte τ
      }
  }
}
```

Comme nous l'avons signalé, à plusieurs reprises, DSI tend à minimiser la rugosité d'une surface en respectant un ensemble de contraintes linéaires attachées aux atomes de la même surface. Avec l'augmentation du nombre de contraintes géométriques permettant la modélisation d'objets complexes, il a été nécessaire d'introduire un mécanisme général pour l'implémentation et la manipulation des contraintes. Il s'agit du modèle Atomique-Contrôleur. Un bref aperçu de ce modèle sera présenté dans le paragraphe suivant. Le lecteur peut se référer à l'article [15], pour avoir plus de détails quant au fonctionnement de ce modèle.

2.2 Modèle Atomique-Contrôleur

Les contraintes de DSI correspondent généralement à un lien entre deux objets :

- un objet dont les atomes contiennent les contraintes à respecter par l'interpolateur DSI. Cet objet doit être composé d'atomes, et sera donc appelé *Atomique*,

- un objet qui contient les données utilisées par les contraintes. Cet objet sera donc appelé *Contrôleur*. Il n'y a aucune restriction quant au type de cet objet.

- **Remarque :**
  Il arrive que l'objet Atomique et l'objet Contrôleur soient confondus. Quelques exemples de ces contraintes seront présentés ultérieurement,

Le modèle Atomique-Contrôleur permet d'implémenter les contraintes DSI sous forme de *lien* entre l'objet Contrôleur et l'objet Atomique.

2.2.1 Définition d'un lien

Un lien permet de connecter dynamiquement deux objets. Pour illustrer cette notion de lien, considérons les trois objets de la figure 2.2. La surface S, attirée par l'ensemble des points V, est filmée par la caméra. Donc cette surface possède deux types de liens :

- un lien avec la caméra. Les informations contenues dans ce lien permettent d'obtenir une image de la surface. Si on modifie la position de la surface et/ou de la caméra, les informations contenues dans ce lien doivent être changées de manière à modifier l'image obtenue par la caméra. Le lien est donc dit dynamique. Le même raisonnement peut être appliqué au lien entre l'ensemble des points et la caméra,

- un lien avec l'ensemble des points. Ce lien contient toutes les informations permettant d'attirer la surface vers l'ensemble des points. Si on change la position de l'ensemble des points et/ou de la surface, les informations contenues dans ce lien doivent être mises à jour pour permettre à l'ensemble de points d'attirer la surface. Dans ce cas aussi le lien est dynamique,

Pour faciliter l'implémentation des contraintes DSI en utilisant le modèle Atomique-Contrôleur, nous avons choisi l'approche de la programmation orientée objet

2.2.2 Éléments de la programmation orientée objet

La P.O.O.[1] est devenue populaire depuis quelques années. Certains auteurs parlent même d'un nouveau *paradigme* qui va permettre d'accroître la

1. abréviation de Programmation Orientée Objet

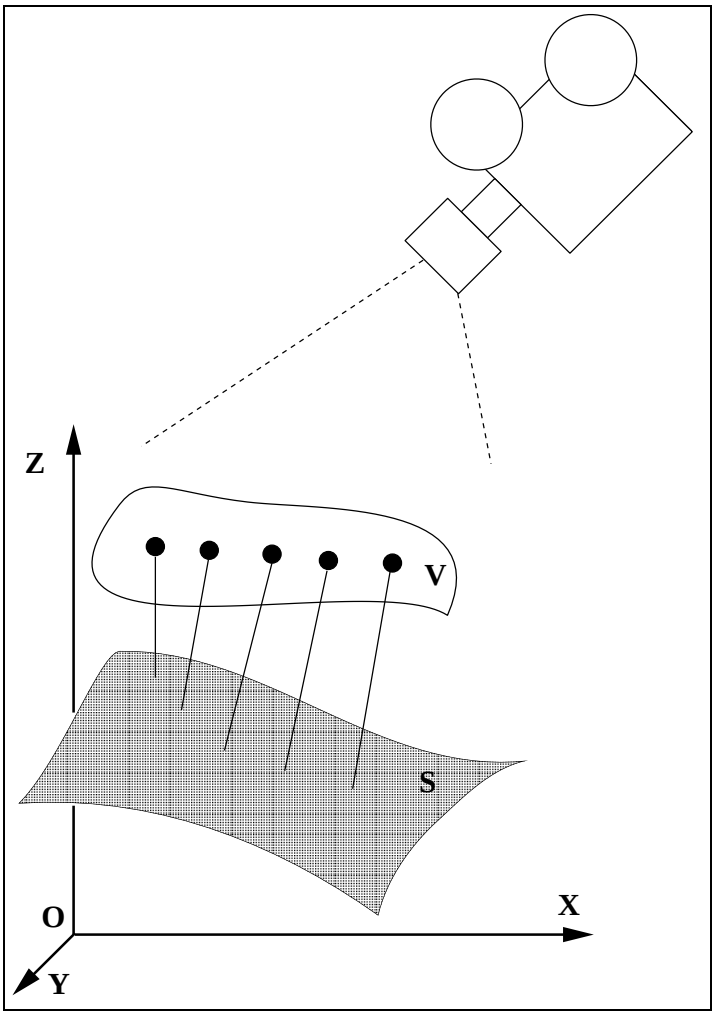

FIG. 2.2 – *Exemple de relations entre objets, d'après S. Collet*

productivité, et améliorer la fiabilité des logiciels ([7]). Pour illustrer quelques idées directrices de la P.O.O. , **Timothy Budd** raconte l'histoire suivante, dans le premier chapitre de son livre intitulé **"Introduction à la programmation par objets"**([7]).

Supposons que je veuille envoyer des fleurs à ma grand-mère (qui s'appelle Elsie) pour son anniversaire. Elle habite dans une ville située à quelques milliers de kilomètres; il est, donc, hors de question que je puisse les cueillir et les lui porter moi-même. Par contre, les envoyer est assez facile. En effet, il me suffit d'aller voir Flo, ma fleuriste préférée, et de lui décrire le type et le nombre de fleurs que je souhaite envoyer; je suis alors certain que les fleurs arriveront. ··· Si je procédais à une enquête je verrais que Flo envoie un message légèrement différent à un autre fleuriste habitant dans la ville de ma grand-mère. Celui-ci compose le bouquet et le confie, avec un message à nouveau différent, à un livreur, et ainsi de suite.

Pour envoyer des fleurs à sa grand-mère, T. Budd a dû envoyer un message, contenant sa demande, à un agent approprié. L'agent approprié est représenté par sa fleuriste. Pour satisfaire la demande de T. Budd, la fleuriste utilise une méthode dont le procédé et les détails sont inconnus de T. Budd. Ce dernier conclut son paragraphe en précisant que la recherche du procédé lui permettant d'envoyer des fleurs à sa grand-mère, coïncide avec le premier principe de la programmation orientée objet, qui consiste à trouver le mécanisme par lequel les activités sont initiées. En P.O.O. , une action est déclenchée par la transmission d'un *message* à un objet responsable de l'action. Ce message contient l'action demandée, et un ensemble d'informations nécessaires à son exécution. Si l'objet qui reçoit le message l'accepte, il se charge d'exécuter l'action indiquée dans le message en appliquant une méthode.

L'histoire imaginée par T. Budd permet de mettre en évidence la notion de l'encapsulation de l'information dans le mécanisme de l'envoi de messages. L'objet qui émet la demande n'a pas besoin de savoir quel moyen sera utilisé par le receveur pour satisfaire la demande.

Une autre innovation de la P.O.O. est l'introduction de la notion de *classe*. Dans l'histoire de T. Budd, la fleuriste est une instance de la classe **FLEURISTE** qui désigne la catégorie de tous les fleuristes. Toutes les instances de la classe **FLEURISTE** utilisent la même méthode (ou fonction membre) pour répondre au même message. Mais la fleuriste est tout d'abord une commerçante au même titre que le livreur. Les deux sont par exemple inscrits à la chambre de commerce. Donc, tout ce qui est valable pour un commerçant est valable pour le livreur ou la fleuriste. On peut définir une classe **COMMERÇANT** qui contient toutes les informations communes à toutes les

catégories de commerçants. La classe FLEURISTE et la classe LIVREUR vont donc hériter un certain nombre de caractéristiques de la **classe de base COMMERÇANT**. C'est le mécanisme d'héritage entre classes. La classe **COMMERÇANT** est aussi qualifiée de classe *abstraite* car elle n'a aucune existence réelle ; la classe **COMMERÇANT** n'existe qu'au travers des classes qui en dérivent (fleuriste,livreur,$\cdots$)

En appliquant le raisonnement de la P.O.O. aux contraintes de DSI, on a défini un couple de classes de base *abstraites* (CNSTR_INFO,CNSTR), qui sont à l'origine de toutes les contraintes DSI. La classe de base CNSTR_INFO correspond au lien entre l'objet Atomique et l'objet Contrôleur. Cette classe dérive de la classe de base GLINK qui décrit le lien entre deux objets. Parmi les données et les fonctions membres introduites par la classe de base CNSTR_INFO on distingue (cf figure 2.3) :

- un container pour toutes les instances de la classe de base CNSTR. Il s'agit des contraintes appliquées sur l'objet Atomique,

- un ensemble de fonctions membres (ou méthodes) *virtuelles* permettant la manipulation des contraintes. Ces fonctions sont dites virtuelles car elles ne peuvent être définies que par les classes dérivant de CNSTR_INFO.

La classe CNSTR définit les contraintes qui seront attachées aux atomes de l'objet Atomique. Cette classe contient :

- des données utilisées par l'interpolateur DSI pour respecter une contrainte,

- un ensemble de fonctions virtuelles dont par exemple :

  - les fonctions qui se chargent de créer ou de détruire une instance de la classe CNSTR,

  - la fonction appelée par l'interpolateur DSI pour prendre en compte une instance de la contrainte CNSTR comme une contrainte floue,

  - la fonction appelée par l'interpolateur DSI pour prendre en compte l'instance de la classe CNSTR comme une contrainte dure.

.

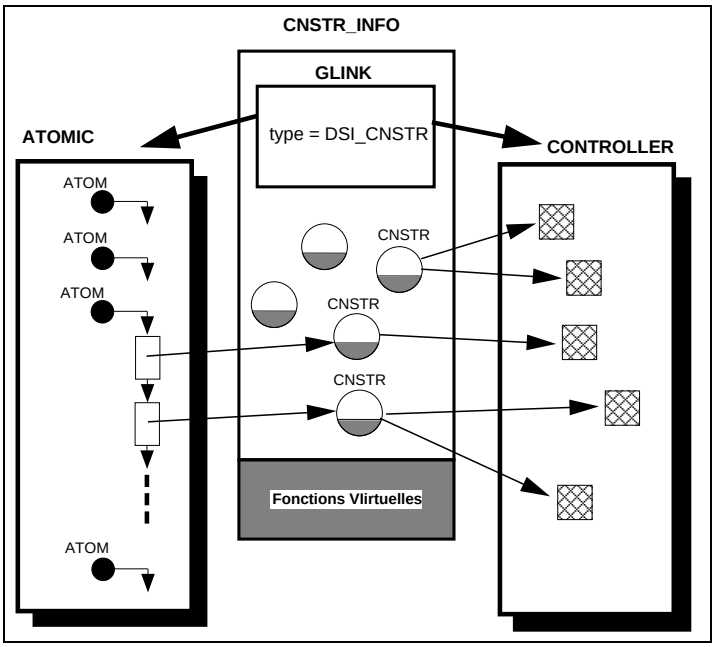

FIG. 2.3 – *Exemple du modèle Atomique-Contrôleur*

Remarque :

Il est nécessaire, lors d'une modification géométrique ou topologique de l'objet Atomique et/ou l'objet Contrôleur, de pouvoir mettre à jour les informations concernant les contraintes. Grâce au mécanisme de *superviseur de projet* développé par S. Collet[8], il est possible de détecter toute modification sur un objet et de mettre à jour toutes les contraintes qui concernent l'objet modifié. Cette mise à jour se fait en deux étapes :

1. avant la modification de l'objet, le superviseur de projet appelle la fonction membre de la classe définissant la contrainte qui permet de retirer toutes les contraintes à partir des atomes de l'objet Atomique. La fonction appelée dépend du type de l'objet modifié (Atomique ou Contrôleur),

2. après la modification de l'objet Atomique ou Contrôleur, le superviseur de projet appelle la fonction de mise en place des contraintes sur les atomes de l'objet Atomique. La fonction appelée dépend du type de l'objet modifié (Atomique ou Contrôleur).

Le mécanisme général de la mise en place des contraintes selon le modèle Atomique-Contrôleur, consiste à associer à chaque nouvelle contrainte XXX un couple de classes (XXX_INFO, XXX) tel que :

- XXX_INFO dérive directement ou indirectement de la classe de base CNSTR_INFO. La classe XXX_INFO peut redéfinir les fonctions membres virtuelles de la classe CNSTR_INFO. Elle peut aussi implémenter de nouvelles informations (fonctions membres et données) qui sont propres à la classe XXX_INFO et des classes qui en dérivent,

- XXX est une classe qui dérive, directement ou indirectement, de la classe de base CNSTR. La classe XXX contient des informations qui seront utilisées par l'interpolateur DSI.

Grâce au modèle Atomique-Contrôleur, et aux concepts de la programmation orientée objet, il est facile d'implémenter de nouvelles contraintes DSI. L'arbre de dérivation de quelques contraintes implémentées dans le système GOCAD, est présenté figure 2.4. Sur cette figure nous pouvons distinguer deux types de contraintes : Les contraintes de base auxquelles est associé un couple (XXX_INFO, XXX) de classes abstraites, et les contraintes réelles auxquelles est associé un couple de classes réelles.

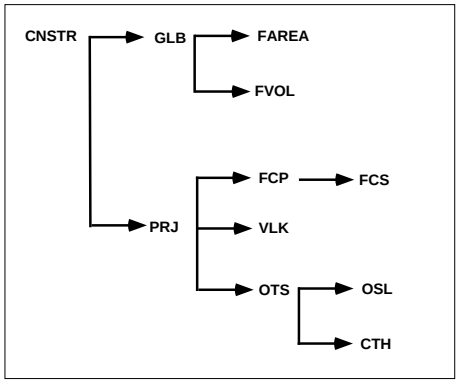

FIG. 2.4 – *Arbre de dérivation de quelques contraintes DSI*

2.2.2.1 Contraintes de base

Ces contraintes rassemblent l'ensemble des fonctionnalités communes à un groupe de contraintes. On distingue trois contraintes de base :

1. la contrainte CNSTR, qui est à la base de toutes les contraintes DSI,
2. la contrainte GLB[2] qui s'applique sur tous les atomes de l'objet Atomique,
3. la contrainte PRJ[3], qui est à la base de toutes les contraintes DSI utilisant la projection de nœuds sur une surface appelée *cible*. A l'inverse de la contrainte GLB, seuls quelques atomes de l'objet Atomique seront concernés par les contraintes de type PRJ.

En raison de l'utilisation importante des contraintes projetées, nous allons présenter les classes de base (PRJ_INFO,PRJ) correspondant à la contrainte PRJ.

La classe de base PRJ

Cette classe introduit un ensemble de données et de fonctions membres permettant d'effectuer et de contrôler la procédure de projection des nœuds sur la surface cible. Parmi ces données on distingue :

- Une instance de la classe ATOMSHOOT qui dérive de la classe ATOM (cf Annexe). La classe ATOMSHOOT introduit les nouveaux champs suivants (voir figure 2.5) :
  - un vecteur qui définit la direction de projection,
  - une référence vers le triangle, de la surface cible, intersecté lors de la procédure de projection,
  - le point d'impact déterminé par la procédure de projection,
  - un tableau des coordonnées barycentriques, du même point d'impact, par rapport aux sommets du triangle intersecté.

  Parmi les fonctions membres introduites par la classe PRJ nous distinguons :
  - une fonction chargée, d'une part d'effectuer la procédure de projection, et d'autre part de la mise à jour de la référence vers le

2. abréviation de contrainte globale
3. abréviation de contrainte projetée

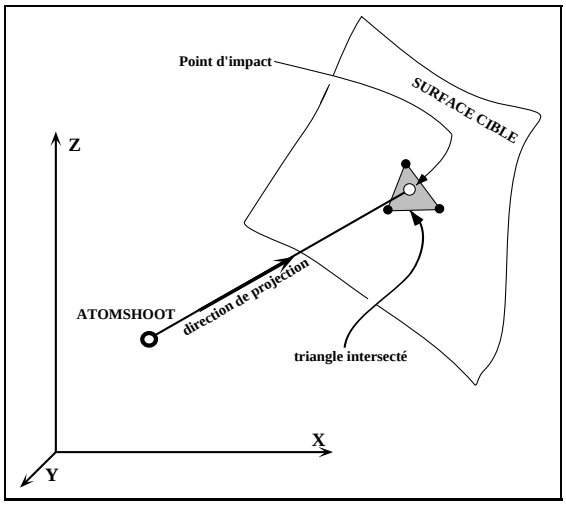

FIG. 2.5 – *Projection d'un nœud sur une surface*

triangle intersecté. Cette fonction sera héritée par les classes dérivées,

- des fonctions abstraites qui permettent d'installer ou de retirer des instances de la classe PRJ, des atomes du triangle intersecté. Ces procédures permettent la mise à jour automatique des contraintes lors de la modification de l'objet Atomique ou Contrôleur,
- une fonction, héritée par les classes dérivées, chargée de calculer les coordonnées euclidiennes du point d'impact, ainsi que les coordonnées barycentriques du point d'impact par rapport aux sommets du triangle intersecté.

La classe de base PRJ_INFO

La classe PRJ_INFO implémente de nouvelles données ainsi que de nouvelles fonctions membres, dont :

- la référence de la surface *cible*,
- la référence de l'objet qui contient toutes les instances de la classe ATOMSHOOT décrite précédemment,
- une fonction qui permet d'optimiser les directions de projections selon le critère de *proximité*. La direction de projection optimisée correspon-

dra au vecteur qui joint le point à projeter au point de la surface cible le plus proche (cf figure 2.6)

– une fonction membre qui permet d'interpoler, en utilisant la méthode DSI, les directions de projections (cf figure 2.7).

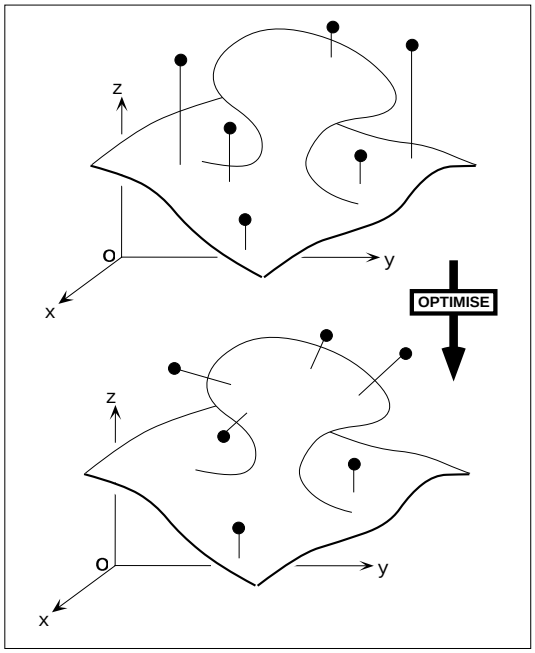

FIG. 2.6 – *Optimisation des directions de projections*

2.2.2.2 Contraintes réelles

Parmi les nombreuses contraintes existant dans le système GOCAD on distingue :

– *FCS:* (abréviation de Fuzzy Control Slope)
Cette contrainte permet à une surface de contrôler localement le pendage d'une autre surface. Sur la figure 2.8, le pendage de la surface S est contrôlé par celui de la surface $\mathcal{G}$. Cette contrainte est prise en compte dans l'équation DSI de manière floue. L'objet Atomique et la surface cible correspondent à la surface S. La surface $\mathcal{G}$ correspond à

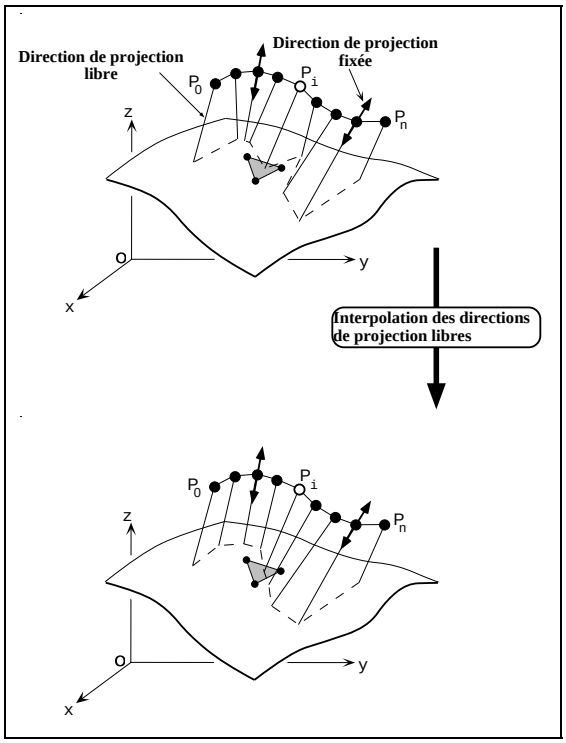

FIG. 2.7 – *Interpolation par DSI des directions de projections*

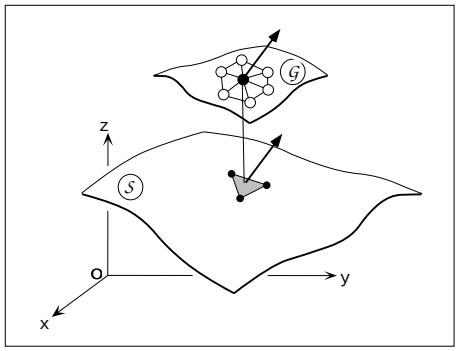

FIG. 2.8 – *Exemple de la contrainte Fuzy Control Slope*

l'objet Contrôleur. La direction de projection est héritée de la classe de base PRJ,

– *OSL*(abréviation de On-Straight-Line)
Cette contrainte, illustrée par la figure 2.9, oblige les nœuds du bord $\mathcal{B}_i$ d'une surface à se déplacer le long d'une droite. Cette contrainte est prise en compte de manière dure dans l'équation de DSI. L'objet Atomique, l'objet Contrôleur et la surface cible sont confondus. La direction de la droite, le long de laquelle les nœuds du bord $\mathcal{B}_i$ doivent se déplacer, correspond à la direction de projection de la classe de base PRJ,

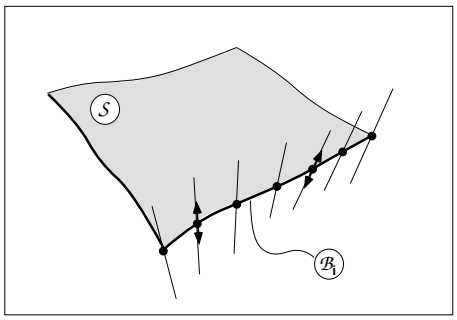

FIG. 2.9 – *Exemple de la contrainte On-Straight-Line*

– *CTH* (abréviation de Control-Thickness)
Il s'agit d'une contrainte floue, qui indique que chaque atome de la surface S doit être à une certaine distance de la surface Σ le long de la direction $\overrightarrow{D}$ (cf figure 2.10). L'objet Contrôleur et la surface cible sont représentés par la surface Σ. L'objet Atomique est représenté par la surface S.

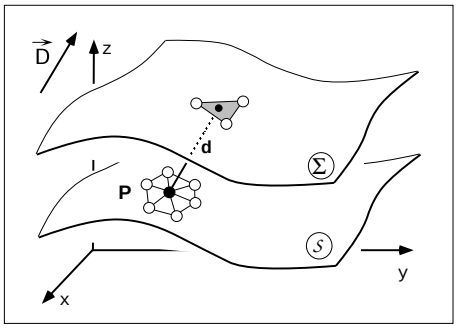

FIG. 2.10 – *Exemple de la contrainte de contrôle d'épaisseur entre deux surfaces*

Après avoir présenté brièvement le principe des contraintes DSI, nous nos proposons d'étudier en détail une des contraintes les plus utilisées dans le domaine de la géologie, il s'agit de la contrainte Fuzzy Control Point

2.2.3 La contrainte Fuzzy Control Points

Cette contrainte a été mise en place par P. Le Mélinaire([36]), pour résoudre le problème d'ajustement d'une surface $\mathcal{S}$ à un ensemble $\mathcal{V}$ de données réelles (cf figure 2.11). Pour ce type de contrainte, l'ensemble des données réelles correspond à l'objet *Contrôleur* et la surface à ajuster correspond à l'objet *Atomique* et à la surface *cible*.

Le principe de cette contrainte consiste à associer à chaque point $\mathbf{P}$ de $\mathcal{V}$

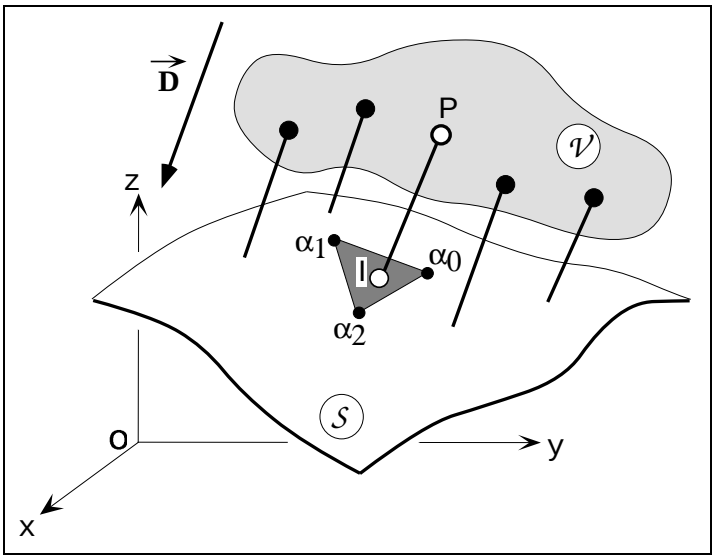

FIG. 2.11 – *Exemple de la contrainte Fuzzy Control Points*

une direction de tir définie par un vecteur $\overrightarrow{D}$. Suivant cette direction, le point $\mathbf{P}$ intercepte un triangle de la surface cible en un point d'impact $\mathbf{I}$. Si $\mathbf{P}$ n'intercepte aucun triangle de la surface à ajuster, la surface ne sera pas attirée vers le point $\mathbf{P}$.

Le but de cette contrainte est de minimiser la distance $d(\mathbf{P},\mathbf{I})$ qui sépare les deux points $\mathbf{P}$ et $\mathbf{I}$. Pour ce faire, il est évidemment nécessaire de calculer la position du point $\mathbf{I}$ relativement aux sommets du triangle intersecté ; ceci est précisément l'objet du paragraphe suivant.

2.2.3.1 Calcul de la position d'un point par rapport à un triangle

Considérons sur la figure 2.12 :

- le triangle $T(\mathbf{p_0}, \mathbf{p_1}, \mathbf{p_2})$,
- le point $\mathbf{P}$,
- une direction $\overrightarrow{D}$ non coplanaire du triangle $\mathbf{T}$,
- le point $\mathbf{p}$, projection de $\mathbf{P}$ sur le triangle T, selon la direction $\overrightarrow{D}$.

Si $\mathbf{O}$ est l'origine de l'espace $3D$, nous supposons dans la suite que le vecteur $\overrightarrow{OQ}$ ainsi que la matrice colonne des coordonnées du point $\mathbf{Q}$, seront notés $\mathbf{Q}$. Cela nous permet d'écrire :

$$\left| \begin{array}{rl} \mathbf{p} & = u \cdot \mathbf{p_1} + v \cdot \mathbf{p_2} + w \cdot \mathbf{p_0} \\ (\mathbf{P - p_0}) & = u \cdot \mathbf{U} + \mathbf{v \cdot V + d \cdot D} \\ \text{avec :} & \left\{ \begin{array}{rcl} \mathbf{U} & = & \mathbf{p_1 - p_0} \\ V & = & \mathbf{p_2 - p_0} \\ w & = & 1 - u - v \end{array} \right. \end{array} \right.$$

$(\mathbf{w}, \mathbf{u}, \mathbf{v})$ sont les coordonnées barycentriques de $\mathbf{p}$ par rapport à $\{\mathbf{p_0}, \mathbf{p_1}, \mathbf{p_2}\}$, alors que $\{\mathbf{u}, \mathbf{v}, \mathbf{d}\}$ sont les coordonnées du point $\mathbf{P}$ relativement à la base $\{\mathbf{U}, \mathbf{V}, \mathbf{D}\}$.
Si $(\mathbf{U^x}, \mathbf{U^y}, \mathbf{U^z})$, $(\mathbf{V^x}, \mathbf{V^y}, \mathbf{V^z})$ et $(\mathbf{D^x}, \mathbf{D^y}, \mathbf{D^z})$ sont les coordonnées respectives des vecteurs $\mathbf{U}$, $\mathbf{V}$ et $\mathbf{D}$ dans la base (ox, oy, oz), on peut alors écrire :

$$\underbrace{\begin{bmatrix} U^x & V^x & D^x \\ U^y & V^y & D^y \\ U^z & V^z & D^z \end{bmatrix}}_{\mathbf{M}} \cdot \begin{bmatrix} u \\ v \\ d \end{bmatrix} = \mathbf{P - p_0}$$

Si $(\mathbf{C_u^x}, \mathbf{C_u^y}, \mathbf{C_u^z})$, $(\mathbf{C_v^x}, \mathbf{C_v^y}, \mathbf{C_v^z})$ et $(\mathbf{C_d^x}, \mathbf{C_d^y}, \mathbf{C_d^z})$ sont les composantes des trois vecteurs $\mathbf{C_u}$, $\mathbf{C_v}$ et $\mathbf{C_d}$ correspondant aux lignes de la matrice $\mathbf{M^{-1}}$:

$$\mathbf{M^{-1}} = \begin{bmatrix} C_u^x & C_u^y & C_u^z \\ C_v^x & C_v^y & C_v^z \\ C_d^x & C_d^y & C_d^z \end{bmatrix}$$

Les coordonnées (u, v, d) peuvent être calculées en utilisant la formule suivante :

$$\left| \begin{array}{rcl} u & = & C_u \cdot (\mathbf{P - p_0}) \\ v & = & C_v \cdot (\mathbf{P - p_0}) \\ d & = & C_d \cdot (\mathbf{P - p_0}) \end{array} \right.$$

La coordonnée barycentrique w peut être déduite de la formule précédente car $\mathbf{w = 1 - u - v}$.

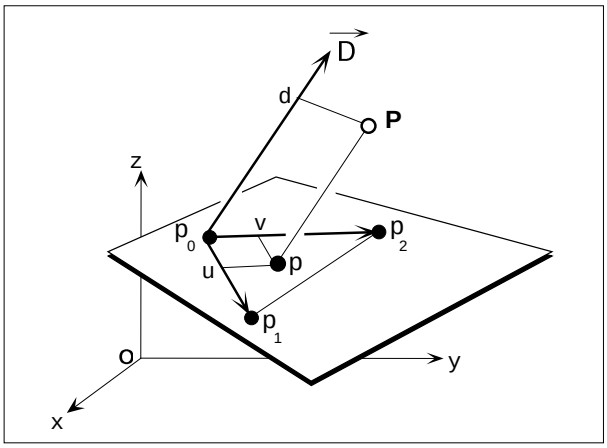

FIG. 2.12 – *Calcul de la position d'un point par rapport à un triangle*

2.2.3.2 Formulation mathématique de la contrainte FCP

Revenons à l'exemple de la figure 2.11, après avoir calculé les coordonnées barycentriques (w, u, v) du point $\mathbf{I}$ par rapport à $(\alpha_0, \alpha_1, \alpha_2)$, les sommets du triangle intersecté, on peut exprimer la contrainte FCP de la manière suivante :

$$\left|\begin{array}{l} \forall \nu = (x, y, z) : w \cdot \varphi^\nu(\alpha_0) + u \cdot \varphi^\nu(\alpha_1) + v \cdot \varphi^\nu(\alpha_2) = P^\nu \\ \varphi(\alpha) \text{ désigne les coordonnées euclidiennes du nœud } \alpha \\ \mathbf{P}^x, \mathbf{P}^y, \mathbf{P}^z \text{ sont les coordonnées euclidiennes du point } P \\ u + v + w = 1 \end{array}\right. \tag{2.4}$$

Au nœud β de la surface à ajuster, cette contrainte s'exprime par la formule :

$$\forall \nu = (x, y, z) : \left[\begin{array}{l} \displaystyle\sum_{\beta \in \{\alpha_0, \alpha_1, \alpha_2\}} A_i^\nu(\beta) \cdot \varphi^\nu(\beta) = B_i^\nu \\ \text{avec} : \left|\begin{array}{ll} A_i^\nu(\beta) & = \left\{\begin{array}{ll} w & \text{si } \beta = \alpha_0 \\ u & \text{si } \beta = \alpha_1 \\ v & \text{si } \beta = \alpha_2 \end{array}\right. \\ B_i^\nu & = \mathbf{P}^\nu \end{array}\right. \end{array}\right. \tag{2.5}$$

Les coefficients $\Gamma_i^\nu(\alpha)$ et $\gamma_i^\nu(\alpha)$, permettant de prendre en compte cette contrainte dans l'équation de DSI, se calculent comme suit :

$$\begin{aligned}
\Gamma_i^\nu(\alpha) &= \begin{cases} \varpi_i^2 \cdot w \quad \cdot(u \cdot \varphi^\nu(\alpha_1) + v \cdot \varphi^\nu(\alpha_2) - \mathbf{P}^\nu) \text{ si } \alpha = \alpha_0 \\ \varpi_i^2 \cdot u \quad \cdot(v \cdot \varphi^\nu(\alpha_1) + w \cdot \varphi^\nu(\alpha_0) - \mathbf{P}^\nu) \text{ si } \alpha = \alpha_1 \\ \varpi_i^2 \cdot v \quad \cdot(w \cdot \varphi^\nu(\alpha_0) + w \cdot \varphi^\nu(\alpha_1) - \mathbf{P}^\nu) \text{ si } \alpha = \alpha_2 \end{cases} \\
\gamma_i^\nu(\alpha) &= \begin{cases} \varpi_i^2 \cdot (w)^2 \quad \forall \alpha = (\alpha_0, \alpha_1, \alpha_2) \\ 0 \qquad\qquad\qquad \text{sinon} \end{cases}
\end{aligned} \tag{2.6}$$

La contrainte qui minimise la distance $d(\mathbf{P}, \mathbf{I})$ se répartit sur les trois sommets du triangle intersecté. A chaque itération de DSI les sommets du triangle intersecté se déplaceront de façon à réduire l'écart entre les points $\mathbf{P}$ et $\mathbf{I}$. Cependant, lors de la même itération, DSI tente de réduire la rugosité de la surface, il est donc possible que le triangle intercepté se trouve décalé de la direction de projection à la suite d'une itération de DSI. Il est donc nécessaire de s'assurer qu'à tout moment au cours des itérations de DSI le point d'impact $\mathbf{I}$ appartient au triangle intercepté. Si ce n'est pas le cas, une procédure calcule le nouveau triangle de la surface qui contient ce point d'impact. Le triangle intercepté correspond alors au nouveau triangle qui contient le point d'impact. Afin de faciliter la procédure de recherche du nouveau triangle, nous avons adopté la convention suivante (cf figure 2.13) : **Considérons un triangle p_trgl. La position du triangle adjacent p_trgl[i] se situe du côté de l'arête opposée au sommet p_atom[i] de p_trgl**

Puisque les valeurs des coordonnées barycentriques dépendent de l'indiçage des sommets des triangles, il est facile de suivre l'évolution d'un point d'impact.

Sur l'exemple de la figure 2.14, la coordonnée barycentrique u du point $\mathbf{I}$ est négative, donc $\mathbf{I}$ se trouve du côté du triangle p_trgl[1]. Cette information permet grâce à une procédure de boucles de retrouver le triangle qui contient réellement le point d'impact. Un triangle $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ contient un point d'impact $\mathbf{I}$, si et seulement si les coordonnées barycentriques (u, v, w) de $\mathbf{I}$, par rapport à $(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$, vérifient la relation suivante :

$$\begin{cases} u \in [0,1] \\ v \in [0,1] \\ u + v \in [0,1] \\ w = 1 - u - v \end{cases}$$

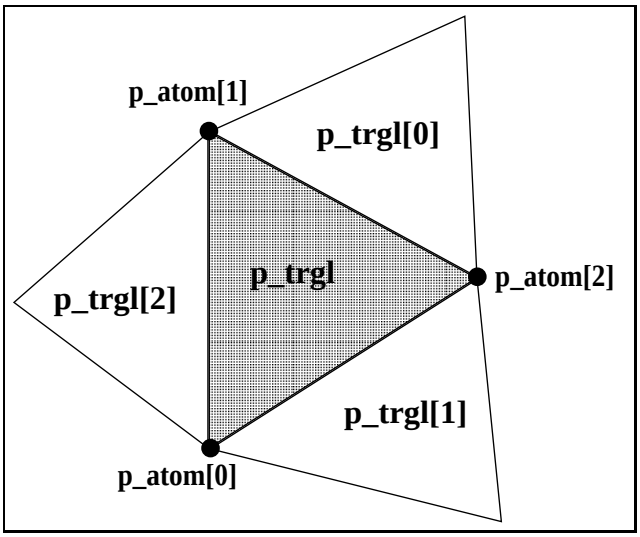

FIG. 2.13 – *Convention de numérotation des triangles*

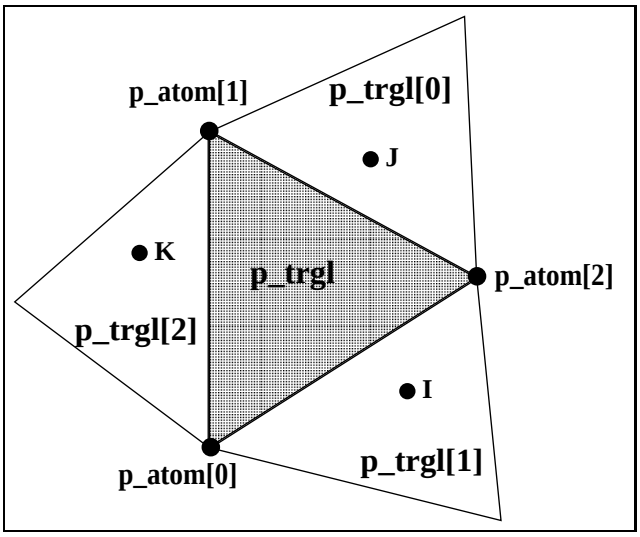

FIG. 2.14 – *Interprétation d'une coordonnée barycentrique* (u, v, w) *négative, d'après P. Le Melinaire*

Si le mécanisme boucle, une procédure de calcul direct du triangle intersecté sera utilisée. Cette procédure, appelée TSURF_Shoot(), a été mise

en place par Y. Huang([24]) et consiste , comme le montre la figure 2.15, à calculer la liste des triangles de la surface H qui sont intersectés par la demi droite issue du nœud **P** et de direction $\overrightarrow{D}$. Comme le montre la figure 2.15, il se peut que plusieurs triangles de la surface H soient intersectés par la demi-droite. Dans ce cas la procédure TSURF_Shoot() retourne le triangle le plus proche de **P**, ainsi que les coordonnées du point d'impact.

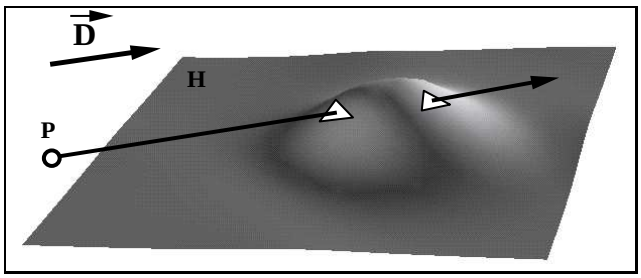

FIG. 2.15 – *Intersection d'une demi-droite et d'une surface*

2.2.3.3 Initialisation de la contrainte FCP

La contrainte FCP est une contrainte projetée, donc les classes FCP_INFO et FCP dérivent respectivement de PRJ_INFO et PRJ. Selon le principe de la programmation orientée objet, les fonctions membres virtuelles des classes PRJ_INFO et PRJ sont définies au niveau des classes FCP_INFO et FCP. Parmi ces fonctions on retrouve :

- les fonctions de création et de destruction des instances de la classe FCP,
- la fonction membre permettant à DSI de prendre en compte une instance de la classe FCP comme une contrainte floue. Cette fonction calcule, selon la formule 2.6, les coefficients Γ_i et γ_i au niveau des atomes de chaque triangle intersecté de la surface cible. Je rappelle que ces coefficients permettent à DSI de respecter, de manière floue, la contrainte FCP,
- les fonctions de mise à jour des contraintes si un des deux objets, Contrôleur ou Atomique, a été modifié,
- la fonction qui calcule l'erreur d'ajustement (ou *la projection inverse*) d'une surface aux données qui l'attirent. Si nous considérons l'exemple

de la figure 2.11, l'erreur d'ajustement, $E(T)$, du triangle $T(\alpha_0, \alpha_1, \alpha_2)$ au point P est donnée par la formule suivante :

$$E(T) = \| \overrightarrow{(P, I)} \|$$

Cette erreur sera considérée comme une propriété physique attachée aux trois atomes du triangle T. Il est donc possible de visualiser cette erreur à l'aide de courbes d'isovaleur (cf chapitre 1),

La mise en place d'une contrainte FCP entre une surface et un ensemble de points, se fait de la manière suivante :

- Quand l'utilisateur spécifie interactivement le couple (Contrôleur, Atomique), la fonction TSURF_Set_FCP() se charge de créer une instance de la classe FCP_INFO qui correspond au lien de type contrainte FCP entre l'objet Atomique et l'objet Contrôleur. Par défaut les données sont supposées avoir le même poids, elles attirent donc la surface avec la même force. Ensuite, la fonction TSURF_Shoot() est appelée automatiquement, par la fonction TSURF_Set_FCP(), pour initialiser les triangles intersectés ainsi que les points d'impact. Pour chaque point de données intersectant un triangle $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ de la surface, une instance de classe FCP sera ajoutée à la liste des contraintes attachées à chaque atome du triangle T.
  Sur la figure 2.16, les points de données de l'objet Contrôleur sont matérialisées par des cubes pleins. Les lignes blanches correspondent aux segments qui joignent chaque point de données au point d'impact qui lui correspond. Pour maintenir la tension sur les bords de la surface nous avons mis en place des contraintes de type $On - Straight - Line$. Sur la même figure 2.16, pour chaque nœud du bord de la surface, la direction de déplacement autorisé est donnée, par un segment rouge.

- Après interpolation par DSI de la surface S, cette surface s'ajuste aux données (cf figure 2.17). La figure 2.18 montre que le maillage de la surface n'est pas perturbé par la disposition des données.

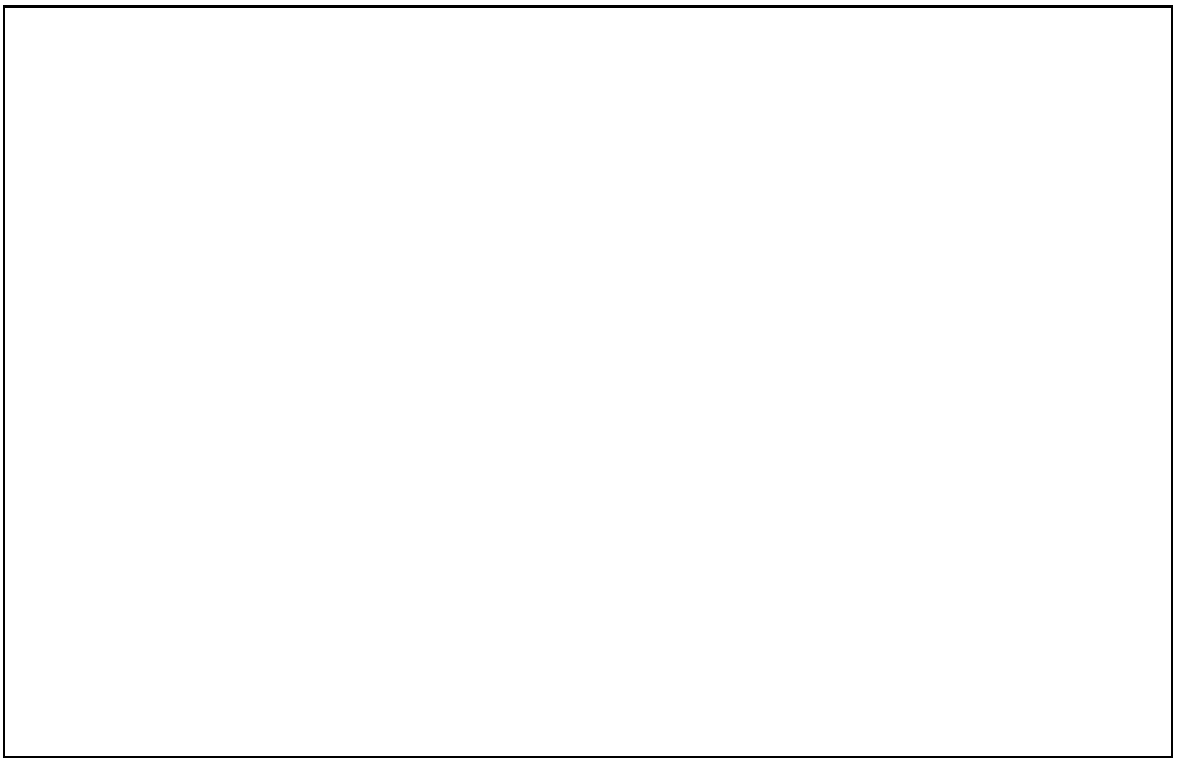

FIG. 2.16 – *Mise en place des contraintes FCP sur une surface à partir d'un semis de points*

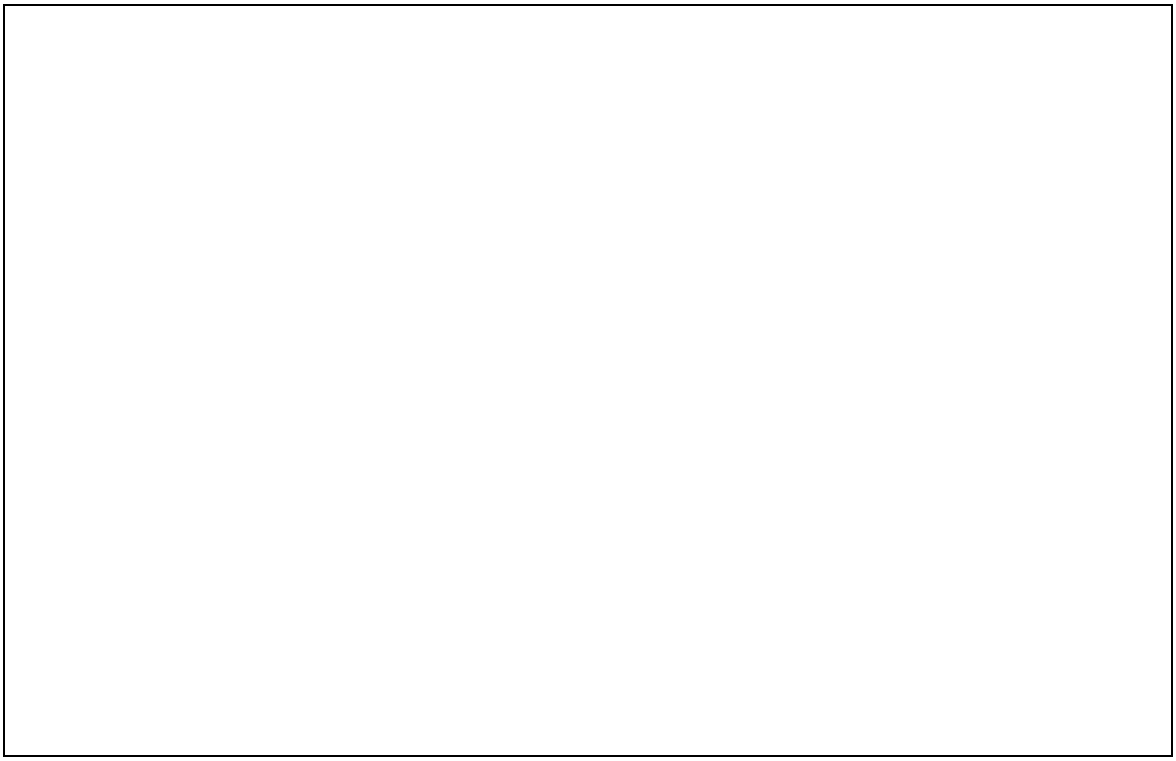

FIG. 2.17 – *Résultat de l'interpolation de la surface par DSI*

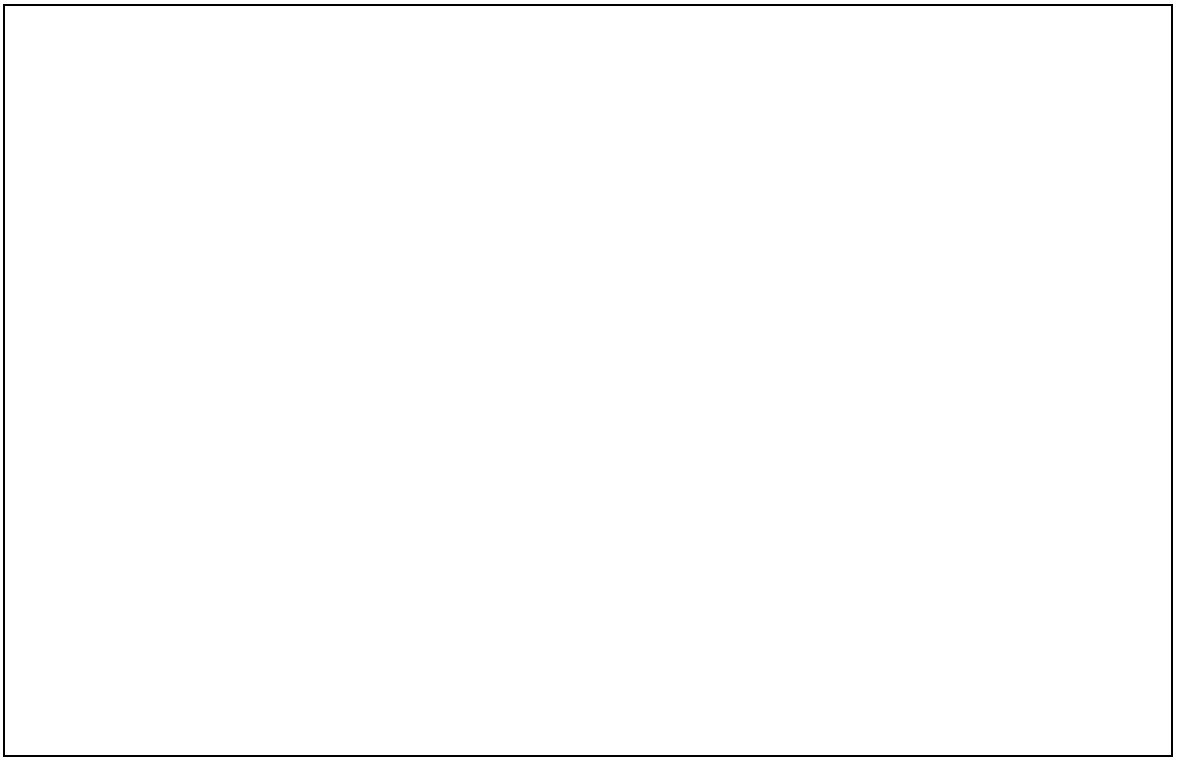

FIG. 2.18 – *Exemple montrant la régularité du maillage*

2.2.3.4 Conclusion sur la contrainte FCP

L'importance de la contrainte FCP, réside dans le fait qu'elle est déconnectée du maillage de la surface. Cela permet d'avoir des surface lisses qui ajustent correctement des données. Cependant, cette méthode présente quelques limites dont :

- *distribution des données*
  L'exemple de la figure 2.19, montre que dans ce cas certains triangles de la surface cible sont simultanément attirés par plusieurs points de données. Dans ce cas, l'intervention de l'utilisateur est nécessaire pour résoudre ce problème en modifiant interactivement la surface cible (cf figure 2.20), ou en modifiant la direction de projections de certains points de données (voir figure 2.21).

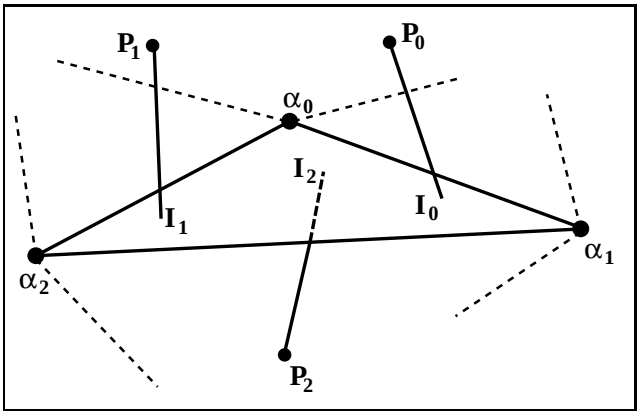

FIG. 2.19 – *Exemple de conflit dans la contrainte FCP*

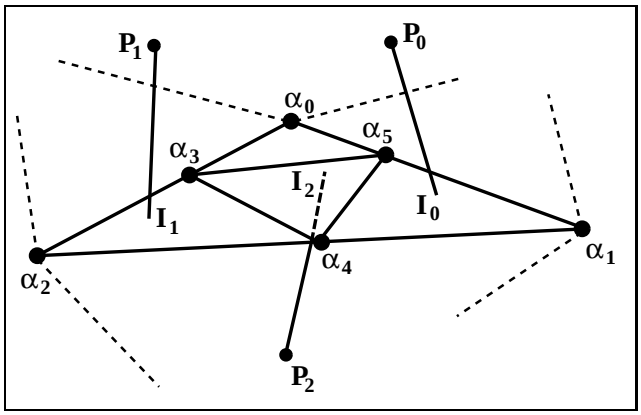

FIG. 2.20 – *Modification interactive de la surface cible par subdivision de certains triangles*

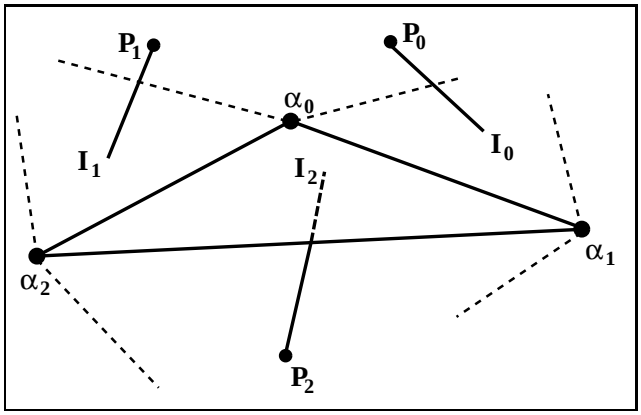

FIG. 2.21 – *Modification de la direction de projection de certains points de données*

2.3 Conclusion

Nous avons montré dans ce chapitre que le modèle Atomique-Contrôleur permet de simplifier la mise au point de nouvelles contraintes DSI. Le style de programmation orientée objet permet d'éliminer les redondances entre les différentes classes de contraintes, ceci va nous permettre de développer de nouvelles contraintes géométriques permettant de modéliser la relation géologique *Faille-Horizon*, qui est une relation importante dans le domaine de l'exploration et l'exploitation pétrolière et minière. La modélisation de cette relation sera présentée dans le chapitre 4, après avoir rappelé quelques notions de géologie dans le chapitre 3.

Chapitre 3

Éléments de géologie structurale

Les roches, constituant la croûte terrestre, présentent des déformations variées dues à des contraintes géologiques. La tectonique cassante se propose d'étudier les déformations subies par les couches après leurs dépôts. Suivant les conditions thermodynamiques des déformations et la rhéologie de la roche, on distingue les déformations géologiques cassantes et ductiles. Dans ce chapitre, nous présenterons un bref aperçu sur les déformations cassantes, et plus précisément les failles.

3.1 Les déformations cassantes

Elles se manifestent par des discontinuités d'origine mécanique, dans les couches géologiques. On parle alors de fractures. Les géologues distinguent plusieurs types de fractures en fonction de la morphologie, de l'extension, de l'ampleur du déplacement $\cdots$. On peut citer par exemple :

- Les diaclases :
  Comme le montre la figure 3.1, il s'agit de discontinuités sans déplacement des parties de part et d'autre des discontinuités,

- Les failles :
  Une faille est une fracture dont les deux compartiments montrent un

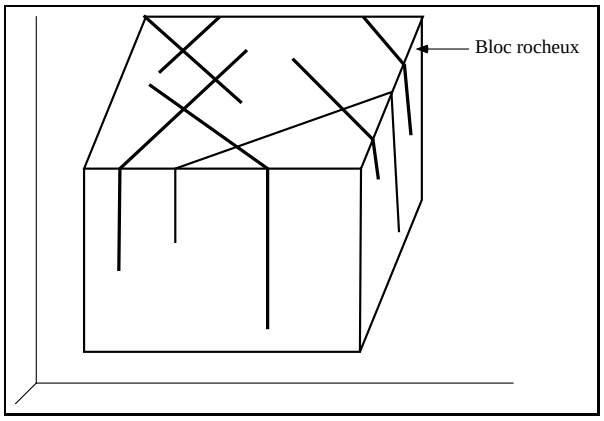

FIG. 3.1 – *Exemples de diaclases dans un bloc rocheux*

déplacement l'un par rapport à l'autre suivant une direction parallèle au plan de la faille.

Dans la suite de ce chapitre nous focaliserons notre étude sur les failles, en raison de l'importance capitale qu'elles revêtent dans le domaine de l'exploration pétrolière et minière.

3.2 Éléments descriptifs d'une faille

Une faille (cf figure 3.2) est constituée par deux compartiments décalés de part et d'autre d'une surface de glissement. La valeur du décalage est appelée *rejet*.

- *Compartiments d'une faille*
  On désigne par compartiments d'une faille, les deux parties d'une couche géologique séparées par une faille.

- *Surface de faille*
  On appelle surface de faille la zone de discontinuité où se concentre le décalage entre les deux compartiments de la faille. La notion de *surface de faille* est impropre, car la zone de discontinuité est souvent dotée d'une épaisseur non négligeable qui peut aller jusqu'à quelques mètres. Il faudrait donc parler plutôt de *zone de faille*.

- *Lèvres d'une faille*
  Si on considère une couche géologique affectée par une faille, les lèvres

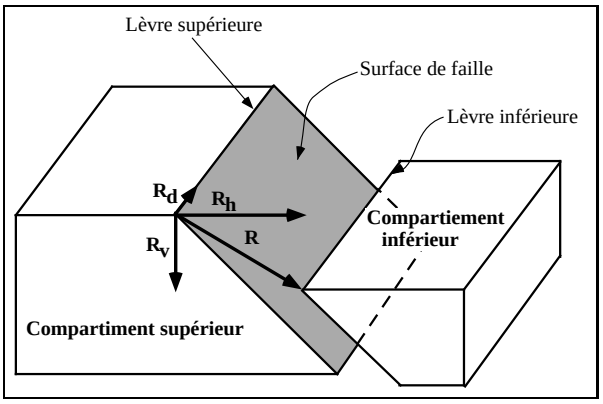

FIG. 3.2 – *Différents paramètres d'une faille*

de la faille correspondent aux surfaces limites des deux compartiments. Pour une meilleure appréhension de cette notion, on va désigner par lèvres de la faille les bords du toit ou du mur d'une couche qui sont au contact de la faille. On parle de *lèvre supérieure* et *lèvre inférieure* de la faille, selon que ces bords appartiennent au compartiment situé respectivement au-dessus ou au-dessous de la zone de discontinuité.

– *Jeu d'une faille*
C'est le phénomène de décalage des deux compartiments d'une faille.

– *Durée de vie d'une faille*
C'est le temps pendant lequel la faille a joué, cette durée peut aller de quelques secondes à des millions d'années.

– *Rejet*
Le rejet d'une faille est le vecteur représentant le déplacement relatif des compartiments, il est généralement calculé à partir de surfaces repères qu'on retrouve de part et d'autre de la faille. Comme le montre la figure 3.2 le vecteur rejet R peut être décomposé en : *rejet horizontal (Rh), rejet vertical (Rv) et rejet de décrochement (Rd).* La notion de rejet n'est pas très simple car une faille peut avoir plusieurs jeux au cours de la durée de vie de la faille. Le rejet qu'on observe sur le terrain, représente donc la somme de tous les rejets issus des jeux successifs de cette faille.

- *Stries de failles*
  Il s'agit de traces visibles sur la surface de faille dûes aux frottements lors du déplacement des compartiments de faille. Des stries de failles entrecroisées indiquent que la faille a eu divers jeux de sens variés.

- *Extension de la faille*
  L'extension de la faille est donnée par sa dimension le long d'une direction orthogonale au sens du déplacement. Sur la figure 3.2, l'extension est donnée par la droite E.

3.2.1 Catégories de failles

Le classement des failles peut se faire selon plusieurs critères géométriques. On peut alors parler de classement selon le type de rejet, l'attitude de la surface de faille, les rapports avec les surfaces avoisinantes $\cdots$. Pour ne pas alourdir l'exposé, nous présenterons le classement selon le type de rejet. Ainsi nous distinguerons :

1. *Les failles à faible composante de rejet de décrochement*
   Dans ce cas, c'est le rejet transversal[1] qui prédomine. Selon le sens de ce dernier on distingue :

   - Les failles normales
     On les appelle aussi failles directes. Le rejet horizontal correspond à une distension (cf figure 3.3). On parle aussi de failles en régime extensif.
   - Les failles inverses
     On les appelle aussi chevauchements. Le rejet horizontal correspond à un raccourcissement (voir figure 3.4). On parle de failles en régime compressif.

2. *Les failles à composante de rejet de décrochement prédominante*
   Ce sont les failles coulissantes ou de décrochement (cf figure 3.5).

1. le rejet transversal est la résultante du rejet vertical et du rejet horizontal

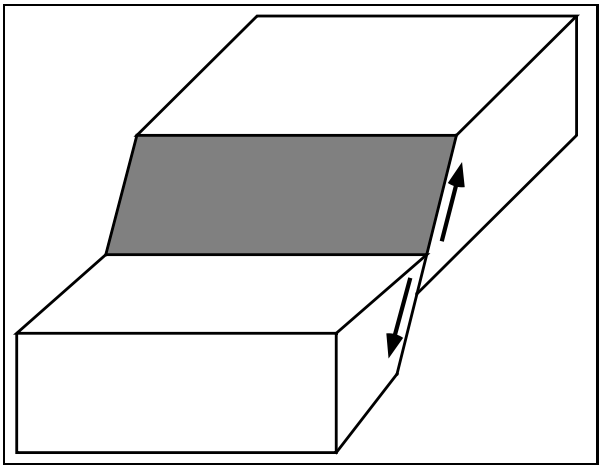

FIG. 3.3 – *Exemple d'une faille normale*

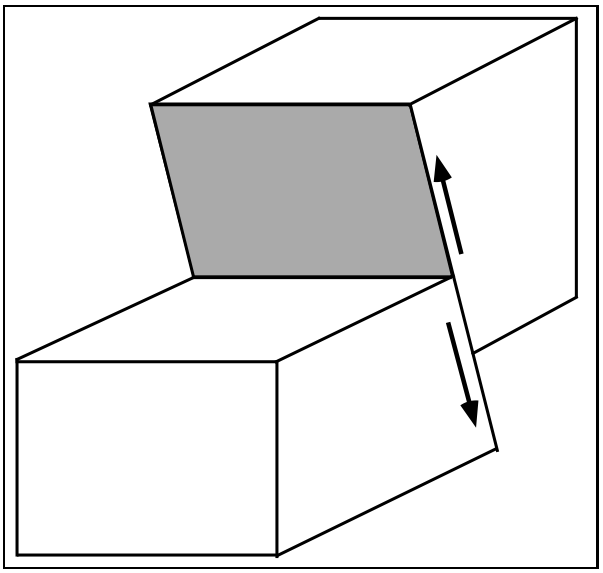

FIG. 3.4 – *Exemple d'une faille inverse*

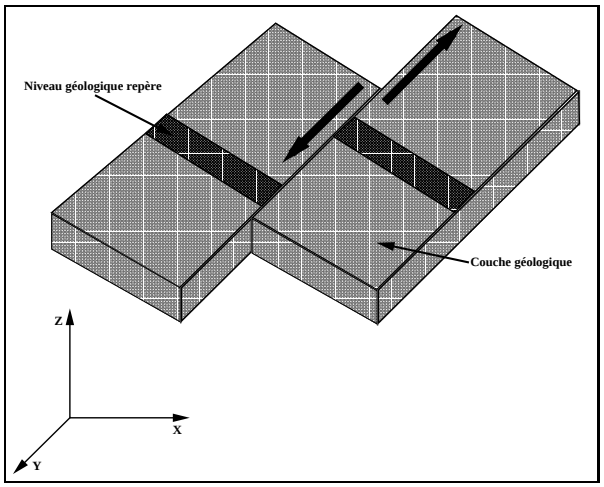

FIG. 3.5 – *Exemple d'une faille en décrochement*

3.3 Conclusion

Les géologues ont remarqué que la plupart des gisements pétroliers ou miniers ont été le siège de déformations géologiques très compliquées, syn ou post formation. Ces déformations se manifestent parfois par des failles. Comme le montre l'exemple de la figure 3.6, un gisement peut se trouver brutalement décalé du fait de la présence d'une ou plusieurs failles. La faille peut aussi jouer un rôle important dans la formation d'un gisement, par le contrôle de la circulation des fluides à l'origine des minéralisations, ainsi une faille peut jouer le rôle d'un filtre spécial qui ne laisse passer que les molécules d'eau ; la précipitations des éléments en solution provoque la formation d'un gisement.

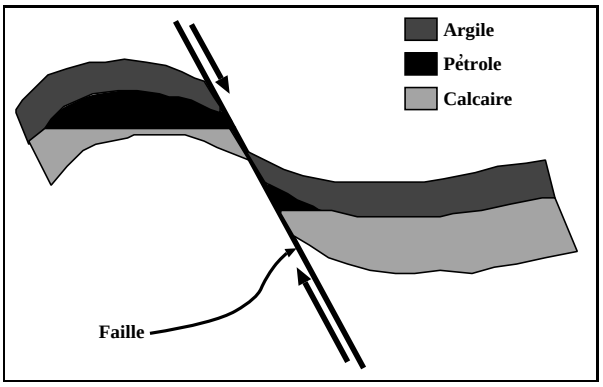

FIG. 3.6 – *Exemple d'un réservoir faillé*

Les failles peuvent aussi fournir des renseignements précieux qui permettent aux géologues de reconstruire l'histoire géologique des couches affectées par ces failles.

Nul doute, qu'un système informatique aussi performant soit-il, ne peut pas modéliser correctement des surfaces géologiques s'il n'arrive pas à prendre en compte la notion de faille. Or cette notion n'est pas correctement prise en compte par les systèmes informatiques classiques ([36]), basés sur des surfaces de type Bézier ou spline.

Je désire insister sur le fait qu'il ne s'agit pas d'une critique des méthodes de $C.A.O.$ classiques qui sont très efficaces pour la modélisation des surfaces

manufacturables, mais ces méthodes ne sont pas très adaptées aux surfaces naturelles.
Fort de son approche destinée à la modélisation de surfaces naturelles, le système GOCAD, grâce à la souplesse de ses structures de données et à la robustesse de la méthode d'interpolation DSI, nous a permis de développer la notion de faille.

Chapitre 4

Modélisation des failles

Si les failles et les horizons[1] sont géologiquement de nature très différents, géométriquement, ils peuvent être représentés par des surfaces. La différence réside dans leur comportement respectif. A l'image de notre société, le sous-sol apparaît comme un univers bien organisé, où tous les individus sont géométriquement identiques, cependant leurs propriétés physiques permettent de définir leurs relations avec les autres individus. Ainsi, une faille se distinguera d'un horizon par ses propriétés physiques qui lui permettent, par exemple, de couper tous les horizons qui lui sont chronologiquement antérieurs.
La modélisation des failles se fera donc en trois étapes :

- Modélisation géométrique. Cette partie est commune à toutes les surfaces géologiques.

- Modélisation des propriétés physiques associées aux failles. Ces propriétés vont permettre aux failles de définir leurs relations avec les horizons qu'elles coupent.

- Modélisation géométrique de la relation faille-horizon. Il s'agit d'une des relations les plus importantes et les plus compliquées, dans le do-

1. Un horizon est une surface qui correspond au toit ou à la base d'une couche géologique

maine de l'exploration pétrolière, car cette relation conditionne le comportement de l'ensemble des horizons coupés par la faille.

4.1 Définition géométrique des failles

Une faille est une surface triangulée (cf chapitre 1) qui possède les mêmes caractéristiques géométriques qu'un horizon géologique. En pratique, les failles sont connues au travers d'un échantillonnage de données variées provenant de campagnes sismiques, de puits, de relevés de terrain, $\cdots$. Il s'agit en résumé, de données hétérogènes irrégulièrement disposées dans l'espace et parfois imprécises.

La modélisation de la géométrie des failles consiste à développer un système performant permettant de les construire, quel que soit le type de données dont on dispose. Malheureusement, il n'existe aucun procédé universel de création automatique d'une surface à partir de données quelconques. L'approche choisie consiste à associer à chaque type de données une technique de construction de surface adaptée à ce type de données, tout en permettant à l'utilisateur d'intervenir pendant toutes les étapes de la construction. En effet, plusieurs techniques de construction de surfaces ont été développées ([6][49] $\cdots$), mais ces méthodes s'adaptent difficilement à la modélisation des surfaces naturelles.
Dans le cadre du projet GOCAD plusieurs techniques de création de surfaces, spécialement conçues pour les surfaces naturelles, ont été développées. Pour ne pas surcharger le mémoire, on n'en présentera que deux.

4.1.1 Construction d'une surface à partir d'un semis de points

Cette technique est adaptée aux données provenant de campagnes sismiques $3D$. Comme le montre la figure 4.1, une campagne sismique consiste à émettre des ondes à partir d'un point appelé *émetteur*. Après réflexion sur les horizons géologiques, ces ondes seront captées par les *récepteurs*. Des calculs complexes ([25]), réalisés à partir des signaux reçus par les récepteurs, permettent de déterminer les coordonnées des points de réflexion, la position des failles, des dômes de sel probables, des anciennes surfaces d'érosion, $\cdots$. Ainsi, à chaque surface géologique S correspondra un semis de points V échantillonnant cette surface.
Une méthode de construction d'une surface géologique quelconque, à partir

d'un semis de points, a été proposée par Y. Chipot et P. Lavest ([31][10]), cette méthode consiste en :

1. La construction d'une surface triangulée H (cf figure 4.2b), qui correspond au plan moyen du semis de points de la figure 4.2a.

2. L'ajustement de la surface H au semis de points en utilisant la contrainte FCP décrite dans le chapitre 2. La figure 4.2c montre le résultat obtenu. Un contrôle qualitatif du résultat obtenu peut être fait, en calculant l'erreur d'ajustement de la surface finale au semis de points (voir chapitre 2).

Cette méthode est parfaitement adaptée à ce type de données, car d'une part elle permet un ajustement rigoureux et rapide de la surface aux données, et d'autre part, même si on modifie les données initiales (par ajout ou retrait de points), le système de superviseur de projet se charge de mettre à jour les contraintes. Le principe du superviseur de projet a été décrit dans le chapitre 2)

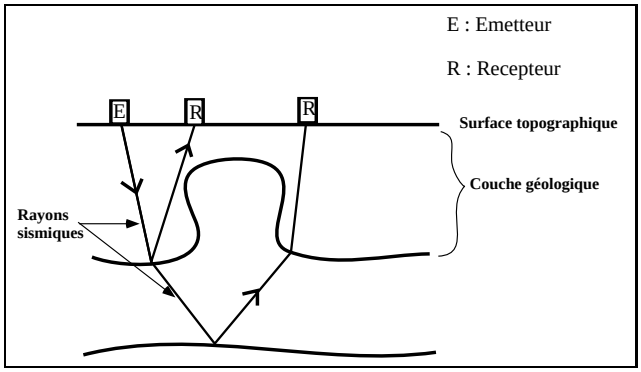

FIG. 4.1 – *Exemple d'une campagne sismique (coupe verticale)*

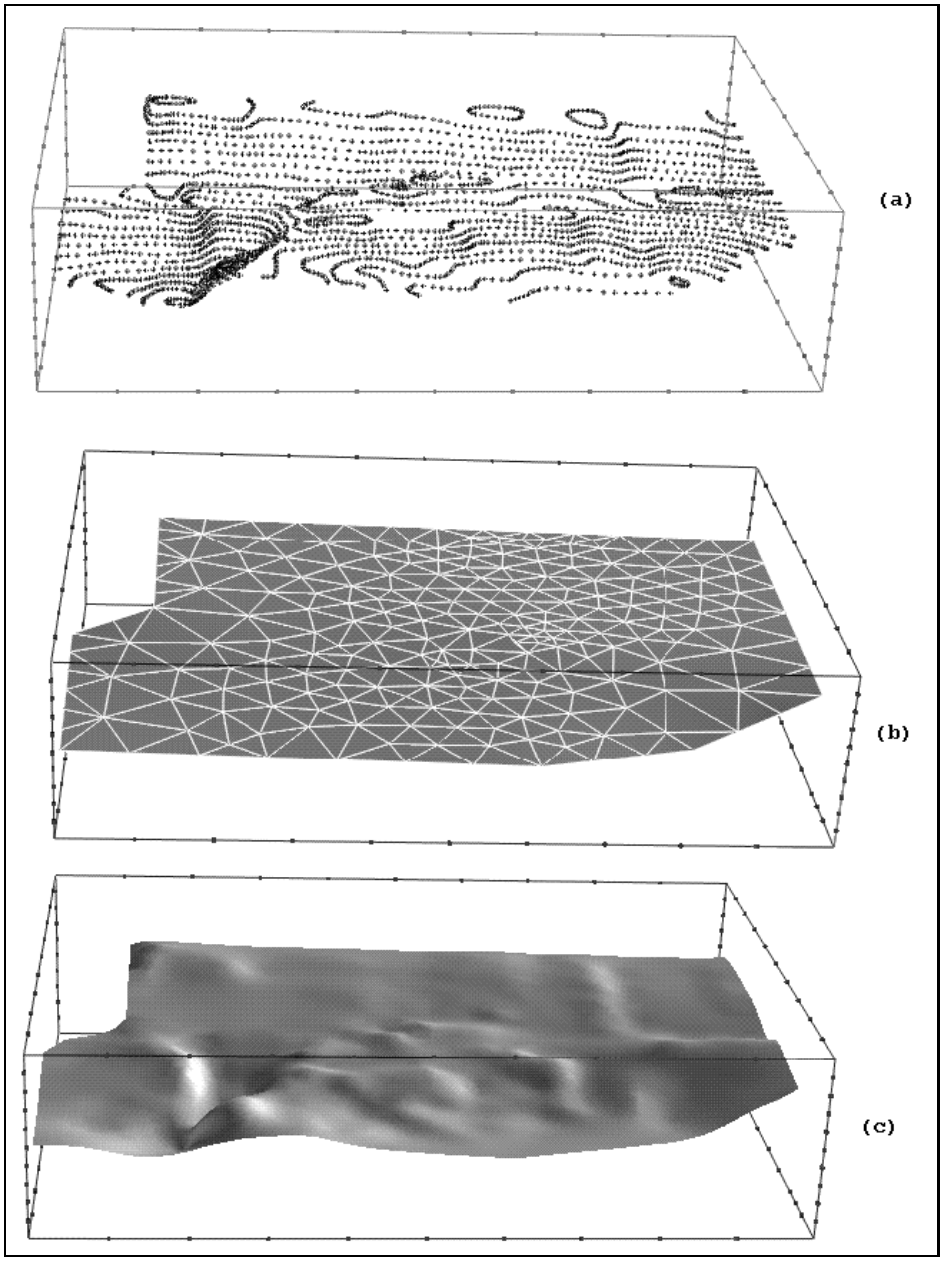

FIG. 4.2 – *Génération d'une surface à partir d'un semis de points: (a) Semis de points, (b) Création du plan moyen du semis, (c) Ajustement de la surface aux données initiales*

– **Remarque**
Pour des surfaces complexes (cf figure 4.3), l'intervention de l'utilisateur peut être nécessaire pour obtenir un meilleur résultat.

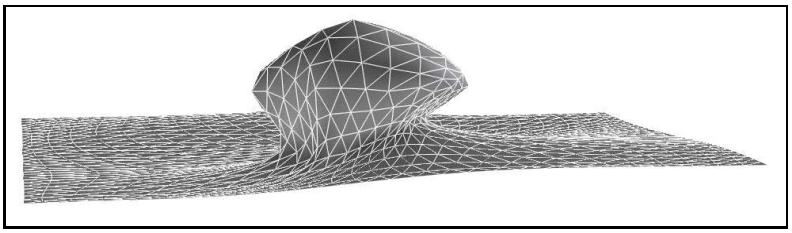

FIG. 4.3 – *Exemple d'un dôme de sel*

4.1.2 Construction d'une surface à partir de lignes

Cette technique est adaptée aux données provenant de coupes sismiques. Sur chaque section sismique plane, on digitalise une ligne polygonale L_i^j qui correspond à un horizon géologique H_i. Par exemple, sur la section sismique de la figure 4.4, on a digitalisé deux lignes polygonales A et B qui correspondent aux traces de deux failles. La méthode choisie ([44]) dans ce cas consiste à créer une surface triangulée qui passe par toutes les lignes polygonales L_i^j correspondant au même horizon géologique H_i.
Un exemple est donné par la figure 4.5. Il montre, que pour construire une surface H_i à partir d'un ensemble de lignes L_i^j (Fig 4.5a), on procédera en deux étapes :

1. à partir de deux lignes polygonales adjacentes L_i^j et L_i^{j+1}, et comme le montre la figure 4.5b, on construit une surface ($S_i^{(j,j+1)}$) ([44])

2. on associe ensuite toutes les surfaces $S_i^{(j,j+1)}$ pour obtenir l'horizon H_i (Fig 4.5c).

Les résultats obtenus par cette technique sont intéressants. Cependant, pour des courbes de géométries très différentes, la méthode décrite précédemment ne peut être utilisée automatiquement. De nouvelles procédures sont à développer pour remédier à ce problème ([45]).

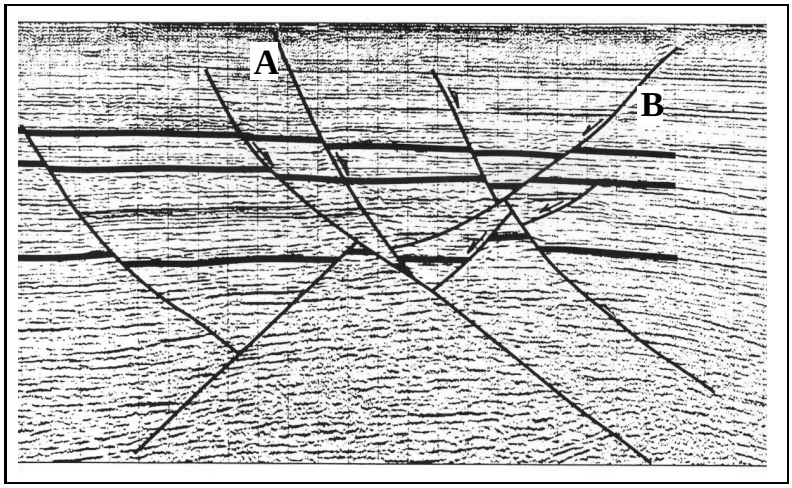

FIG. 4.4 – *Exemple d'une section sismique*

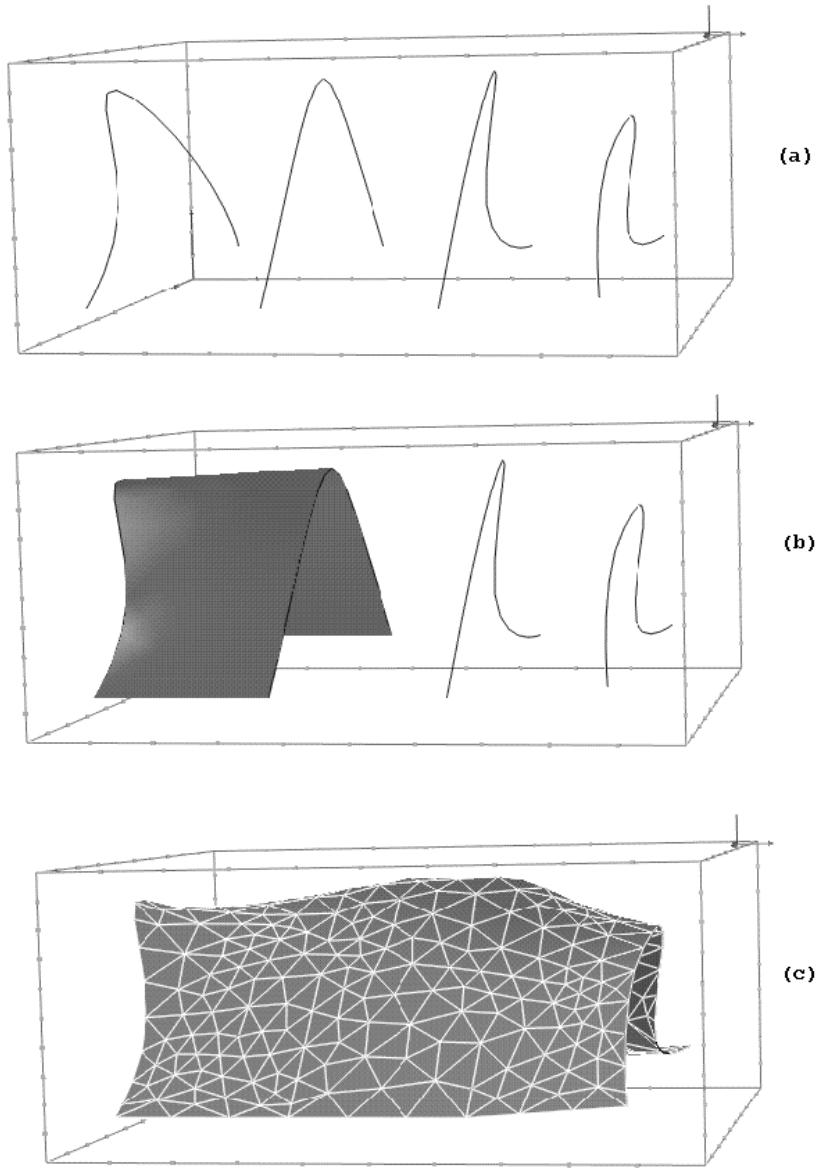

FIG. 4.5 – *Génération d'une surface à partir de lignes*

4.2 Modélisation des propriétés physiques d'une faille

Les failles présentent des propriétés physiques qui permettent de les distinguer des autres surfaces géologiques. On distingue par exemple :

– *Le rejet de la faille*
Je rappelle que le rejet d'une faille est le vecteur déplacement que la faille engendre sur ses deux compartiments (voir chapitre 3). Des études géologiques ([55] [56] [57] [50][4]) ont montré que le rejet d'une faille est une propriété physique qui peut varier verticalement et latéralement. La figure 4.6 montre la cartographie des rejets d'une faille. Sur le plan de cette dernière, on remarque que ce rejet passe d'une valeur minimale vers les bords de la faille, à une valeur maximale au centre. On explique cela par la variation des caractéristiques mécaniques des couches faillées ([34][27]), et par la dissipation de l'énergie associée aux phénomènes tectoniques à l'origine de la faille.

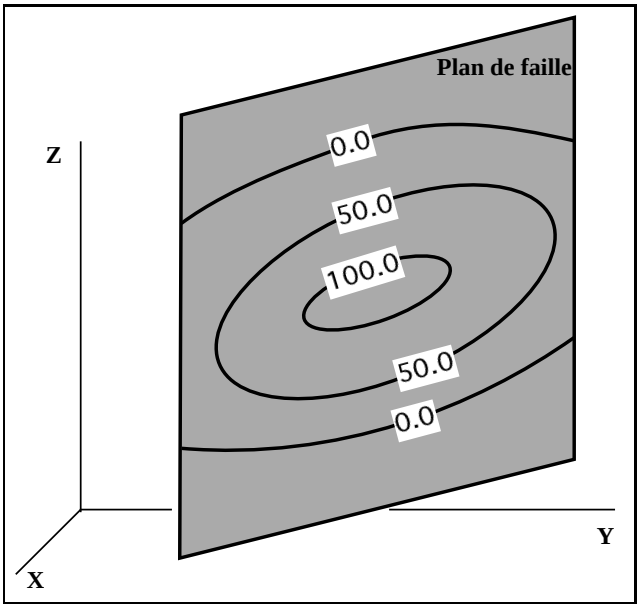

FIG. 4.6 – *Cartographie de l'amplitude du rejet d'une faille sur son plan*

– *La perméabilité de la faille*
Les pétroliers et les mineurs considèrent la perméabilité, comme une propriété importante des failles ([27] [26] [59]), car elle permet d'étudier

le comportement d'un gisement brusquement coupé par une faille. Un exemple est donné par un des gisements pétroliers du golfe du Mexique ([58]). Dans ce dernier, et comme le montre la coupe sismique de la figure 4.7, les couches géologiques, pouvant servir de réservoir au pétrole, sont tabulaires (horizontales) et donc faciles à étudier. Cependant, la présence d'un nombre important de failles, représentées par les lignes épaisses sur la figure 4.7, affectant l'ensemble géologique, rend difficile l'étude et l'exploitation des réservoirs pétroliers dans le golfe du Mexique. De plus, le comportement du pétrole au voisinage d'une faille suit des lois physiques complexes, d'où l'intérêt de pouvoir étudier, avec un maximum de précision, la variation de la perméabilité le long d'une faille.

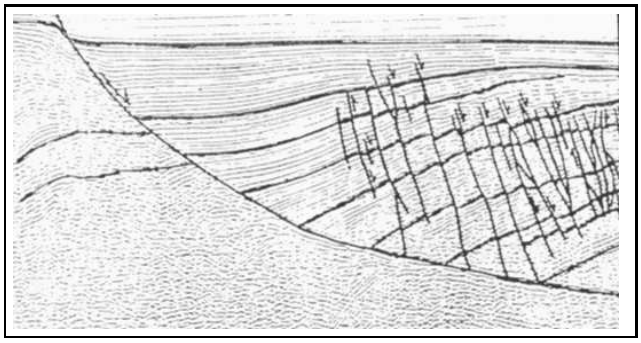

FIG. 4.7 – *Section sismique interprétée du golfe du Mexique, D'après Wernick et Burchfiel 1982*

En pratique, les propriétés physiques des failles sont données par un échantillonnage de points, irrégulièrement disposés dans l'espace $3D$, au niveau desquels les valeurs des propriétés sont connues à un coefficient de précision donné près. Le but est d'interpoler ces propriétés sur toute la surface de la faille. Des études faites par R. Cognot ([14]) ont montré que parmi les nombreuses méthodes d'interpolation susceptibles de résoudre ce problème, la méthode DSI est la mieux adaptée à ce type de données, à savoir des données ponctuelles, irrégulièrement disposées dans l'espace, et possédant un degré de précision variable.

Un exemple est donné par la figure 4.8. Il consiste à interpoler une propriété physique quelconque sur une surface triangulée. Les données sont représentées par un échantillonnage de points $\{P_0, P_1, \cdots, P_n\}$. Au niveau de

chaque point P_i, la propriété physique vaut ϕ_i. La technique utilisée est illustrée par la figure 4.8, elle consiste à projeter chaque point P_i sur la surface S. Supposons que P_i, au niveau duquel la propriété physique vaut ϕ_i, intersecte un triangle $T(\varphi_0, \varphi_1, \varphi_2)$ de la surface S en un point I_i, la valeur de la propriété physique au niveau de chaque nœud φ_i du triangle T est donnée par une fonction $\psi(\varphi_i, P_i, I_i)$ ([14]). Après avoir calculé la valeur de la propriété physique au niveau des sommets des triangles intersectés, la méthode DSI se charge ensuite d'interpoler ces propriétés sur toute la surface. Ces propriétés peuvent ensuite être visualisées sur la surface à l'aide de courbes d'isovaleur.

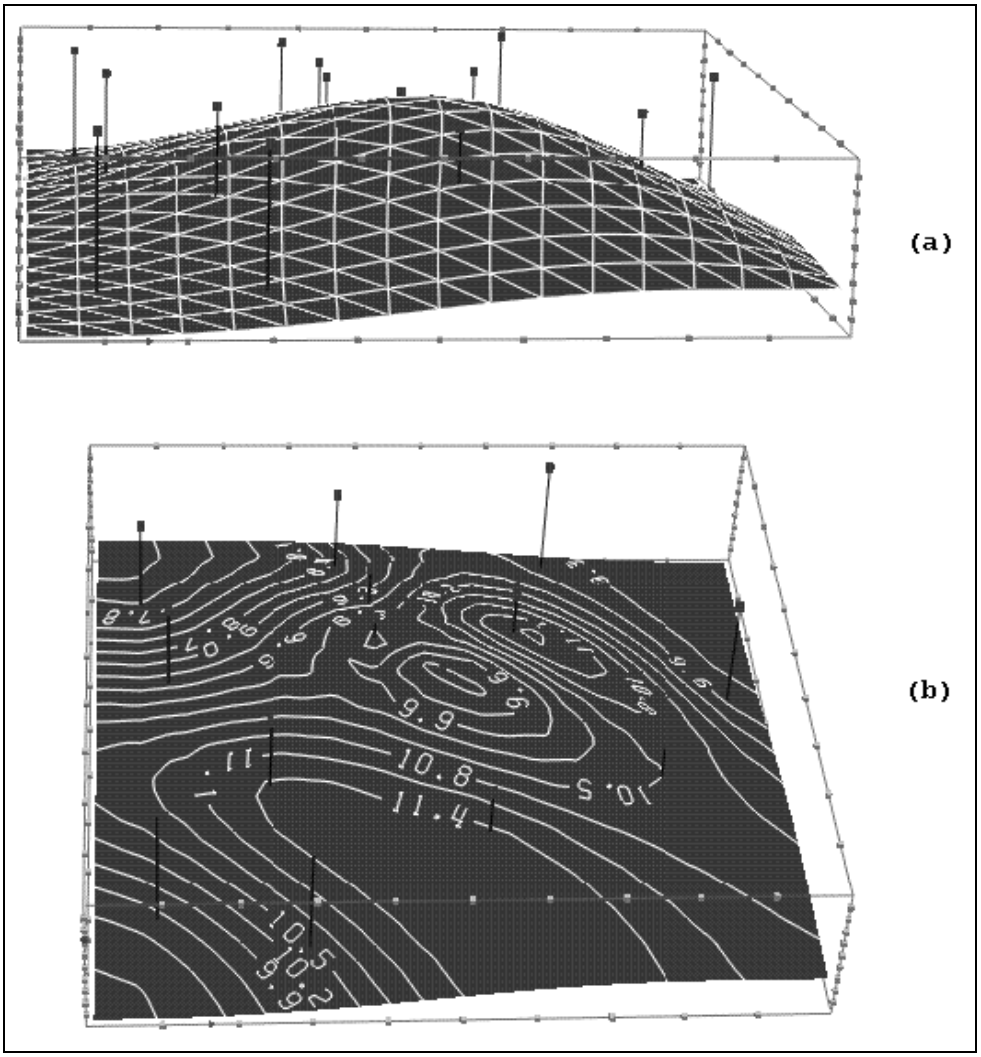

FIG. 4.8 – *Interpolation des propriétés sur une surface triangulée. a) Projection des points de données sur une surface triangulée, b) Les courbes de niveau correspondent aux valeurs de la propriété après interpolation par DSI*

4.3 Relation faille-horizon

La présence d'une faille au sein d'un ensemble géologique augmente considérablement la complexité d'un modèle, car elle conditionne le comportement des horizons intersectés. La figure 4.9, montre un exemple où l'horizon glisse le long de la faille, sans pour autant qu'il puisse s'en décoller. De plus, la faille contrôle le déplacement qu'elle engendre au niveau des bords de l'horizon coupé.

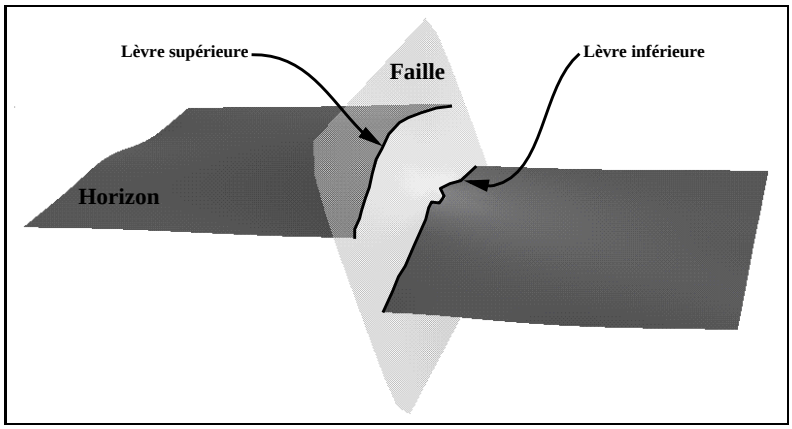

FIG. 4.9 – *La relation faille-horizon*

La relation (faille, horizon) ne peut donc être correctement définie que si on peut contrôler à la fois le contact *faille-horizon* et l'ampleur du déplacement des lèvres de la faille. La méthode DSI présentée précédemment, est parfaitement adaptée à ce type de problème ; elle permet de tenir compte de ces conditions au moyen de contraintes géométriques ([1][2]) qui seront appliquées sur les lèvres de la faille afin d'assurer le contact *horizon-faille*, et de contrôler le déplacement des lèvres de la faille. Ce contact est en fait modélisé par les deux contraintes principales suivantes :

1. *La contrainte On-Tsurf* (cf figure 4.11)
   Cette contrainte permet de coller les bords d'un horizon à la faille. Il s'agit dans ce cas d'un contact que l'on qualifie de *souple*, car les bords de l'horizon peuvent se déplacer le long de la faille.

2. *La contrainte Veclink* (cf figure 4.17)
   Elle permet de contrôler le rejet d'une faille, qui se matérialise par le déplacement relatif des lèvres de la faille.

Géologiquement, si une faille est interne à un horizon (cf figure 4.10), les extrémités des lèvres de la faille doivent obligatoirement glisser le long du bord de la faille. Malheureusement, la contrainte $OnTsurf$ ne peut pas obliger un nœud des lèvres de la faille à se déplacer le long d'une ligne polygonale correspondant au bord de la faille. Une troisième contrainte géométrique supplémentaire, appelée ($OnBorder$), a donc été mise en place. Cette contrainte sera traitée dans le paragraphe 4.3.3.

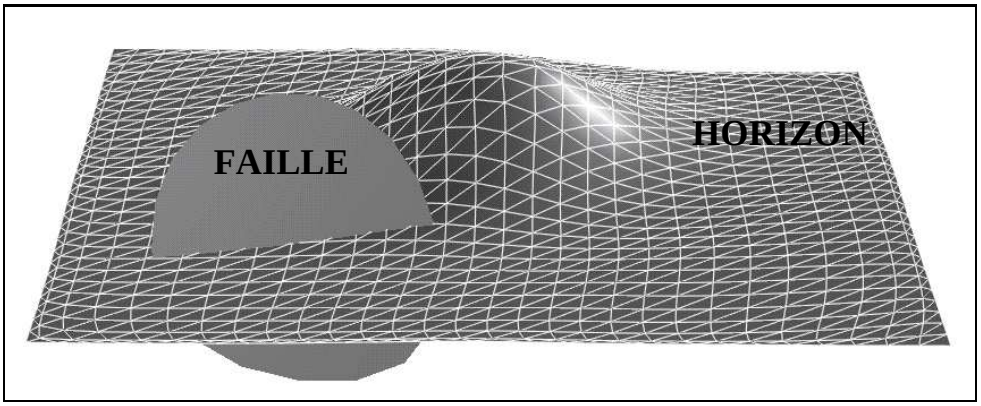

FIG. 4.10 – *Exemple d'une faille interne à un horizon*

4.3.1 La contrainte On-Tsurf

Cette contrainte a été développée afin d'obliger le bord d'une surface à se déplacer le long d'une autre surface. Les tentatives de modélisation de cette contrainte, à l'aide des méthodes de $C.A.O.$ classiques n'ont pas donné de résultats satisfaisants ([36][43]), une nouvelle approche a donc été introduite.

4.3.1.1 L'originalité du principe adopté

Le principe adopté, et illustré par la figure 4.11, a été initialement proposé par P. Le Mélinaire ([36]), nous l'avons ensuite modifié pour prendre en compte la notion *Atomique-Contrôleur* présentée dans le chapitre 2. La contrainte On-Tsurf consiste à projeter chaque nœud des lèvres de la faille sur la surface de la faille. Chaque nœud possède une direction de tir $\overrightarrow{Tir}$, et une cible associée. Par exemple sur la figure 4.11, la droite issue du nœud A, intersecte le triangle T de la surface F au point d'impact I. Si le nœud A est

déplacé vers le point I, le contact entre les deux surfaces sera réalisé.

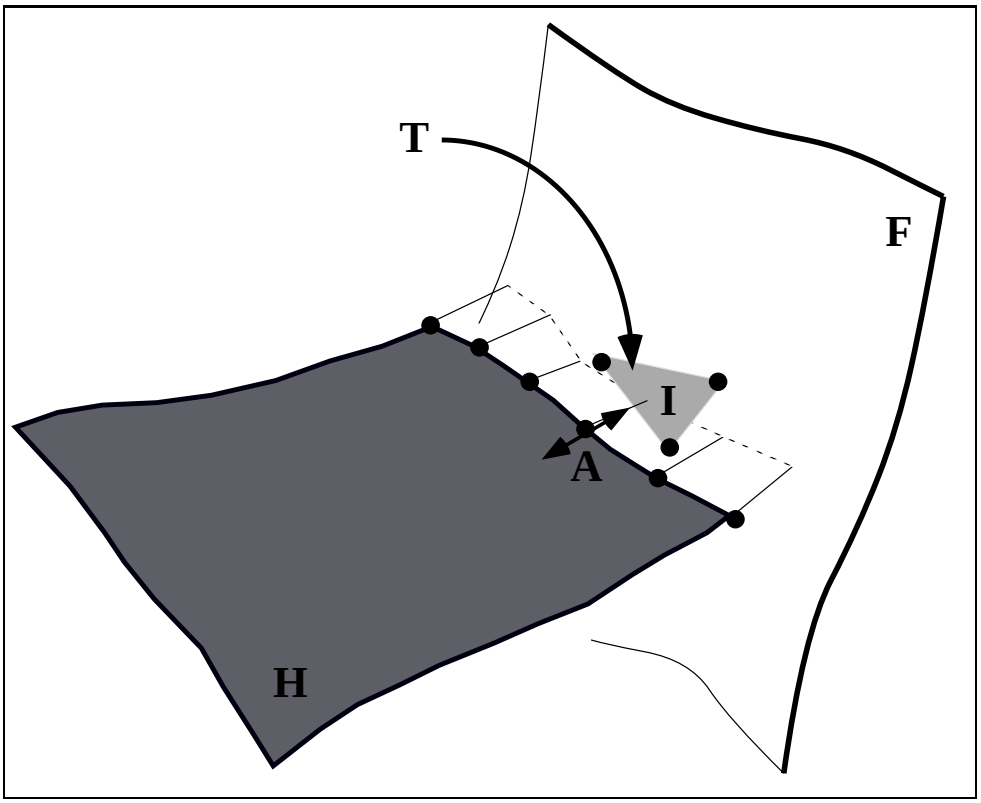

FIG. 4.11 – *Installation des contraintes On-Tsurf sur une lèvre d'une faille*

Ce type de contact doit être automatiquement recalculé chaque fois qu'une des deux surfaces a été modifiée. Cela est possible grâce au système de *superviseur de projets* présenté dans le chapitre 2.

Parmi les problèmes rencontrés lors de l'implémentation de cette contrainte on peut citer :

- **Définition de la direction de tir**
  La direction de tir est fonction de la forme et de la position relative des deux surfaces. Cependant, on peut automatiser son initialisation par la méthode suivante:
  Considérons la figure 4.11, grâce à une méthode de recherche de proximité, on calcule la position de **I**, le point de la faille le plus proche de

A. Si $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ est le triangle de la faille auquel appartient le point **I**, le vecteur de tir $\overrightarrow{Tir}$ est égal au vecteur d'origine **A** et d'extrémité **I**, et le triangle intercepté est le triangle T. Le sens du vecteur $\overrightarrow{Tir}$ a peu d'importance, car la procédure $TSURF\_Shoot()$ (cf chapitre 2), qui calcule les triangles interceptés, effectue la recherche dans les deux directions opposées portées par ce vecteur. Cette direction de tir devra être recalculée chaque fois que la surface sera modifiée, car la position du point **I** dépend de la position du nœud **A** et de la forme géométrique de la faille.

- **Calcul de la projection**
  La première façon de calculer la projection d'un nœud consiste à utiliser la procédure $TSURF\_Shoot()$. Cette fonction détermine, sur l'ensemble de la surface, le triangle intercepté et le point d'intersection. Cependant, cette fonction n'est pas assez performante car si nous supposons que n est le nombre de triangles de la surface cible, alors cette fonction est en $O(\mathrm{n})$. Une seconde approche a donc été envisagée. Elle est basée sur la valeur des coordonnées barycentriques du point d'impact dans le triangle intercepté. Ceci permet de suivre la trace du point d'impact lorsqu'il se déplace d'un triangle à un autre ([36]). Le principe de cette méthode a été présenté dans le chapitre 2, lors de l'étude de la contrainte Fuzzy-Control-Point.

- **Prise en compte de cette contrainte dans l'équation de DSI**
  Considérons l'exemple de la figure 4.12 :

  - $\mathbf{P}_i$ est un nœud du bord de la surface $horizon$,
  - $\overrightarrow{D}$ est la direction de projection de $\mathbf{P}_i$ sur la surface $faille$,
  - $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ est le triangle intersecté lors de la procédure de projection.
  - $\mathbf{I}_i$ est le point de projection de $\mathbf{P}_i$ le long de la direction $\overrightarrow{D}$ sur le plan supportant le triangle $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$.

  Dans la suite on suppose que Le vecteur $\overrightarrow{OQ}$ qui joint l'origine des abscisses au point **Q** ainsi que la matrice colonne contenant les coordonnées de ce vecteur seront notés **Q**.
  La normale $\overrightarrow{N}$ au triangle T peut être calculée selon la formule suivante, sachant que $A \times B$ est le produit vectoriel du vecteur A par le vecteur B :

$$\overrightarrow{N} = \overrightarrow{(\mathbf{p}_1 - \mathbf{p}_0)} \times \overrightarrow{(\mathbf{p}_2 - \mathbf{p}_0)}$$

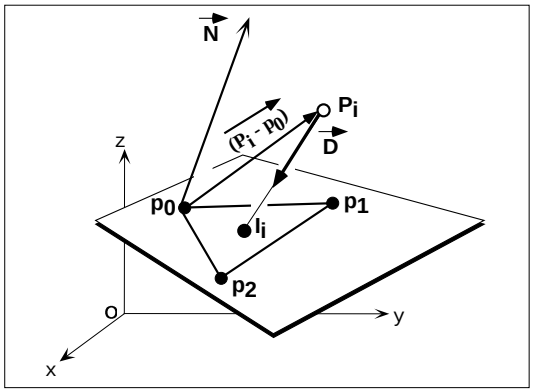

FIG. 4.12 – *Principe géométrique de la contrainte OTS*

Le point $\mathbf{P}_i$ appartient au triangle $T(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$ si et seulement si la relation suivante est vérifiée, sachant que $A \cdot B$ est le produit scalaire des deux vecteurs A et B :

$$\overrightarrow{N} \cdot \overrightarrow{(\mathbf{P}_i - \mathbf{p}_0)} = 0 \tag{4.1}$$

La relation précédente peut s'écrire de la manière suivante :

$$\overrightarrow{N} \cdot \mathbf{P}_i = \overrightarrow{N} \cdot \mathbf{p}_0$$

Si on pose :

$$\begin{array}{rcl} A^\nu & = & N^\nu \\ b & = & \overrightarrow{N} \cdot \mathbf{p}_0 \end{array}$$

On peut alors écrire :

$$\forall \nu = (x, y, z) : A^\nu \cdot \mathbf{P}_i^\nu = b$$

Il s'agit de la formulation d'une contrainte floue dans l'équation de DSI (cf chapitre 2). Si la contrainte OTS est la $i^{\text{ème}}$ contrainte au niveau du nœud P_i, et si on affecte à cette contrainte le poids ϖ_i^2, les coefficients $\Gamma_i^\nu(\alpha)$ et γ_i^ν, permettant de prendre en compte cette contrainte dans l'équation de DSI, se calculent comme suit :

$$\forall \nu = (x, y, z) : \left\{ \begin{array}{rcl} \Gamma_i^\nu & = & \varpi_i^2 \cdot A^\nu N^\nu \cdot \{ \sum_{\substack{\lambda = (x,y,x) \\ \lambda \neq \nu}} A^\lambda \mathbf{P}_i^\lambda - b \} \\ \gamma_i^\nu & = & \varpi_i^2 \cdot (N^\nu)^2 \end{array} \right. \tag{4.2}$$

La contrainte OTS sera installée au niveau de chaque nœud $\mathbf{P}_i$ des lèvres de la faille. A chaque itération de DSI le nœud $\mathbf{P}_i$ se déplacera vers la faille de manière à vérifier la relation 4.1. Cependant dans le cas de la contrainte OTS le contact entre le bord de l'horizon et la faille doit être parfait, nous avons donc décidé de considérer la contrainte OTS comme étant une contrainte dure (cf chapitre 2).
Reprenons la figure 4.11, la contrainte **dure** consiste à déplacer le nœud **A** vers le point d'impact **I**.
La prise en compte de cette contrainte par l'interpolateur DSI sera donc faite de deux manières :

1. de manière **floue** en calculant les coefficients Γ_i^ν et γ_i^ν,
2. de manière **dure** en déplaçant chaque point $\mathbf{P}_i$ du bord de l'horizon vers le point d'impact $\mathbf{I}_i$ correspondant sur la faille (cf figure 4.12).

La contrainte OTS est une contrainte projetée, donc les classes OTS_INFO et OTS, correspondant à cette contrainte, dérivent respectivement de PRJ_INFO et PRJ (cf chapitre 2). Les fonctions, membres virtuelles des classes PRJ_INFO et PRJ, seront définies au niveau des classes OTS_INFO et OTS. Parmi ces fonctions nous pouvons citer :

- les fonctions de création et de destruction des instances de la classe OTS,
- la fonction *fuzzyCnstr()*, permettant à DSI de prendre en compte une instance de la classe OTS de manière floue (cf chapitre 2). Cette fonction consiste à calculer les coefficients Γ_i et γ_i au niveau de chaque atome des lèvres de la faille. Ces coefficients seront calculés selon la formule 4.2, il seront utilisés par DSI pour respecter la contrainte OTS,
- la fonction *hardCnstr()*, permettant à DSI de prendre en compte une instance de la classe OTS comme une contrainte dure. Cela permet d'avoir un contact parfait entre le bord de l'horizon et la faille,
- les fonctions de mise à jour des contraintes si un des deux objets, horizon ou la faille, a été modifié.

4.3.1.2 Résultats obtenus

Les résultats présentés correspondent à trois séries d'événements qui illustrent l'ajustement du contact entre une faille (représentée par la surface ver-

ticale), et un horizon.

- *Initialisation de la ligne de soudure*
  La figure 4.13 représente une ligne de soudure entre un horizon (en jaune) et une faille (en bleu). Chaque nœud de la lèvre de la faille intercepte la faille en un point d'impact. Les autres nœuds du bord de l'horizon sont autorisés à se déplacer dans le sens vertical, pour maintenir la tension sur l'horizon. Après quelques itérations de l'interpolateur DSI, le contact entre l'horizon et la faille se trouve réalisé (cf figure 4.14).

- *Translation de la surface*
  Sur la figure 4.15, la faille a été déplacée pour que les deux surfaces s'interpénètrent. Le résultat, après interpolation, montre que le contact a été restitué (cf figure 4.16).

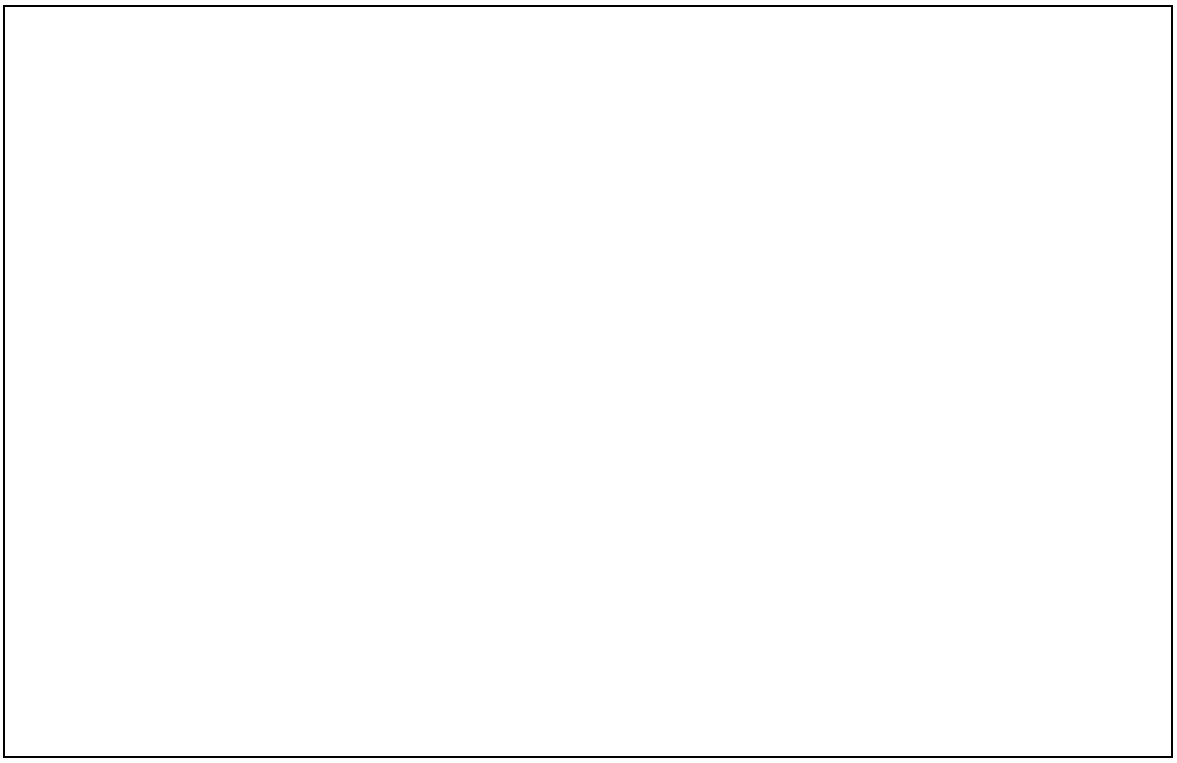

FIG. 4.13 – *Installation de la contrainte On-Tsurf sur une lèvre de la faille, les lignes blanches correspondent aux directions de projections des nœuds du bord de l'horizon*

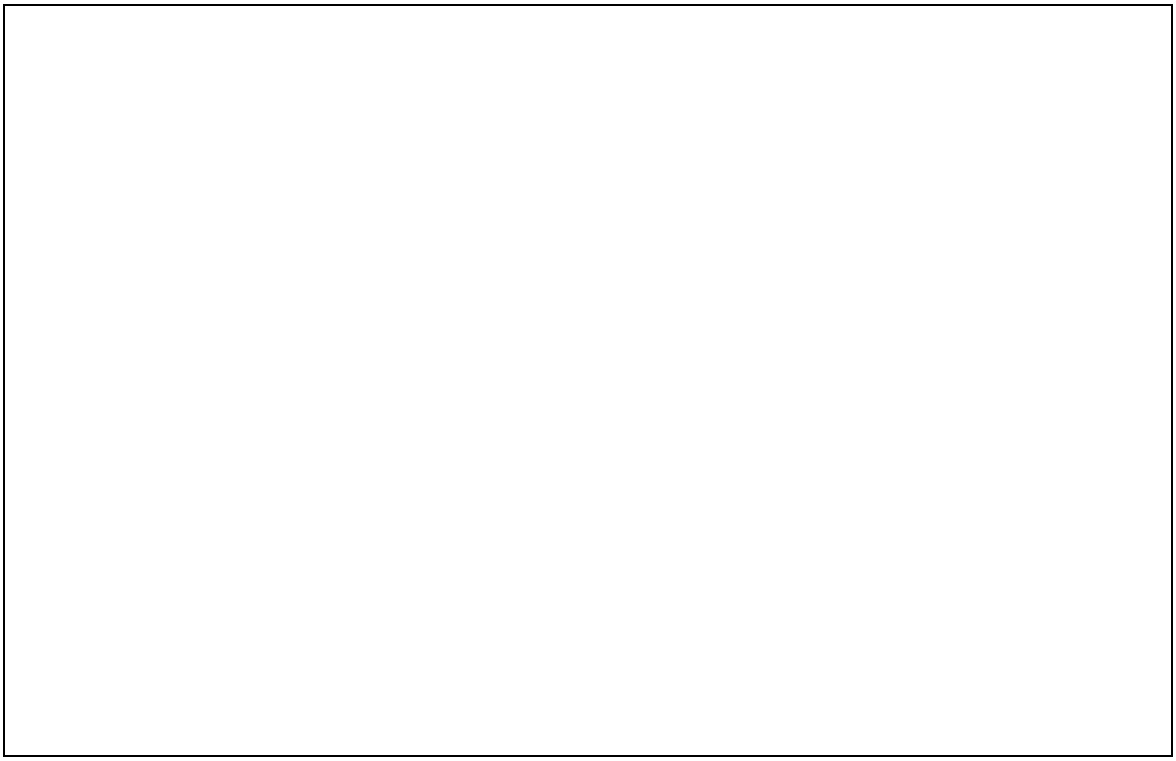

FIG. 4.14 – *Restitution du contact entre l'horizon et la faille*

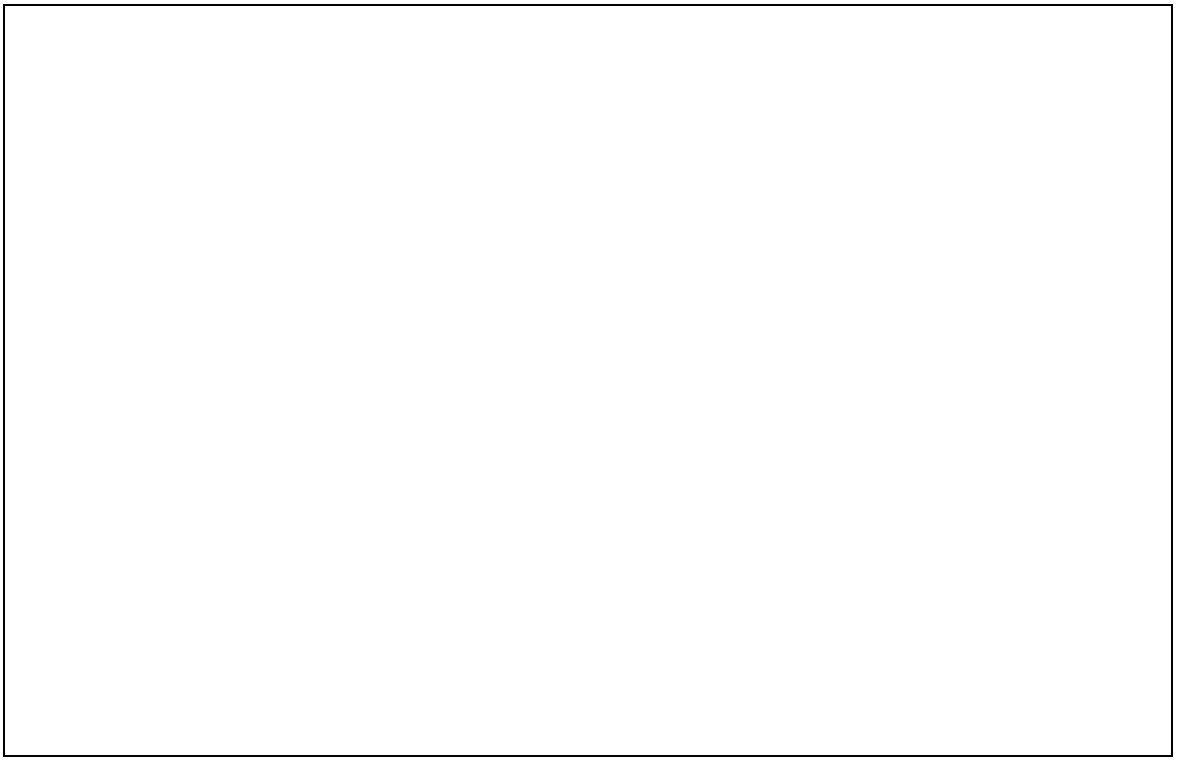

FIG. 4.15 – *Translation de la faille*

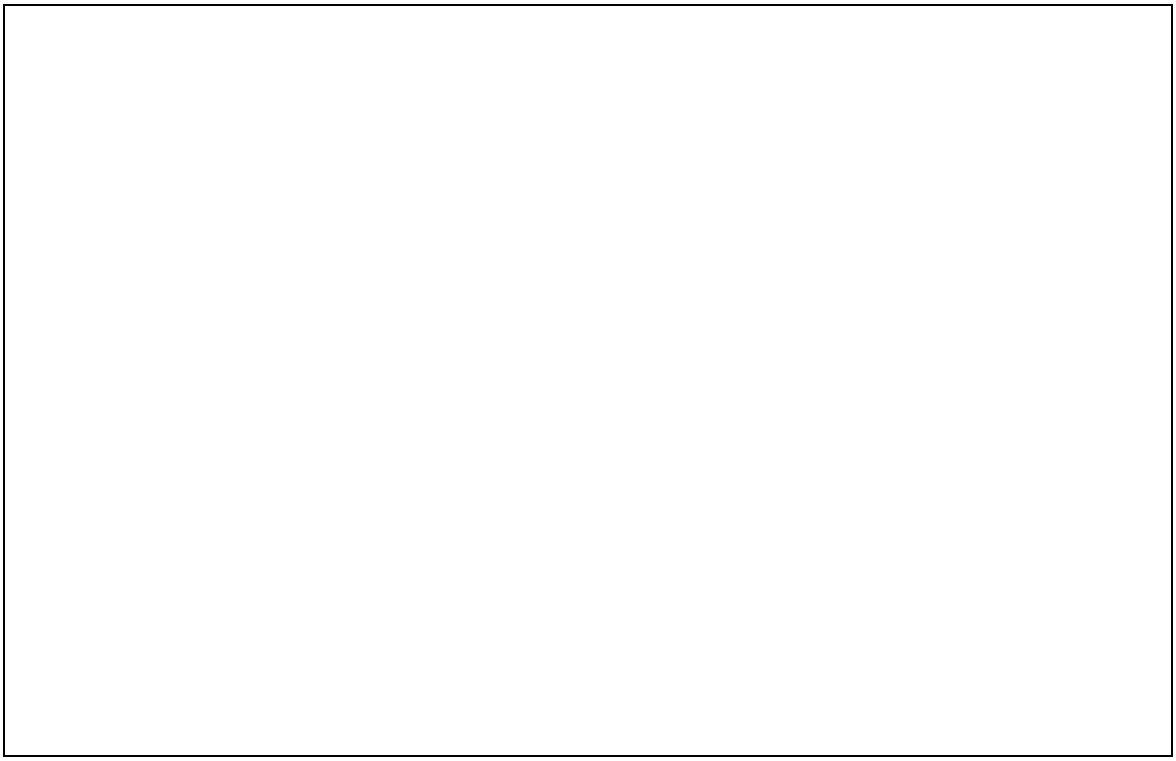

FIG. 4.16 – *Restitution du contact entre l'horizon et la faille*

4.3.2 La contrainte VecLink

La contrainte VLK[2] permet de contrôler l'ampleur du rejet d'une faille, en contrôlant le déplacement des lèvres de la faille. Ce problème a été évoqué depuis les années 1980 ([55] [56] $\cdots$), cependant les méthodes proposées pour le résoudre n'ont pas donné de résultats satisfaisants car elles supposent que les failles sont des surfaces planes verticales. Nous avons donc conçu une nouvelle méthode générale qui fera l'objet de ce paragraphe.

4.3.2.1 Principe adopté

L'idée consiste à subdiviser, comme le montre la figure 4.17, le bord de l'horizon en deux parties (chaque partie est associée à une lèvre de la faille), qui correspondent à deux courbes paramétriques $\{P(s) : s \in [0,1]\}$ et $\{P'(s) : s \in [0,1]\}$. Un vecteur rejet $T(s)$ sera défini par :

$$\overrightarrow{T(s)} = \overrightarrow{P(s)P'(s)} \; \forall s \in [0,1]$$

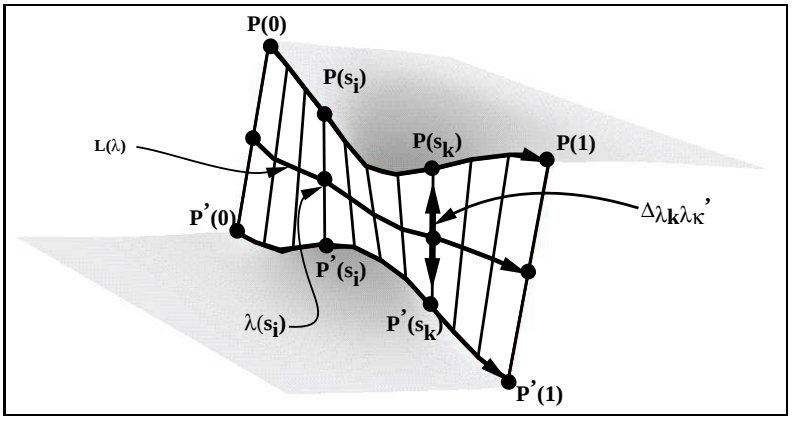

FIG. 4.17 – *Orientation des lèvres d'une faille*

Sur chacune des deux courbes, on échantillonne un ensemble de $(m+1)$ points équidistants $\{P(s_i) : i \in [0,m]$ et $\{P'(s_i) : i \in [0,m]\}$ avec $s_i = \frac{i}{m}$ (cf figure 4.18). Nous pouvons ainsi créer, comme le montre la figure 4.17, une courbe L dont les nœuds $\{\lambda_i : i \in [0,m]\}$ sont définis par la formule suivante:

$$\lambda_i = \frac{P(s_i) + P'(s_i)}{2}$$

2. abréviation de VecLink

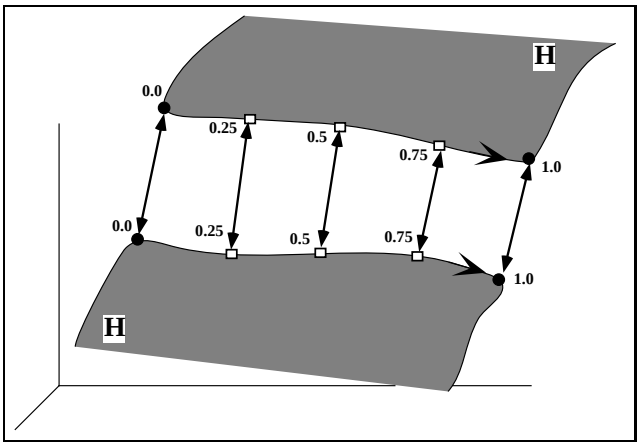

FIG. 4.18 – *Echantillonnage des points sur les lèvres d'une faille*

Remarque :

Il n'existe pas de relation entre la disposition des points d'échantillonnage des deux courbes paramétriques $P(s)$ et $P'(s)$ et la disposition des nœuds des lèvres de la faille, cela permet de prendre en compte des lèvres de failles de forme géométrique complexe.

A chaque nœud λ_i de la courbe L, est associé un vecteur rejet *courant* $T(s_i)$. Nous disposons aussi au même nœud λ_i, d'un vecteur rejet *réel* $\Delta_{\lambda_i\lambda'_i}$, qui correspond à la vraie valeur du rejet de la faille à cette position.

Si, en plus, nous connaîssons avec précision la valeur de ce rejet réel au niveau de quelques vecteurs, il est possible, comme le montre la figure 4.26, d'estimer la valeur du rejet réel au niveau des autres noeuds de la courbe L.

Le but de la contrainte $VecLink$ est d'assurer que le rejet *courant* soit égal au rejet *réel*. Cela se matérialise par la relation suivante :

$$T(s_i) = \Delta_{\lambda_i\lambda'_i} \tag{4.3}$$

Dans la suite de l'exposé, nous désignerons par **lien** le vecteur **rejet courant**.

Si la nouvelle notion de $VecLink$ paraît simple, il n'est pas possible d'en dire autant de sa formulation mathématique, de son implémentation et de

sa gestion. Les questions suivantes mettent en évidence l'ensemble de ces problèmes:

- **Comment prendre en compte cette notion dans l'équation de DSI?**

Comme nous l'avons signalé dans le chapitre 1, un horizon géologique peut être considéré comme un graphe bidimensionnel $G(\Omega)$. Chaque lèvre de la faille peut être considérée comme un ensemble de segments S_i qui partagent leurs nœuds α_0, α_1 avec le graphe $G(\Omega)$. Je rappelle qu'aux deux lèvres d'une faille on associe deux courbes paramétriques $P(s)$ et $P'(s)$. Considérons deux nœuds $\mathbf{p}_i$ et $\mathbf{p}'_i$ définis, respectivement sur $P(s)$ et $P'(s)$, par leur abscisse curviligne s_i. Comme le montre la figure 4.19, les coordonnées $\varphi(\mathbf{p}_i)$ $\varphi(\mathbf{p}'_i)$ de ces nœuds peuvent être formulées comme suit :

$$\left| \begin{array}{l} \left\{ \begin{array}{l} \varphi(\mathbf{p}_i) = u \cdot \varphi(\alpha_0) + (1-u) \cdot \varphi(\alpha_1) \\ \qquad \text{avec} : u = \dfrac{\text{distance}_{(\alpha_1, \mathbf{p}_i)}}{\text{distance}_{(\alpha_0, \alpha_1)}} \end{array} \right. \\ \\ \left\{ \begin{array}{l} \varphi(\mathbf{p}'_i) = u' \cdot \varphi(\alpha'_0) + (1-u') \cdot \varphi(\alpha'_1) \\ \qquad \text{avec} : u' = \dfrac{\text{distance}_{(\alpha'_1, \mathbf{p}'_i)}}{\text{distance}_{(\alpha'_0, \alpha'_1)}} \end{array} \right. \end{array} \right.$$

L'équation 4.3 s'écrit de la manière suivante :

$$u \cdot \varphi(\alpha_0) + (1-u) \cdot \varphi(\alpha_1) - u' \cdot \varphi(\alpha'_0) - (1-u') \cdot \varphi(\alpha'_1) = \Delta_{\lambda_i \lambda'_i} \quad (4.4)$$

avec $\Delta_{\lambda_i \lambda'_i}$, le vecteur rejet *réel* à respecter.
L'expression locale de l'équation $DSI(\alpha)$ de cette contrainte s'exprime par la formule:

$$\forall \nu = (x,y,z) : \left[\begin{array}{l} \displaystyle\sum_{\alpha \in (\alpha_0, \alpha_1, \alpha'_0, \alpha'_1)} A_i^\nu(\alpha) \cdot \varphi^\nu(\alpha) = b_i^\nu \\ \\ avec : \left| \begin{array}{lcl} A_i^\nu(\alpha) & = & \left\{ \begin{array}{ccc} u & \text{si} & \alpha = \alpha_0 \\ 1-u & \text{si} & \alpha = \alpha_1 \\ -u' & \text{si} & \alpha = \alpha'_0 \\ -(1-u') & \text{si} & \alpha = \alpha'_1 \\ 0 & sinon & \end{array} \right. \\ \\ b_i^\nu & = & \Delta_{\lambda_i \lambda'_i} \end{array} \right. \end{array} \right.$$

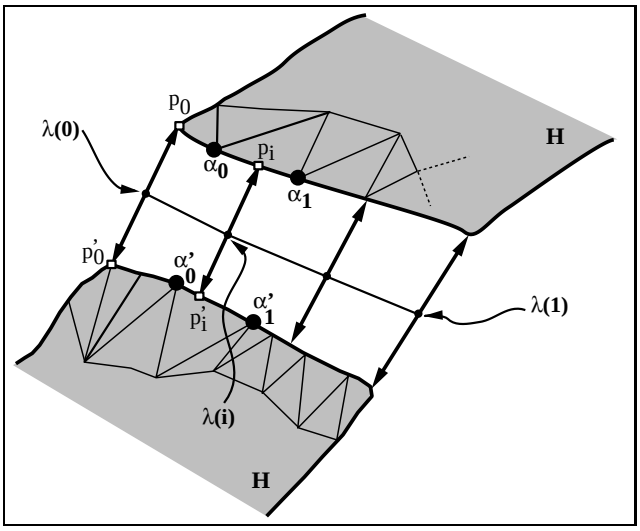

FIG. 4.19 – *Calcul des coordonnées des liens par rapport aux nœuds des lèvres de la faille*

Ainsi, au nœud α_0 de la figure 4.19, les paramètres γ^ν et Γ^ν seront les suivants:

$$\gamma^\nu(\alpha_0) = \varpi_i^2 \cdot (1-u)^2$$

$$\Gamma^\nu(\alpha_0) = \varpi_i^2 \cdot (1-u) \cdot \{\sum_{\beta \neq \alpha} A^\nu(\alpha) \cdot \varphi^\nu(\beta) - \Delta^\nu_{\mathbf{p}_i\mathbf{p}'_i}\}$$

Une fois les coefficients $\gamma(\alpha_0)$ et $\Gamma(\alpha_0)$ calculés, la formule présentée dans le chapitre 2 permet de calculer, au cours des itérations de DSI, la position du nœud α_0 qui minimise au sens de DSI la rugosité de la surface, tout en respectant la contrainte définie.

– **Comment définir les lèvres d'une faille?**
Comme nous l'avons signalé, la ligne d'intersection d'une faille et d'un horizon, représente les bords de l'horizon qui correspondent aux lèvres de la faille. En pratique, la définition des lèvres de la faille dépend de la forme géométrique de l'horizon et de la faille. L'exemple de la figure 4.20, illustre cette difficulté et montre que géologiquement le nœud (A)

peut faire partie de la lèvre de la faille alors que le nœud (B) ne peut pas en faire partie.

L'ambiguïté s'amplifie si un horizon est coupé par plusieurs failles qui s'intersectent (cf figure 4.21).

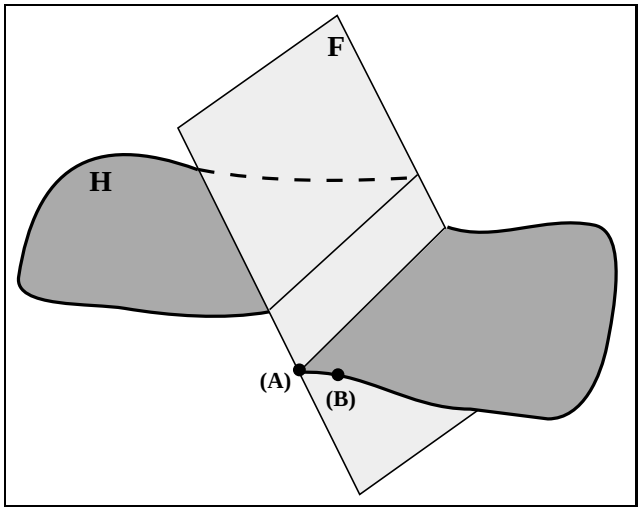

FIG. 4.20 – *Détermination des lèvres d'une faille*

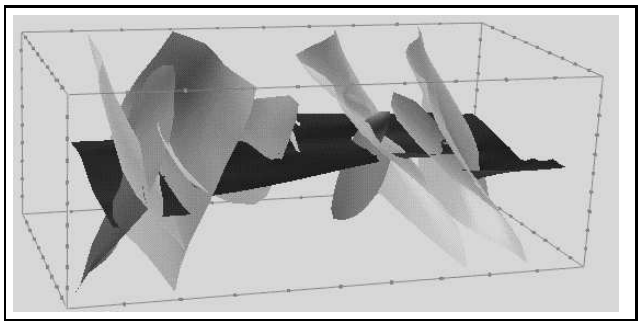

FIG. 4.21 – *Exemple d'un horizon (en noir) coupé par plusieurs failles (en gris) qui s'intersectent entre elles*

Une première solution consiste à faire appel au bon sens de l'utilisateur pour sélectionner les bords d'un horizon qui correspondent aux lèvres de chaque faille, mais ce travail devient fastidieux à partir du moment où les horizons sont complexes et/ou nombreux. La définition auto-

matique des lèvres de la faille, est néanmoins possible lors du calcul de l'intersection d'un horizon par une faille. Dans le système GOCAD, la fonction $TSURF\_Cut\_by\_Tsurf()$, permet de découper une surface le long de la ligne de son intersection avec une autre surface ([24]). Le principe de cette procédure consiste à créer, au niveau de la surface coupée, deux rangées de nœuds dédoublés le long de la ligne d'intersection entre les deux surfaces ; la triangulation est ensuite réévaluée de part et d'autre de cette ligne d'intersection.
Dans notre cas la surface découpée correspond à l'horizon géologique, et l'ensemble des nœuds dédoublés correspond aux deux lèvres de la faille (voir figure 4.22). En conclusion, si le découpage de l'horizon par la faille se fait en utilisant les procédures du système GOCAD , les lèvres de la faille seront automatiquement calculées, sinon l'utilisateur devra définir interactivement les deux lèvres de la faille.

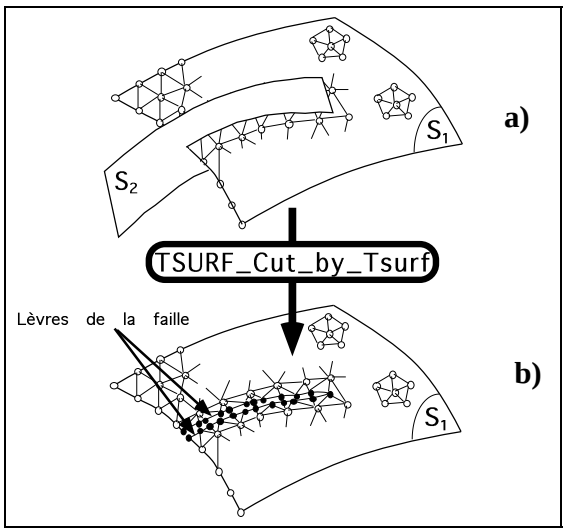

FIG. 4.22 – *Calcul automatique des lèvres d'une faille*

– **Comment calculer les directions des liens ?**
Si les différents jeux d'une faille se sont faits dans la même direction, nous pouvons supposer que les liens de la contrainte VLK correspondent aux *stries des failles*. Le calcul automatique des liens associés à la contrainte VLK consiste donc à calculer les stries de faille. Je rappelle que les stries d'une faille sont les traces, sur la surface de la faille, du déplacement des compartiments de la faille (cf figure 4.23). Pratiquement, il est difficile de mesurer ces stries sur le terrain, par suite de l'érosion des couches géologiques faillées et de la difficulté d'accès aux zones faillées $\cdots$.
Des méthodes mathématiques ont été proposées pour résoudre ce type

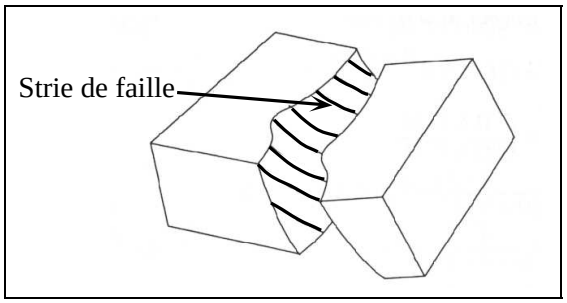

FIG. 4.23 – *Les stries d'une faille, d'après M. Thibaut*

de problème ([35]). Elles supposent que si le déplacement engendré par une faille ne s'accompagne pas d'une déformation interne des couches géologiques affectées (on parlera de l'approximation *bloc rigide*), la surface de la faille peut être assimilée à un filetage ([35]).

– *Définition d'un filetage*
Les filetages, par analogie à une vis et un écrou, sont des surfaces qui peuvent glisser les unes sur les autres en restant inchangées. Une surface S est un filetage si et seulement si, il existe un torseur non nul tangent à la surface.

– *Définition d'un torseur*
Un torseur τ est un champ de vecteurs $\overrightarrow{T(M)}$ défini en tout point M par la relation suivante :

$$\overrightarrow{T(M)} = \overrightarrow{V} + \overrightarrow{OM} \times \Omega \tag{4.5}$$

Dans cette expression, O est l'origine de l'espace $3D$, $\overrightarrow{V}$ est le torseur à l'origine et Ω est un vecteur non unitaire de rotation.

En géologie, si la surface de la faille est considérée comme un filetage, alors les lignes de champ du torseur associé à cette surface correspondent aux stries de la faille. Cela implique que la surface de la faille est partout tangente à un torseur ν. Une première approche de la prise en compte de la propriété des stries de failles a été proposée par M. Thibaut ([35]). Cette approche suppose que la surface de faille doit être construite de manière à respecter plusieurs critères dont la tangence à un torseur ν donné.
Dans notre cas, nous supposons que la surface de la faille (notée F) est connue et nous nous proposons de calculer le torseur décrivant le déplacement engendré par F.
Le vecteur joignant l'origine de l'espace $3D$ à un nœud α de F est noté $\varphi(\alpha)$. La recherche du torseur décrivant le déplacement de deux blocs rigides de part et d'autre de F revient à trouver les vecteurs V et Ω qui vérifient la relation suivante au niveau de chaque point α de F :

$$\forall \alpha \in F \ \overrightarrow{T(\alpha)} \cdot \overrightarrow{N_\alpha} \simeq 0 \tag{4.6}$$

– $\overrightarrow{N_\alpha}$ est la normale à la faille au point α,

Le problème se ramène donc à minimiser la relation :

$$J(\Omega, V) = \sum_{\alpha} |N_\alpha \cdot V + (\varphi(\alpha) \times \Omega) \cdot \overrightarrow{N_\alpha}|^2 \tag{4.7}$$

Si on pose :

$$N_\alpha = \begin{pmatrix} N_x \\ N_y \\ N_z \end{pmatrix} \quad \overrightarrow{\Omega} = \begin{pmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix}$$

$$\overrightarrow{V} = \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} \quad \varphi(\alpha) = \begin{pmatrix} \Delta_x \\ \Delta_y \\ \Delta_z \end{pmatrix}$$

Alors $J(\Omega, V)$ peut s'écrire de la façon suivante :

$$J(\Omega, V) = \sum_{M} |N_x \cdot V_x + N_y \cdot V_y + N_z \cdot V_z + A_x \cdot \Omega_x + A_y \cdot \Omega_y + A_z \cdot \Omega_z|^2 = 0$$

avec :

$$A_x = \begin{vmatrix} N_y & N_z \\ \Delta_y & \Delta_z \end{vmatrix}$$

$$A_y = \begin{vmatrix} N_x & N_z \\ \Delta_x & \Delta_z \end{vmatrix}$$

$$A_z = \begin{vmatrix} N_x & N_y \\ \Delta_x & \Delta_y \end{vmatrix}$$

Si on pose :

$$X = \begin{vmatrix} V_x \\ V_y \\ V_z \\ \Omega_x \\ \Omega_y \\ \Omega_z \end{vmatrix} \qquad Y(\alpha) = \begin{vmatrix} N_x \\ N_y \\ N_z \\ A_x \\ A_y \\ A_z \end{vmatrix}$$

Alors on vérifie facilement que l'on a :

$$\begin{aligned} J(\Omega, V) &= \textstyle\sum_\alpha |X^t \cdot Y(\alpha)|^2 \\ &= \textstyle\sum_\alpha \{X^t \cdot Y(\alpha)\} \cdot \{Y^t(\alpha) \cdot X\} \\ &= X^t \cdot (\textstyle\sum_\alpha Y(\alpha) \cdot Y^t(\alpha)) \cdot X \end{aligned}$$

Si on pose :

$$Q = \sum_\alpha Y(\alpha) \cdot Y^t(\alpha)$$

alors $J(\Omega, V)$ peut s'écrire :

$$J(\Omega, V) = J(X) = X^t \cdot Q \cdot X$$

Comme on peut le voir $J(\Omega, V) = J(X)$ est une forme quadratique dont le minimum est atteint lorsque X est le vecteur propre de Q correspondant à la plus petite valeur propre λ, et l'on a alors :

$$J(\Omega, V) = J(X) = \lambda \cdot \| X \|^2$$

Les composantes Ω et V du vecteur X ainsi définies permettent de calculer le torseur $T(\alpha)$ en tout point α de la faille. Si l'hypothèse de *bloc rigide* est vérifiée alors $T(\alpha)$ définit la direction de déplacement de la faille en tout point α de la faille.

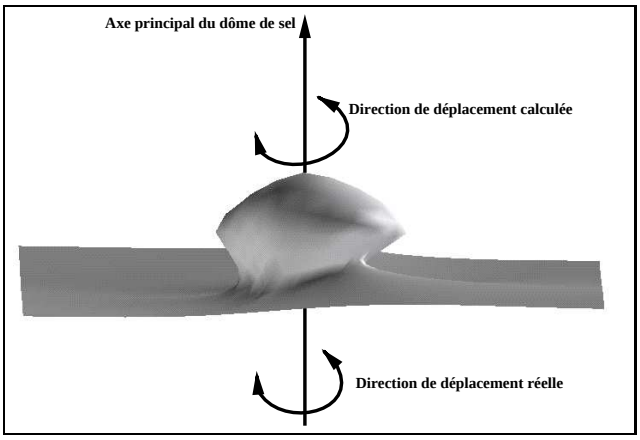

FIG. 4.24 – *Pour un dôme de sel le mouvement de déplacement prédominant est la rotation autour de l'axe principal du dôme de sel. La direction de déplacement réelle et la direction de déplacement calculée sont confondues*

Des tests ont été faits sur des cas réels, les résultats obtenus sont intéressants (cf figure 4.24).

Cependant, comme nous l'avons signalé dans le chapitre 3, le jeu d'une faille dépend non seulement des paramètres géométriques de la surface de cette faille, mais aussi des paramètres thermodynamiques, mécaniques et géologiques (cf figure 4.25). Les résultats proposés par la méthode présentée seront alors considérés comme solution initiale du problème de calcul des directions des liens. Cette solution est susceptible d'être transformée par l'utilisateur.

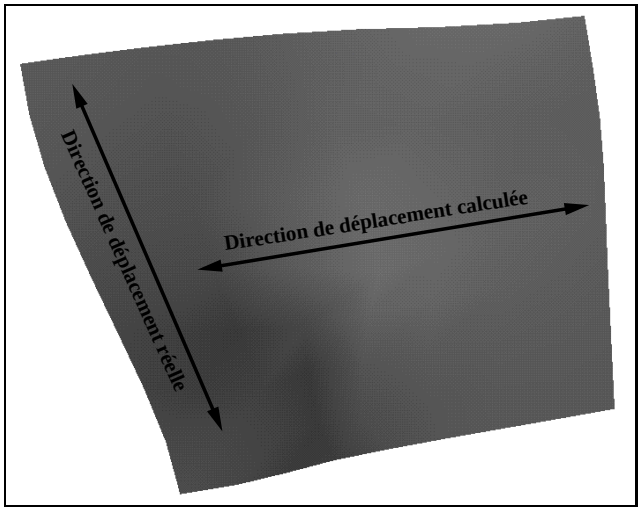

FIG. 4.25 – *La direction de déplacement calculée est différente de la direction de déplacement réelle*

- **Comment mettre à jour les liens ?**
  Il arrive parfois que les lèvres d'une faille soient modifiées, certains liens perdent alors le contact avec les lèvres de la faille et deviennent donc inefficaces. Ce résultat est inacceptable car le contact entre les deux bords de la faille, via les vecteurs rejets, se trouve rompu. Il faut donc réévaluer la position des liens pour assurer un contact permanent entre les deux lèvres d'une faille. Dans le système GOCAD le mécanisme du *superviseur de projets*, présenté dans le chapitre 2, permet de détecter toutes les modifications géométriques et/ou topologiques qui affectent un objet. Ce mécanisme déclenche par la suite un ensemble de procédures qui mettent à jour tous les liens qu'un objet peut posséder avec les autres objets ([8]). Dans le cas d'un horizon faillé, toute modification de ce dernier sera détectée par le *superviseur de projets*, qui appelle la procédure de mise à jour de la contrainte.

La conception de la contrainte VLK a nécessité la mise en place d'un couple de classes *(VLK_ INFO, VLK)* (cf chapitre 2). Ces classes dérivent respectivement de CNSTR_INFO et CNSTR. Les fonctions, membres virtuelles des classes CNSTR_INFO et CNSTR, seront définies au niveau des classes VLK_INFO et VLK. Parmi ces fonctions on trouve :

- les fonctions de création et de destruction des instances de la classe VLK,
- la fonction *fuzzyCnstr()*, qui permet à DSI de prendre en compte une instance de la classe VLK de manière floue (cf chapitre 2). La fonction *fuzzyCnstr()* consiste à calculer les coefficients Γ_i et γ_i au niveau de chaque atome des lèvres de la faille. Ces coefficients seront calculés selon la formule 4.1, il seront utilisés par DSI pour respecter la contrainte VLK,
- les fonctions de mise à jour des contraintes si l'horizon a été géométriquement ou géologiquement modifié.

De nouvelles fonctions membres, spécifiques à la contrainte VLK, seront ajoutées dans les classes VLK_INFO et VLK. Parmi ces fonctions on trouve :

- une fonction membre qui se charge de fixer le déplacement engendré par la faille au niveau d'un lien. Cette fonction sera indirectement appelée par l'utilisateur pour fixer, interactivement, le rejet d'une faille au niveau d'un lien,

– une fonction membre qui calcule la position des liens sur les lèvres d'une faille.

4.3.2.2 Résultats obtenus

Les résultats présentés correspondent à une série d'images illustrant le contrôle du déplacement engendré par une faille sur un horizon. Ces exemples montrent que la notion de $VecLink$ introduit une souplesse dans la gestion du rejet d'une faille.

– **Initialisation et modification des liens**
La figure 4.26 montre un ensemble de liens définis entre les deux lèvres d'une faille, quelques vecteurs $\overrightarrow{T(s_i)}$ (représentés par des flèches) ont été fixés. Le résultat de l'interpolation, à l'aide de la méthode DSI des vecteurs $\overrightarrow{T(s_i)}$ restés libres, est donné par la figure 4.27.

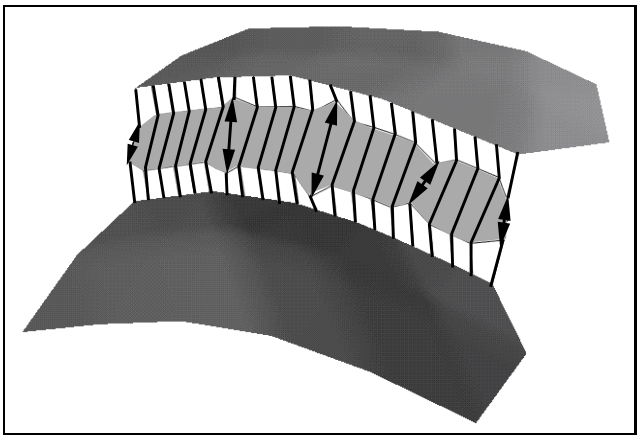

FIG. 4.26 – *Mise en place de la contrainte VecLink sur les lèvres d'une faille*

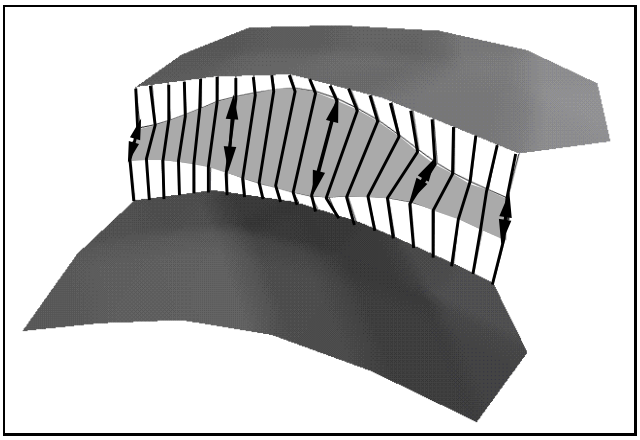

FIG. 4.27 – *Interpolation par la méthode DSI des rejets réels d'une faille*

– **Interpolation de la surface avec respect des rejets imposés**
Les figures 4.28 et 4.29 montrent le résultat de l'interpolation, en utilisant DSI, d'une surface avec respect des rejets imposés par la contrainte $VecLink$. Pour chaque faille, le rejet imposé correspond à la bande grise. Pour maintenir la tension au niveau de la surface, les bords ne correspondant pas aux lèvres des failles sont autorisés à se déplacer selon la direction verticale (cf chapitre 2).

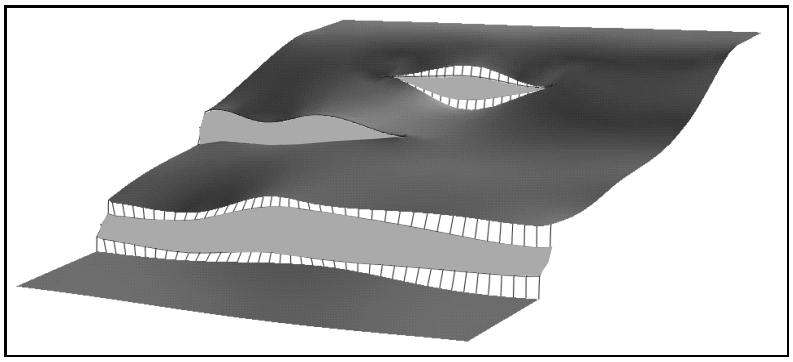

FIG. 4.28 – *Mise en place des contraintes VecLink au niveau des bords de la surface*

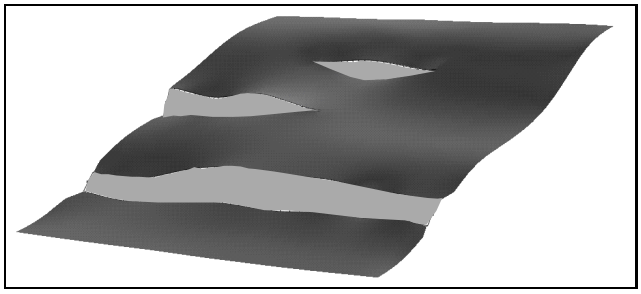

FIG. 4.29 – *Résultat de l'interpolation de la surface par la méthode DSI*

– **Transformation du rejet normal d'une faille en rejet inverse**
La facilité de manipulation de la contrainte $VecLink$, permet de modifier le sens du rejet d'une faille. Ainsi un rejet normal (cf figure 4.30) peut facilement devenir inverse (cf figure 4.31).

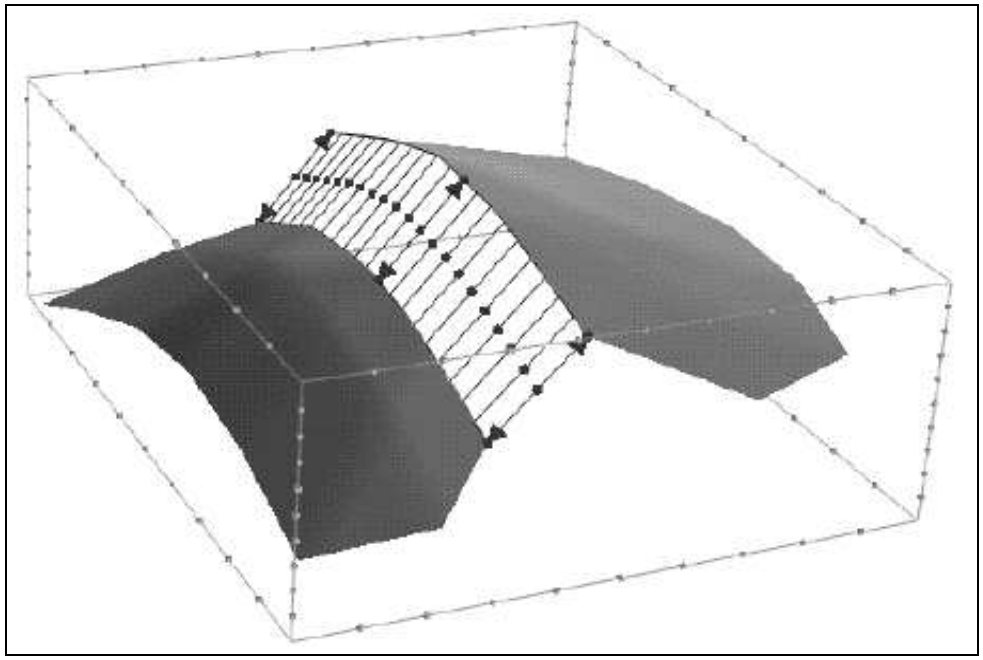

FIG. 4.30 – *Rejet normal d'une faille*

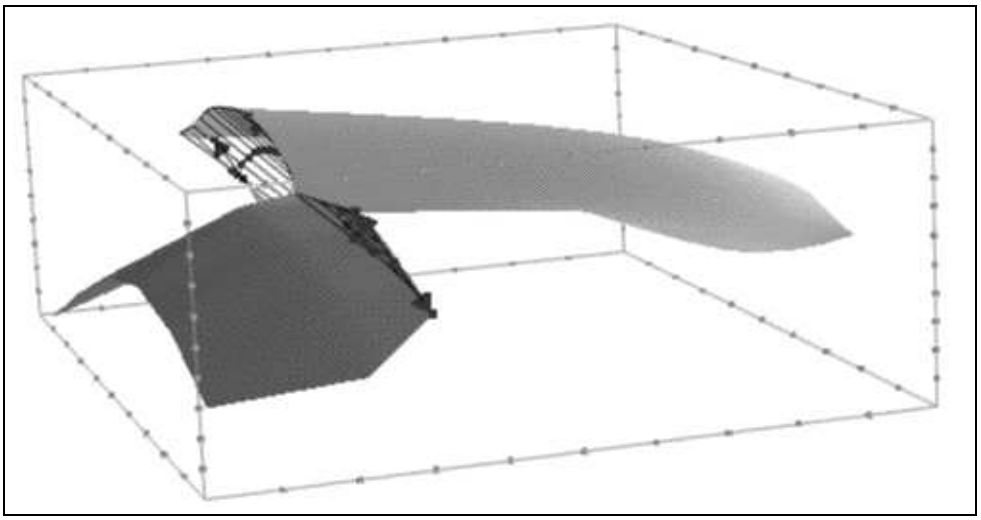

FIG. 4.31 – *Rejet inverse d'une faille*

4.3.2.3 Conclusion sur la contrainte Veclink

La robustesse de la nouvelle contrainte $VecLink$, que je viens de présenter, réside dans :

- La facilité de manipulation et de gestion de cette contrainte à condition de définir correctement les lèvres de la faille.
- L'indépendance de la position des liens et du maillage des lèvres de la faille. La figure 4.32 montre le résultat obtenu dans le cas où les liens sont connectés aux noeuds des lèvres de la faille.

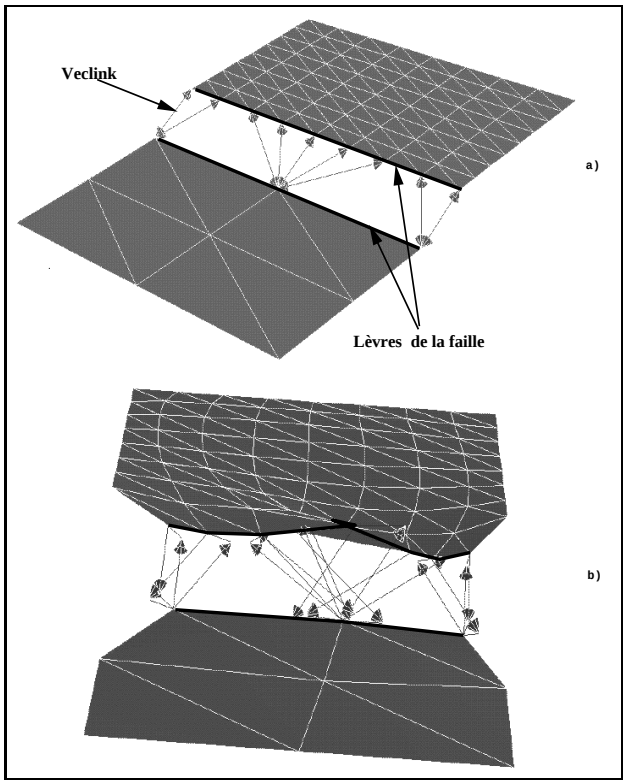

FIG. 4.32 – *Exemple d'enchevêtrement des triangles et des liens. a) Mise en place des liens entre les noeuds des lèvres de la faille, b) Résultat obtenu après interpolation par DSI*

- La possibilité d'introduction, au niveau d'une surface faillée, d'une notion de continuité physique le long d'une discontinuité géométrique re-

présentée par la faille ([13][14]). Ce principe est très important car les propriétés physiques intrinsèques, d'un horizon, de part et d'autre d'une zone de faille ne sont pas affectées par la présence d'une faille. Sur la figure 4.33, les courbes d'isovaleur montrent la discontinuité d'une propriété physique, de part et d'autre de la zone de faille. Après installation de la contrainte $VecLink$ sur les lèvres de la faille, la continuité de la propriété physique de part et d'autre de la zone de faille est rétablie (cf figure 4.34).

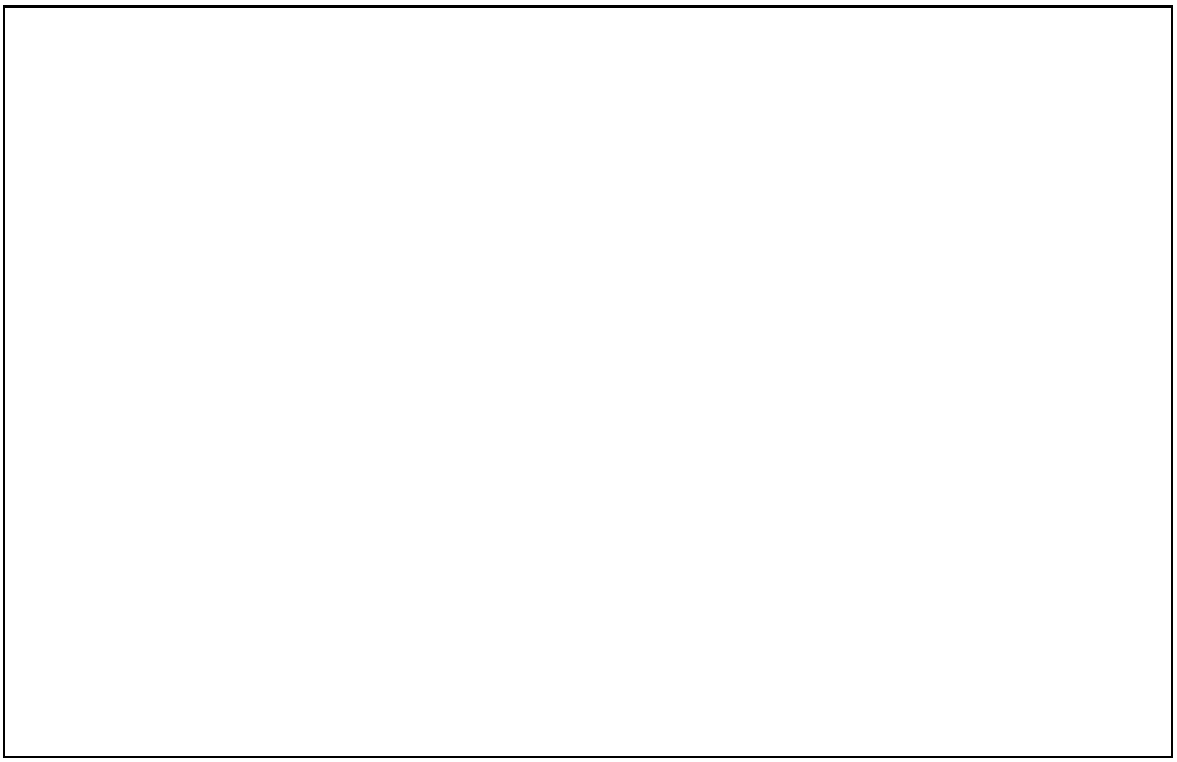

FIG. 4.33 – *Discontinuité d'une propriété physique de part et d'autre d'une faille*

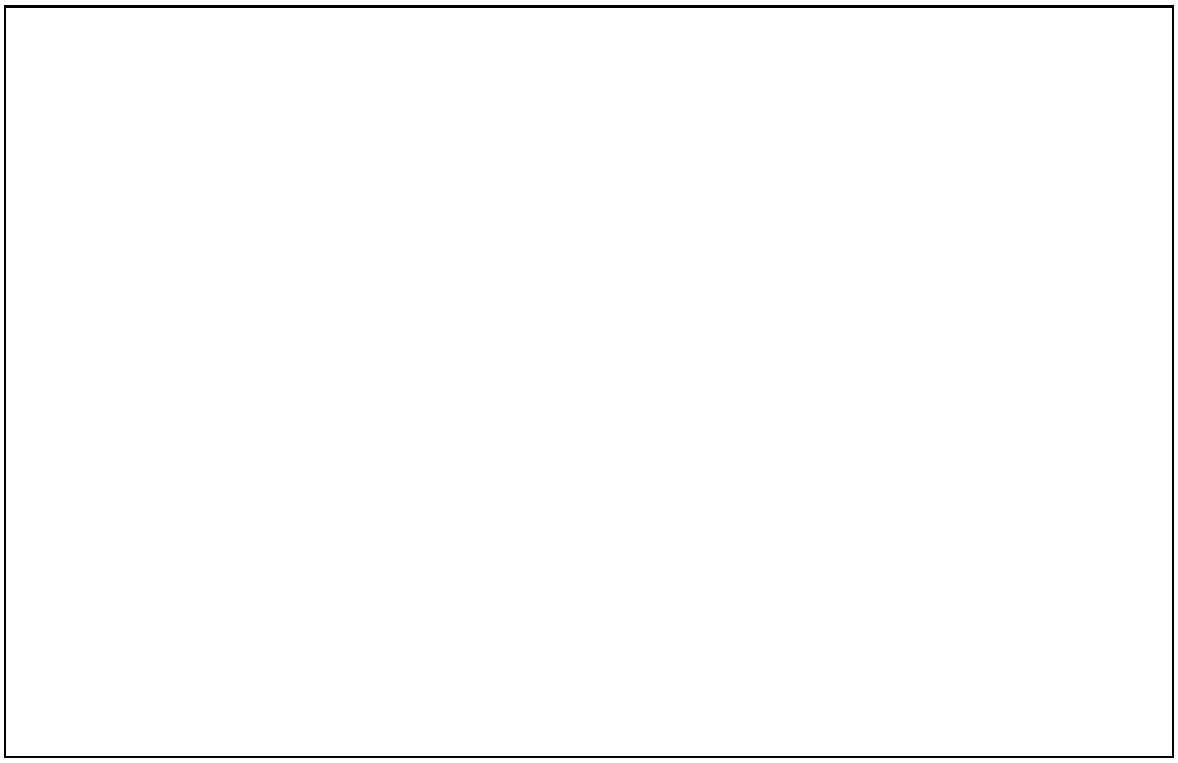

FIG. 4.34 – *Rétablissement de la continuité d'une propriété physique de part et d'autre de la zone de faille*

Il est, cependant, difficile du point de vue pratique d'évaluer la variation du rejet engendré par une faille sur l'ensemble des horizons qu'elle coupe. La figure 4.35 illustre ce propos. Elle montre une faille F qui coupe quatre horizons géologiques (H_0, H_1, H_2, H_3). En effet, les géologues ont l'habitude d'étudier les failles à partir de leur géométrie et de la cartographie de leurs rejets. Cependant les méthodes existantes ne permettent de cartographier le rejet d'une faille que sur le plan de la faille. Cela peut introduire des erreurs d'estimation importantes. Nous avons donc développé une nouvelle méthode de cartographie des rejets d'une faille. Cette cartographie se fera sur la surface de faille et non pas sur le plan de la faille.

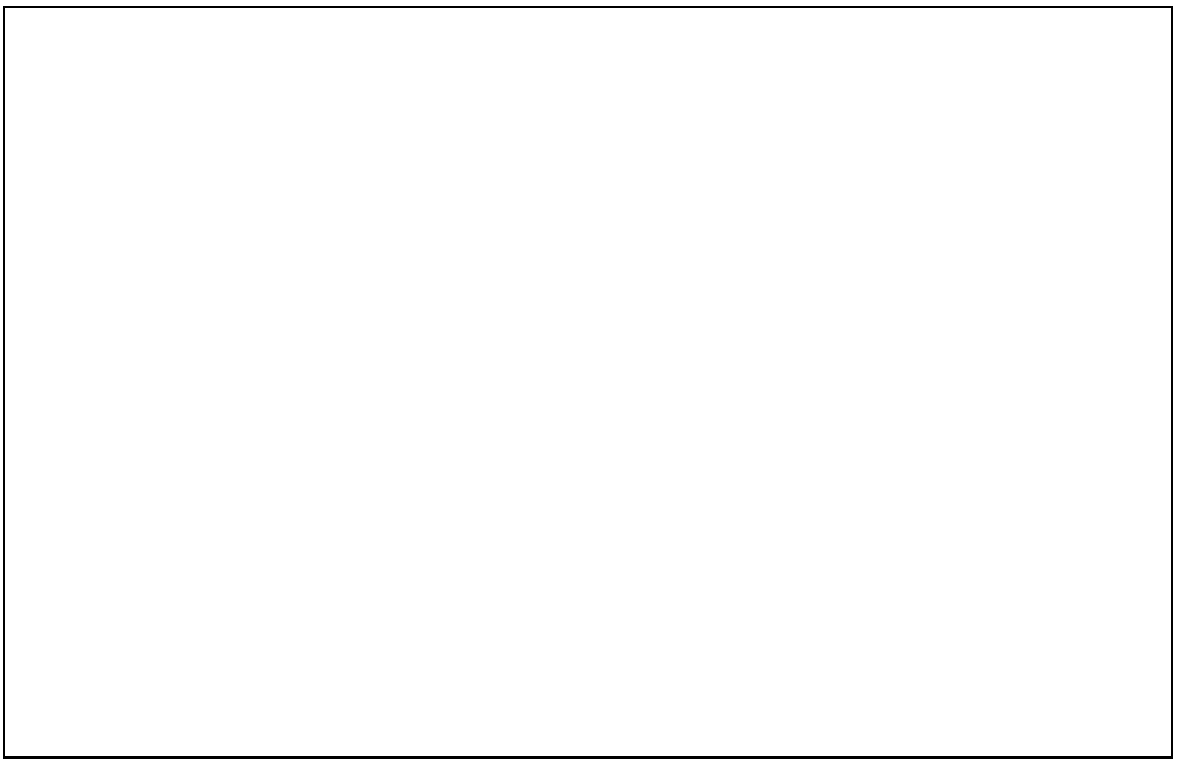

FIG. 4.35 – *Exemple d'un ensemble d'horizons géologiques coupés par la même faille*

4.3.2.4 Cartographie des rejets de faille

Considérons, sur la figure 4.36, la surface S_0 qui est coupée par la faille F. La contrainte $VecLink$ a été installée sur les lèvres de la faille. Si on considère le point λ_i de la lèvre supérieure de la faille, la valeur absolue du rejet engendré par la faille en ce point est donnée par la norme du vecteur $\overrightarrow{(\lambda_i, \lambda_i^{'})}$. Ainsi, pour chaque lèvre de la faille nous définissons un ensemble de couples $E_{S_0}((\lambda_0, \lambda_0^{'}), (\lambda_1, \lambda_1^{'}), \cdots, (\lambda_n, \lambda_n^{'}))$, permettant d'estimer la valeur du rejet de la faille en chaque point de ses lèvres. La cartographie des rejets de la faille F consiste en :

- La mise en place d'un lien (cf chapitre 2) entre chaque faille et le groupe d'horizons géologiques qu'elle coupe. Ce lien contient toutes les informations nécessaires pour recartographier le rejet de la faille si une modification survient sur les horizons et/ou la faille qui les coupe.
- La projection, selon une direction $\overrightarrow{D_{\lambda_i}}$, de chaque point λ_i de E_{S_0} sur la surface de faille. La direction de projection peut être calculée selon la procédure adoptée pour la contrainte $OnTsurf$. Si λ_i intercepte un triangle $T(\varphi_0, \varphi_1, \varphi_2)$ de la surface de faille en un point d'impact I_{λ_i}, nous affectons alors à chaque nœud φ_i du triangle intersecté, une valeur de propriété qui correspond au rejet de la faille au point λ_i. Ces propriétés pourront être interpolées sur la surface de faille en utilisant la méthode DSI ([14]). La visualisation sur la faille, de ces propriétés interpolées, peut se faire alors à l'aide des courbes d'isovaleur présentées dans le chapitre 1.

Cette méthode a été testée sur un modèle contenant trois horizons géologiques coupés par une faille. Les résultats sont donnés par les figures suivantes :

- la figure 4.37 montre le contact entre la faille et l'horizon H_0,
- le résultat de l'interpolation des rejets de la faille au niveau des trois horizons est présenté sur la figure 4.38. Sur cette figure, le rejet de la faille peut être estimé à l'aide des courbes d'isovaleurs ou par l'épaisseur des différentes bandes de rejet.

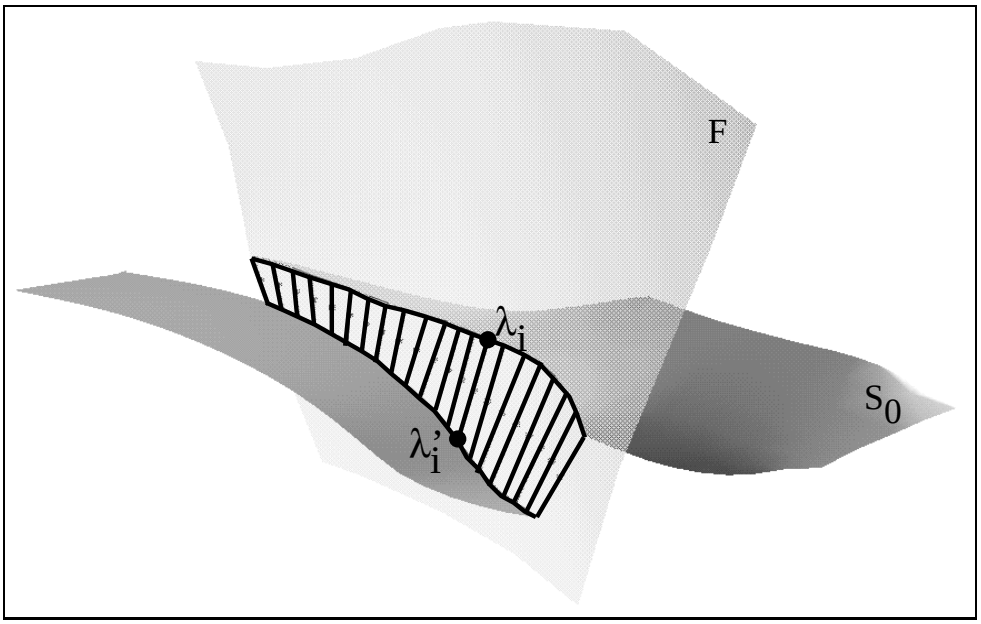

FIG. 4.36 – *Exemple d'une surface S_0 coupée par une faille F, avec installation des $VecLink$ sur les lèvres de F*

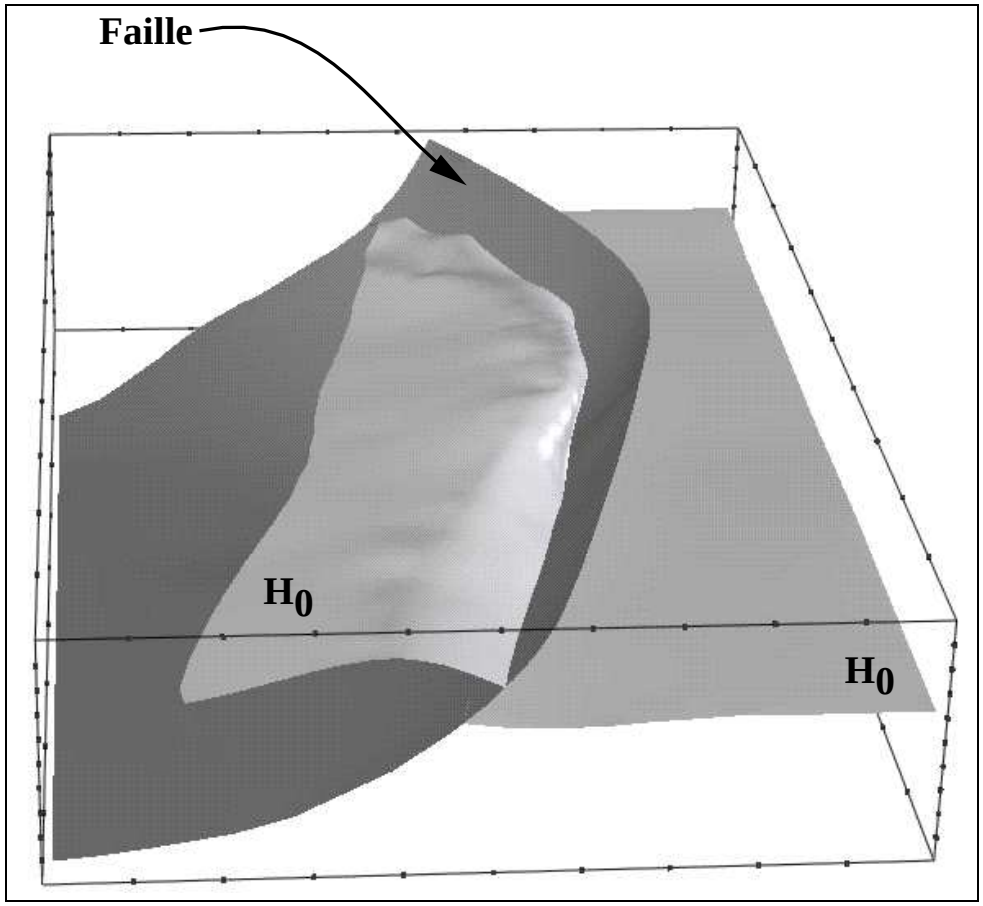

FIG. 4.37 – *Exemple d'un contact faille-horizon*

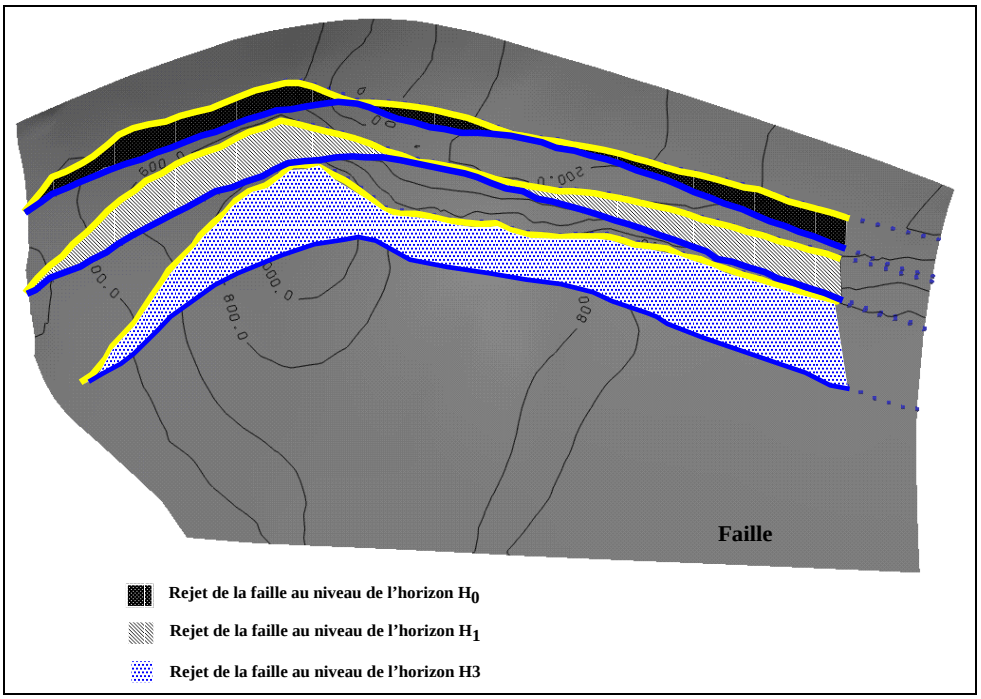

FIG. 4.38 – *Cartographie, sur la surface de la faille, du rejet engendré sur 3 horizons*

4.3.3 La contrainte OnBorder

Je rappelle que pour modéliser la relation *Faille-Horizon* j'ai développé deux types de contraintes géométriques : la contrainte *On-Tsurf* et la contrainte *VecLink*. Ces deux contraintes sont indispensables, mais, comme nous l'avons signalé au début du paragraphe 4.3, elles sont insuffisantes pour obliger les extrémités des lèvres d'une faille à se déplacer le long du bord de la faille. J'ai donc développé la contrainte *OnBorder*. En général cette contrainte consiste à obliger un nœud d'une surface à glisser le long d'une courbe polygonale.

4.3.3.1 Principe de la contrainte OnBorder

Le principe de la méthode choisie consiste, comme le montre la figure 4.39, à projeter un nœud d'une surface sur la ligne polygonale correspondant au bord d'une autre surface. La droite issue du nœud A, appartenant à l'horizon, intersecte le bord de la faille au point d'impact I. Le but de la contrainte OBD[3] est de déplacer le nœud A vers le point I. A l'image des contraintes DSI présentées, le contact entre le nœud A et le bord de la faille doit être recalculé chaque fois qu'une des deux surfaces a été modifiée.
Lors de l'implémentation de cette contrainte, nous avons rencontré plusieurs problèmes dont :

- **La détection des atomes concernés par cette contrainte**
  Comme nous l'avons signalé, lors de l'étude des contraintes $OnTsurf$ et $VecLink$, il est difficile de définir automatiquement les lèvres de la faille. Ainsi la spécification des atomes qui seront concernés par la contrainte *OnBorder* ne peut être faite sans l'aide de l'utilisateur. Cependant, dans le cas où l'intersection de la faille et de l'horizon a été faite par la procédure *TSURF_Cut_by_Tsurf()* décrite précédemment, il est possible de détecter les atomes de l'horizon qui correspondent aux extrémités des lèvres de la faille.

- **Le calcul de la position du point d'impact**
  La position du point d'impact I dépend en premier lieu de la direction de projection du nœud A. Plusieurs directions de projections peuvent être choisies par l'utilisateur, et à chaque direction de tir $\overrightarrow{Tir_i}$ correspond un point d'impact I_i. Cependant, pour automatiser la procédure

3. abréviation de OnBorder

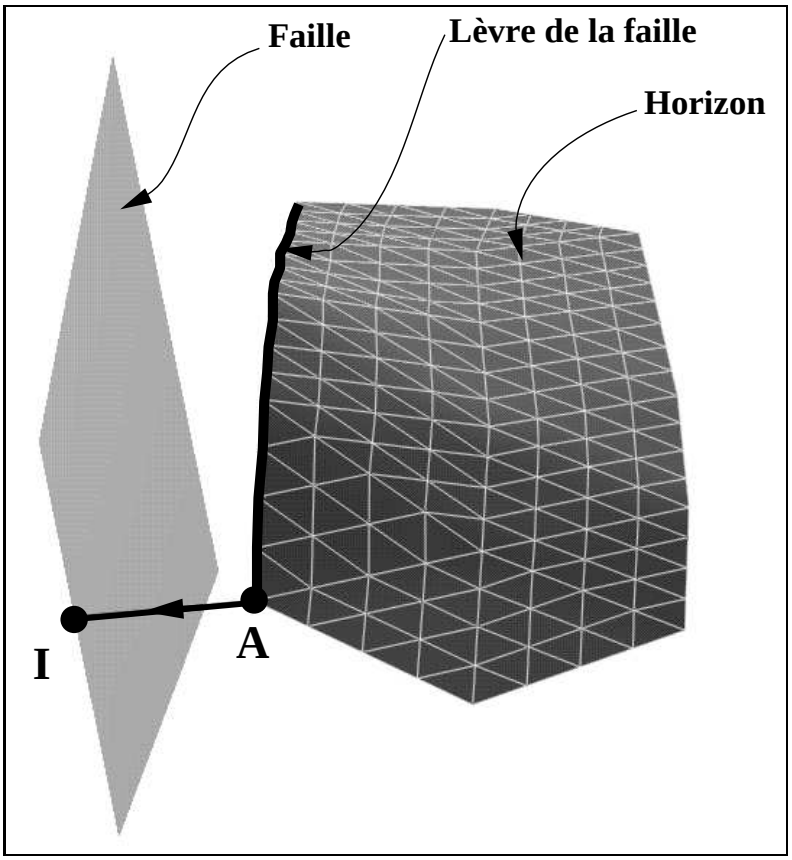

FIG. 4.39 – *Exemple de la contrainte OnBorder*

de projection et de calcul du point d'impact, nous proposons le principe suivant : Si, comme l'illustre la figure 4.40, $S(\mathbf{p}_0, \mathbf{p}_1)$ est le segment de la polygonale L le plus proche du point $\mathbf{P}_i$, le point d'impact $\mathbf{I}_i$ correspondra à la projection orthogonale de $\mathbf{P}_i$ sur la droite portant le segment S. Si le point d'impact $\mathbf{I}_i$ est extérieur au segment S, le point $\mathbf{P}_i$ sera projeté sur le segment voisin. Pour le calcul du segment voisin, considérons u la coordonnée barycentrique du point d'impact $\mathbf{I}_i$ par rapport à $\{p_0, p_1\}$ (u vaut 0 au nœud p_0 et 1 au nœud p_1). Comme pour les triangles, nous adopterons la convention suivante (cf figure 4.41):

Considérons un segment p_seg. La position du segment adjacent p_seg[i] se situe du côté opposé au sommet p_atom[i]

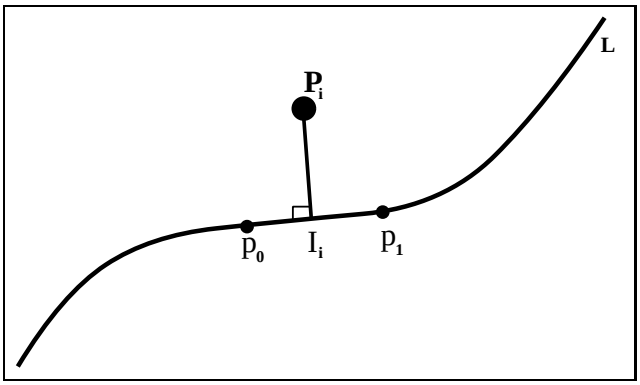

FIG. 4.40 – *Projection orthogonale d'un point sur une ligne polygonale*

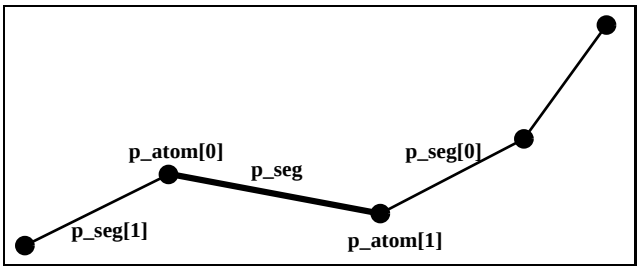

FIG. 4.41 – *Convention de la numérotation adoptée*

de p_seg Si par exemple u est supérieur à 1.0, le point $\mathbf{P}_i$ sera projeté sur le segment voisin $p\_seg[0]$. Cette méthode est locale et évite de considérer tous les segments de L.

– **Prise en compte de cette contrainte dans l'équation de** DSI
Considérons sur la figure 4.42 :

  – **P** est le point à projeter,

  – **L** est la droite sur laquelle on désire projeter le point **P**. Cette droite est définie par un point $\mathbf{M}_0(x_0, y_0, z_0)$ et par son vecteur directeur $\mathbf{D}(D^x, D^y, D^z)$,

  – **p** est le point de projection orthogonale de **P** sur la droite L. Le point **p** est défini par ses coordonnées (p_x, p_y, p_z) dans le repère $(O|x, y, z)$

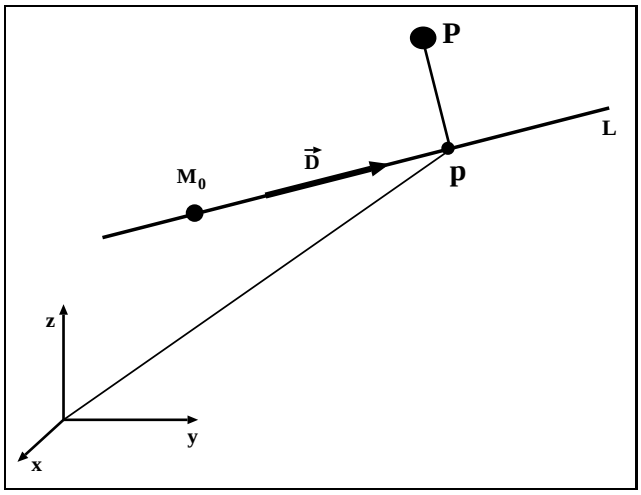

FIG. 4.42 – *Signification de l'équation canonique*

L'équation canonique de la droite L nous permet d'écrire :

$$\frac{\mathbf{p}_x - x_0}{D_x} = \frac{\mathbf{p}_y - y_0}{D_y} = \frac{\mathbf{p}_z - z_0}{D_z} \tag{4.8}$$

Cette relation peut s'écrire de la façon suivante :

$$\begin{cases} D_y \cdot \mathbf{p}_x &= D_x \cdot (\mathbf{p}_y - y_0) + x_0 \cdot D_y \\ D_z \cdot \mathbf{p}_x &= D_x \cdot (\mathbf{p}_z - z_0) + x_0 \cdot D_z \end{cases}$$

$$\begin{cases} D_x \cdot \mathbf{p}_y &= D_y \cdot (\mathbf{p}_x - x_0) + y_0 \cdot D_x \\ D_z \cdot \mathbf{p}_y &= D_y \cdot (\mathbf{p}_z - z_0) + y_0 \cdot D_z \end{cases}$$

$$\begin{cases} D_y \cdot \mathbf{p}_z &= D_z \cdot (\mathbf{p}_y - y_0) + z_0 \cdot D_y \\ D_x \cdot \mathbf{p}_z &= D_z \cdot (\mathbf{p}_x - x_0) + z_0 \cdot D_x \end{cases}$$

Ce système d'équations peut être traduit sous forme d'une contrainte DSI de type **floue** (cf chapitre 2). Ainsi, si la contrainte OBD est la $i^{\text{ème}}$ contrainte au niveau du nœud $\mathbf{P}$, et si j'affecte à cette contrainte le poids ϖ_i^2, les coefficients permettant de prendre en compte cette

contrainte dans l'équation de DSI seront les suivants :

$$\begin{cases} \Gamma_i^x &= \varpi_i^2 \cdot D_z \cdot (z_0 \cdot D_x - x_0 \cdot D_z - D_x \cdot p_z) \\ &+ \varpi_i^2 \cdot D_y \cdot (y_0 \cdot D_x - x_0 \cdot D_y - D_x \cdot p_y) \\ \Gamma_i^y &= \varpi_i^2 \cdot D_z \cdot (z_0 \cdot D_y - y_0 \cdot D_z - D_y \cdot p_z) \\ &+ \varpi_i^2 \cdot D_x \cdot (x_0 \cdot D_y - y_0 \cdot D_x - D_y \cdot p_x) \\ \Gamma_i^z &= \varpi_i^2 \cdot D_y \cdot (y_0 \cdot D_z - z_0 \cdot D_y - D_z \cdot p_y) \\ &+ \varpi_i^2 \cdot D_x \cdot (z_0 \cdot D_x - x_0 \cdot D_z - D_z \cdot p_x) \end{cases}$$

$$\begin{cases} \gamma_i^x &= D_y^2 + D_z^2 \\ \gamma_i^y &= D_x^2 + D_z^2 \\ \gamma_i^z &= D_y^2 + D_x^2 \end{cases}$$

La contrainte OBD sera installée au niveau des extrémités des lèvres de la faille, afin de les obliger à glisser le long du bord de la faille. A chaque itération de DSI chaque nœud **P**, concerné par cette contrainte, se déplacera vers le bord de la faille pour vérifier la relation 4.8. Pour que le contact entre les extrémités des lèvres de la faille et le bord de la faille soit parfait, la contrainte OBD sera considérée comme une contrainte de type **dure** (cf chapitre 2). Si nous reprenons la figure 4.42, la contrainte OBD consistera à déplacer le nœud **P** vers le point **p**.

La contrainte OBD sera donc prise en compte dans l'équation DSI de deux manières :

- de manière **floue** en calculant les coefficients Γ_i^ν et γ_i^ν,
- de manière dure en déplaçant chaque nœud $\mathbf{P}_i$ vers le point d'impact $\mathbf{p}_i$ correspondant.

A l'image des deux contraintes précédentes, la mise en place de la contrainte OBD a nécessité la création d'un couple de classes *(OBD_INFO,OBD)* (cf chapitre 2). Ces classes dérivent respectivement de CNSTR_INFO et CNSTR. Les fonctions membres virtuelles des classes CNSTR_INFO et CNSTR seront définies au niveau des classes OBD_INFO et OBD. Parmi ces fonctions on retrouve :

- les fonctions de création et de destruction des instances de la classe OBD,

- la fonction *fuzzyCnstr()* qui se charge de calculer les coefficients Γ_i et γ_i permettant à DSI de prendre en compte une instance de la classe OBD comme une contrainte floue.

- la fonction *hardCnstr()*, permettant à DSI de prendre en compte une instance de la classe OBD comme une contrainte dure. Cela permet d'avoir un contact parfait entre l'atome de l'horizon concerné par la contrainte et le bord de la faille,

- les fonctions de mise à jour des contraintes si un des deux objets, horizon ou la faille, a été modifié.

4.3.3.2 Conclusion sur la contrainte OnBorder

L'avantage de la méthode présentée réside dans la facilité d'implémentation et les bons résultats qu'elle permet d'obtenir. Je désire rappeler, encore une fois, que cette contrainte ne peut être utilisée, dans le cadre de la relation Faille-Horizon, que si la faille s'amortit à l'intérieur de l'horizon qu'elle coupe. Un exemple de la combinaison de la contrainte $OnBorder$ et la contrainte $OnTsurf$ est donné par les figures suivante :

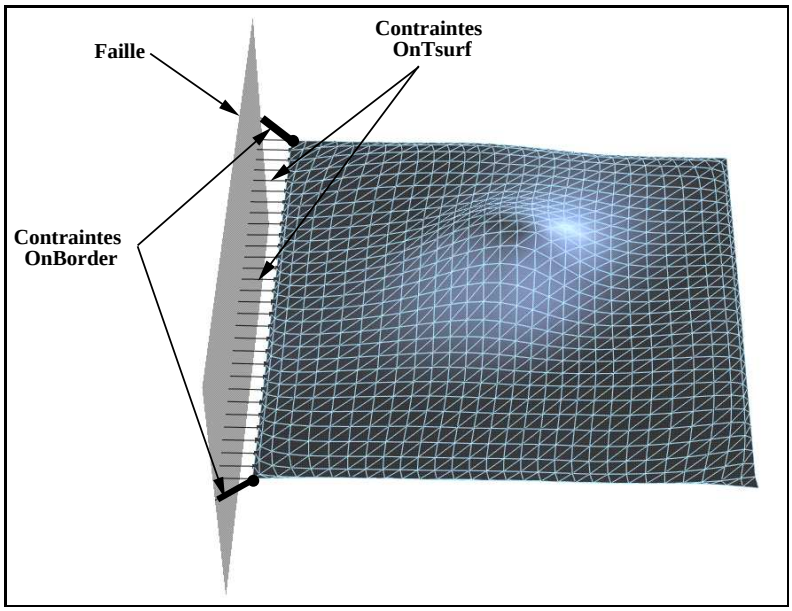

FIG. 4.43 – *Mise en place de la contrainte OnBorder et la contrainte OnTsurf sur le bord d'une horizon*

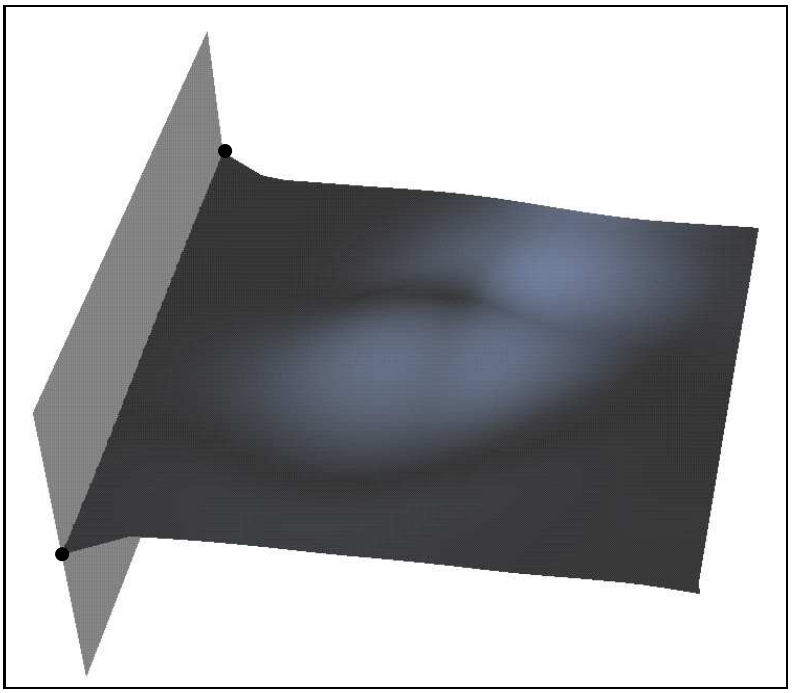

FIG. 4.44 – *Résultat de l'interpolation de l'horizon avec DSI*

4.4 Conclusions sur la modélisation des failles

Le but de ce chapitre était de montrer qu'il est possible de modéliser des relations géométriques complexes entre les surfaces en combinant des contraintes géométriques DSI. Ces contraintes sont implémentées selon le modèle Atomique-Contrôleur et en utilisant les concepts de la programmation orientée objet.

Si les contraintes développées dans ce chapitre consistaient à modéliser la relation faille-horizon, il faut noter que ces mêmes contraintes s'inscrivent dans le cadre général du contrôle géométrique des bords d'une surface par une autre surface.

Attention ! il ne faut pas crier victoire. La modélisation géométrique des failles en général, et de la relation faille-horizon en particulier, est loin d'être achevée et cela pour différentes raisons, dont :

- les nombreuses lacunes dans la connaissance des phénomènes géologiques à l'origine des failles,
- la rareté et l'imprécision des informations concernant les failles,
- l'approche adoptée suppose qu'une faille peut être assimilée à une surface, or dans la pratique, il faut parler d'une zone de faille car les failles peuvent avoir des épaisseurs non négligeables (cf chapitre 3).

Cependant, notre approche peut être qualifiée d'originale car elle permet de modéliser les failles tout en prenant en compte les relations qu'elle peuvent avoir avec les horizons coupés par la faille. Cette approche va contribuer à la mise en place d'un prototype de *générateur automatique de modèles géologiques*, permettant de générer des modèles géologiques complexes, sous le contrôle du géologue.

Chapitre 5

Vers un générateur automatique de modŁles géologiques

Le rêve de tout géologue est de disposer d'un système suffisamment performant, pour construire automatiquement des modèles géologiques contenant des surfaces complexes, à partir de données hétérogènes (semis de points, coupes seriées, données de puits, $\cdots$). Le développement d'un tel système nécessite la prise en compte de règles géologiques élémentaires, telles que la relation *faille-horizon*, de règles géométriques telles que la représentation des surfaces géologiques par des surfaces lisses (cf chapitre 1). Je désire insister sur le fait que ce chapitre n'est pas un manuel d'utilisation de la nouvelle approche de génération automatique de modèles géologiques, mais il s'agit plutôt de présenter le principe général de la nouvelle approche.

5.1 Introduction

De nombreux logiciels de modélisation géométrique des surfaces naturelles ont tenté de résoudre le problème de génération automatique de modèles géologiques. Pour des raisons de confidentialité je n'ai pas pu étudier en détail le fonctionnement de ces logiciels, cependant à partir des présentations, lors

des congrès de géophysique, de ces logiciels je peux dire :

- Les systèmes proposés sont entièrement automatiques. L'utilisateur introduit les informations concernant son modèle géologique (données géologiques et caractéristiques des surfaces composant le modèle) et le système se charge de générer automatiquement le modèle géologique.
- Si l'utilisateur désire modifier une des surfaces géologiques composant le modèle, il doit reconstruire son modèle géologique depuis le début.
- L'utilisateur ne peut pas modifier le mode de construction des surfaces de son modèle géologique.

En résumé, ces systèmes peuvent être qualifiés de *boîtes noires*, dans lesquelles l'utilisateur introduit les informations géologiques concernant son modèle ; en sortie le système fournit un modèle géologique figé (cf figure 5.1). Pour des modèles géologiques simples on peut espérer obtenir des résultats intéressants avec ces systèmes. Cependant, l'impossibilité de modification du modèle, pendant ou après sa construction, handicape ces logiciels. Pour remédier à ce problème nous avons mis au point un prototype de générateur et d'éditeur de modèles géologiques (G.E.M.G) dans le cadre du projet GOCAD.

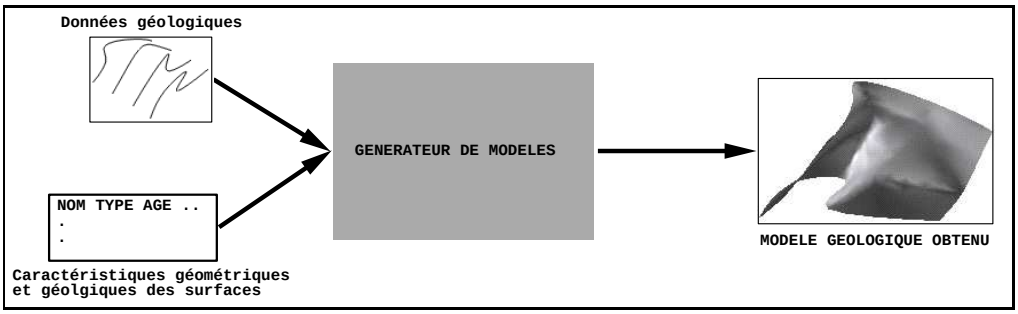

FIG. 5.1 – *Exemple d'un modeleur géologique de type boîte noire*

5.2 Principe du G.E.M.G

Le G.E.M.G est basé sur la construction d'un modèle à partir d'un ensemble de données variées, en se basant sur les caractéristiques géométriques

et géologiques des surfaces composant le modèle. L'utilisateur peut intervenir à tout moment au cours de la construction du modèle pour contrôler et modifier les surfaces construites. La nouveauté apportée par notre méthode est que le mécanisme de construction du modèle géologique est accessible, dans sa totalité à l'utilisateur ; cela permet au géologue de guider, grâce à son expérience géologique, la construction du modèle géologique.

5.2.1 Données géologiques

Dans le domaine de l'exploration pétrolière, les données géologiques les plus utilisées pour la modélisation de surfaces géologiques sont les données sismiques telles que les semis de points et les courbes polygonales digitalisées (cf chapitre 4). D'autres données telles que celles provenant des puits peuvent compléter les informations fournies par les données sismiques. On désigne par puits un forage profond, avec carottage des couches traversées. Des mesures de propriétés géométriques (pendage[1] ou azimut[2] $\cdots$) ou physiques (porosité, perméabilité, $\cdots$) peuvent être réalisées sur les surfaces géologiques traversées lors du forage. Les données de puits sont plus précises mais moins denses que celles obtenues par les campagnes sismiques, il est donc difficile de baser la construction des surfaces sur les seules données de puits. En effet, ces dernières interviendront lors de l'ajustement final des surfaces géologiques en prenant en compte par exemple le pendage mesuré au niveau des puits (cf figure 5.2).

En résumé, les données initiales prises en compte par notre générateur de modèles géologiques seront seulement les semis de points et les courbes polygonales provenant par exemple de la digitalisation de sections sismiques.

5.2.2 Caractéristiques des surfaces du modèle

Bien que les surfaces géologiques soient mathématiquement identiques, d'un point de vue géologique on peut les classer en plusieurs catégories selon plusieurs critères géologiques et/ou géométriques tels que le mode et l'environnement de dépôt des couches, la priorité d'intersection des surfaces géologiques $\cdots$. Parmi ces critères nous avons choisi d'utiliser le critère de la

1. Le pendage d'une surface est l'angle que fait la ligne de plus grande pente avec l'horizontal

2. L'azimut est l'angle que fait le nord magnétique avec la ligne d'intersection d'une surface avec le plan horizontale

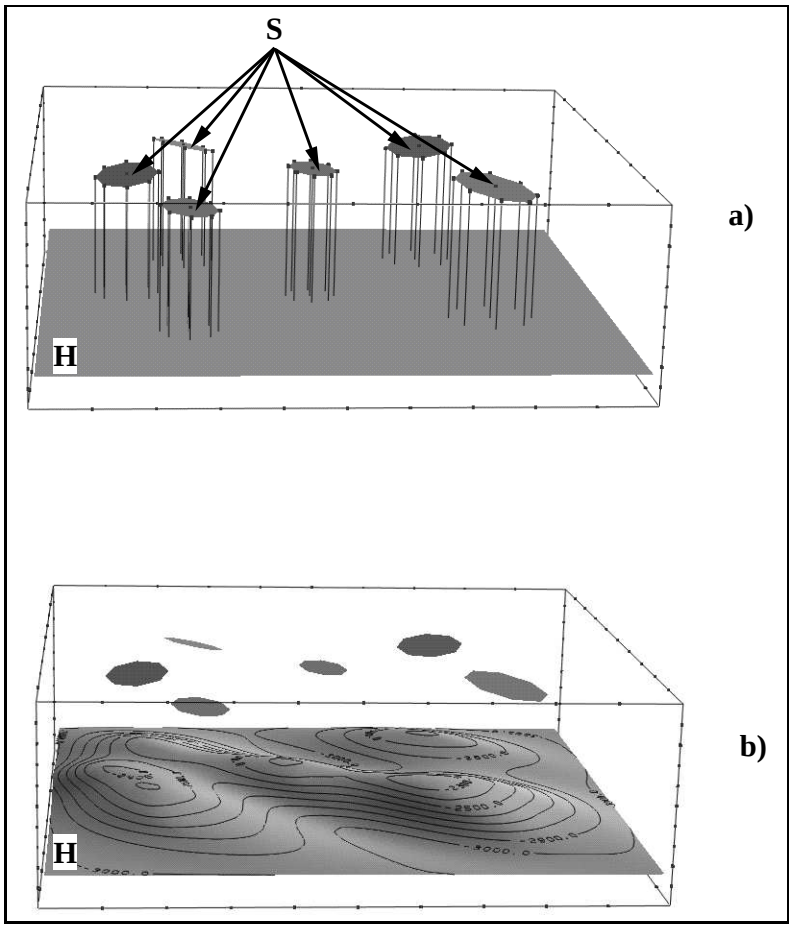

FIG. 5.2 – *Contrôle du pendage de la surface H par la surface S, a) Mise en place de la contrainte de pendage sur la surface H à partir de la surface S, b) Après interpolation par DSI de H, le pendage de la surface se retrouve localement identique à celui de S.*

priorité d'intersection car les zones d'intersections des surfaces géologiques présentent des caractéristiques spéciales. Un exemple a été présenté dans le chapitre 4. Cet exemple concerne la zone de contact entre un horizon géologique et une faille.
En 1990, J. Hamburger ([23]) a proposé un critère de découpage des objets géologiques en $2D$, selon la priorité d'intersection. Dans la suite de ce chapitre nous nous proposons d'étendre ce découpage en $3D$. Ainsi, nous

pouvons distinguer (cf figure 5.3) :

1. *Les surfaces d'érosion*
   Il s'agit de surfaces qui peuvent couper toutes celles qui leurs sont antérieures.

2. *Les surfaces failles*
   Ce sont des surfaces qui peuvent couper celles qui leur sont chronologiquement antérieures et provoquer un décalage des deux compartiments de la faille (cf chapitre 4).

3. *Les surfaces dômes*
   Il s'agit de surfaces qui peuvent couper toutes celles qui ne sont pas des surfaces d'érosion, sans tenir compte de la chronologie de dépôt des couches.

4. *Les surfaces horizons*
   Dans cette catégorie on place les surfaces qui correspondent aux toits ou aux bases des couches géologiques. Ces surfaces ne peuvent couper que les surfaces horizons chronologiquement antérieures.

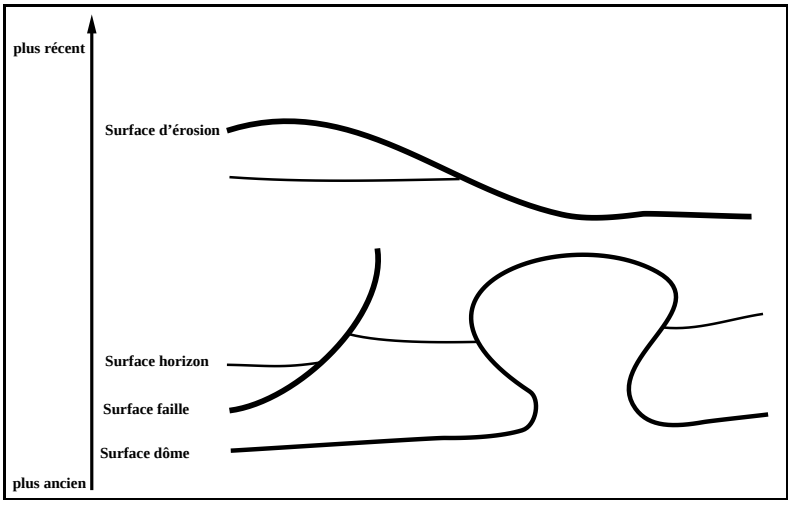

FIG. 5.3 – *Exemple des différentes catégories de surfaces géologiques*

Remarque :

Le découpage proposé est simple mais on assiste parfois à des ambiguïtés qui ne peuvent être levées sans l'intervention de l'utilisateur. Un exemple sera donné dans les paragraphes suivants.

Pour générer le modèle géologique, l'utilisateur doit fournir les caractéristiques des surfaces qui composent son modèle. La figure 5.4 montre un exemple de ces caractéristiques :

- La surface H1 est un horizon d'âge JURASSIQUE ($\simeq$ 200 à 140 Millions d'années) qui sera construit à partir d'un semis de points (ou VSET).

- La surface H2 est une faille d'âge CRÉTACÉ ($\simeq$ 140 à 60 Millions d'années), et les données qui lui correspondent sont des courbes polygonales.

- La surface H3 est une surface d'érosion d'âge QUATERNAIRE (1.8 Millions d'années à l'époque actuelle), elle sera construite à partir d'un semis de points.

V_{F1}	VSET	CRETACEOUS	FLT
V_{TJU}	VSET	JURASSIC	HRZ
V_{L1}	VSET	CRETACEOUS	FLT
V_{S1}	VSET	CRETACEOUS	FLT
V_{L2}	VSET	CRETACEOUS	FLT
V_{F2}	VSET	CRETACEOUS	FLT
V_{TGR}	VSET	TRIASIC	HRZ
...			

FIG. 5.4 – *Exemple du fichier ascii fourni par un géologue*

Une fois les données géologiques fournies et les caractéristiques des surfaces, composant le modèle à construire, spécifiées par le géologue, le générateur automatique se charge de construire le modèle géologique.

5.2.3 Génération du modèle géologique

La génération d'un modèle géologique se décompose en deux étapes :

- Construction des différentes surfaces géologiques qui composent le modèle.
- Calcul de l'intersection entre les différentes surfaces.

5.2.3.1 Construction des surfaces géologiques

Le mode de construction automatique de chaque surface géologique dépend de son type et des données qui lui correspondent. Par exemple, un dôme de sel peut être assimilé à un cylindre déformé, nous avons donc adopté la démarche suivante pour construire un dôme de sel à partir d'un semis de points :

1. Calcul des trois directions principales d'inertie du semis de points. Ces trois directions sont désignées par le triplet (u, v, w) (cf figure 5.5a).
2. Construction d'un cylindre, qui contient tous les points du semis. La génératrice du cylindre correspond à l'axe u de plus grande inertie, et la base correspond à l'enveloppe convexe de la projection des points dans le plan (v, w) (voir figure 5.5b).
3. Ajustement du cylindre construit, au semis de points en utilisant la contrainte *Fuzzy-Control-Point* présentée dans le chapitre 2. Nous obtenons alors un dôme de sel (5.5c).

Le résultat obtenu peut être contrôlé par l'utilisateur de deux manières :

- *Contrôle qualitatif*
  L'utilisateur se base sur ses connaissances de la surface construite et sur son expérience géologique pour valider le résultat obtenu. Un ensemble d'outils a été mis en place pour assister l'utilisateur dans son contrôle. Par exemple sur la figure 5.6, après calcul de la courbure principale de la surface H ([52][16]), des zones de cette surface se distinguent par leur forte courbure. Cela peut être expliqué de plusieurs manières dont par exemple :
  1. Un phénomène de compression locale de la surface suite à l'avènement de phénomènes géologiques. La forte courbure locale de la surface est donc géologiquement justifiée.

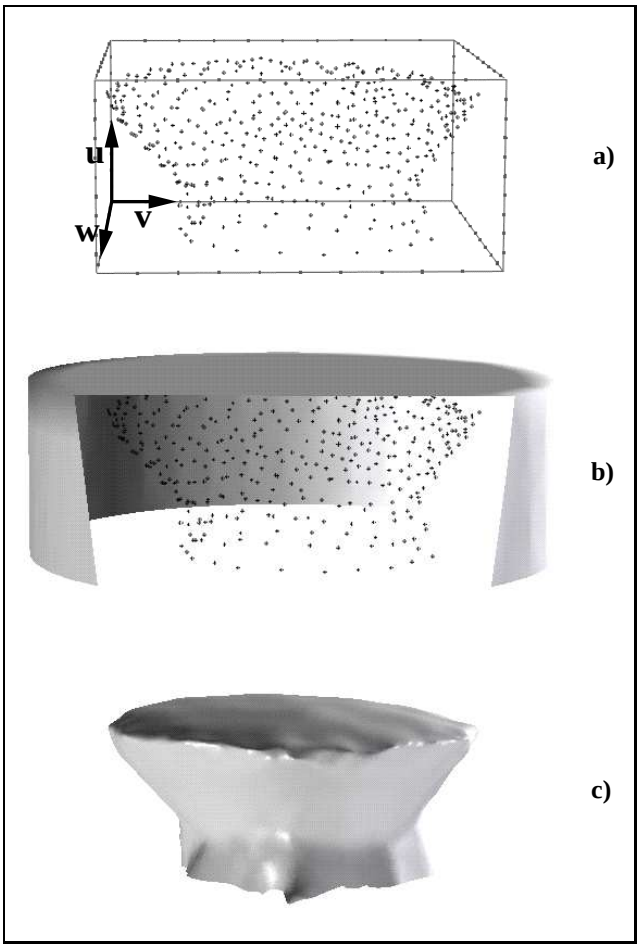

FIG. 5.5 – *Les différentes étapes de construction d'un dôme de sel à partir d'un semis de points, a) Les données correspondant au semis de points, b) Construction du cylindre qui contient tous les points, c) Ajustement avec la méthode DSI du cylindre au semis de points*

2. Certaines données initiales erronées peuvent produire une forte courbure locale de la surface. Pour remédier à ce problème, l'utilisateur pourra modifier la force d'attraction de la surface par ces données erronées en modifiant les poids locaux des contraintes associées aux points erronés (cf chapitre 2).

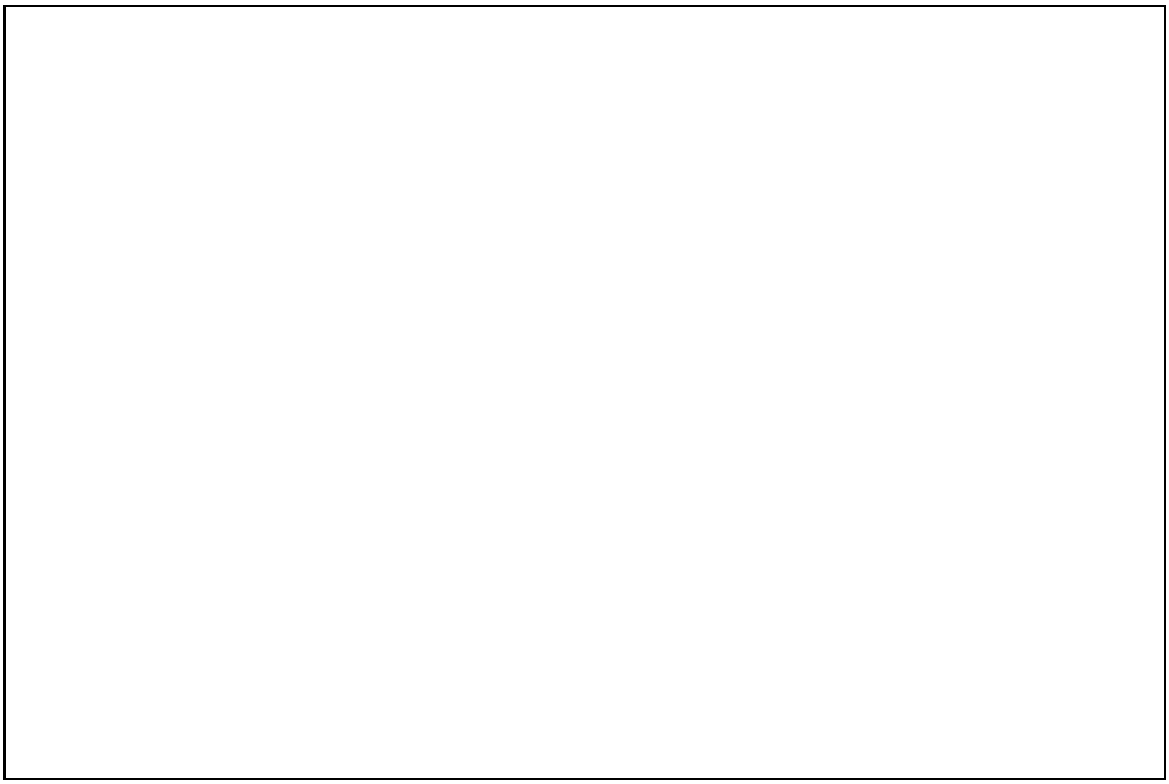

FIG. 5.6 – *Cartographie de la courbure principale d'une surface triangulée. L'échelle des couleurs permet d'estimer la valeur de la courbure*

– *Contrôle quantitatif*

Parmi les méthodes proposées pour le contrôle quantitatif du résultat obtenu, nous pouvons citer l'erreur d'ajustement de chaque surface du modèle aux données qui lui correspondent (cf chapitre 2).
L'utilisateur peut réduire cette erreur en modifiant localement le maillage de la surface.

Un deuxième exemple de construction automatique de surfaces géologiques a été présenté dans le chapitre 4. Cet exemple concerne les surfaces de failles.

Lorsque toutes les surfaces du modèle ont été automatiquement construites par le générateur de modèles géologiques et contrôlées par l'utilisateur, nous obtenons le *modèle géologique initial* qui contient l'ensemble des surfaces du modèle.

La deuxième étape de l'algorithme du générateur de modèles géologiques consiste à calculer les intersections entre les différentes surfaces.

5.2.3.2 Calcul des intersections entre les surfaces

Le calcul d'intersections entre les surfaces du modèle géologique prend en compte les règles d'intersections proposées dans le paragraphe 5.2.2. Par exemple, une faille peut couper tous les horizons qui lui sont chronologiquement antérieurs sauf les dômes de sel. Afin de respecter le principe général du G.E.M.G qui consiste à permettre à l'utilisateur d'intervenir à tout moment au cours de la construction du modèle, nous avons mis en place un système de graphe d'intersection dont le principe est le suivant : *Toutes les surfaces du modèle seront positionnées sur le graphe en fonction de leurs âges géologiques et en se basant sur les règles de priorité décrites dans le paragraphe 5.2.2.* Par exemple, si un horizon et une faille ont le même âge, c'est la faille qui sera considérée comme plus récente et pourra couper l'horizon. L'intervention de l'utilisateur consiste à repositionner les différentes surfaces géologiques dans le graphe d'intersection. L'utilisateur sera appelé aussi à lever des ambiguïtés, par exemple si une faille et une surface d'érosion ont le même âge, l'utilisateur devra modifier interactivement le graphe d'intersection.

Lorsque toutes les ambiguïtés sont levées, le G.E.M.G procède au calcul des intersections entre les différentes surfaces selon l'ordre donné par le graphe d'intersection. Ce calcul des intersections s'accompagne, dans le cas d'un contact Faille-Horizon, de la mise en place de contraintes géométriques décrites dans le chapitre 4, je rappelle que ces contraintes vont permettre :

- d'obliger le bord de l'horizon à glisser le long de la faille, grâce à la contrainte $OnTsurf$,
- de contrôler la distance entre les bords de la faille, grâce à la contrainte $VecLink$,
- de contrôler l'extension maximale des lèvres de la faille grâce à la contrainte $OnBorder$.

L'ensemble des surfaces géologiques obtenues après calcul des intersections forme le *modèle final*. Ce dernier pourra être localement édité par l'utilisateur selon le principe présenté dans le paragraphe suivant.

5.2.4 Édition interactive du modèle géologique

Nous disposons à cette étape, du modèle géologique final et du graphe d'intersection correspondant. L'utilisateur pourra repositionner , dans le graphe d'intersection, des surfaces du modèle en fonction de règles géologiques spécifiques correspondant, par exemple, à des études précises sur certaines surfaces du modèle. La mise à jour du modèle consiste à recalculer les intersections des surfaces qui se trouvent directement ou indirectement intéressées par cette modification. Par exemple sur la figure 5.7 si on repositionne la surface $F2$, les surfaces $S3$ et $S4$ pourront être coupées par $F2$. Le G.E.M.G devra donc recalculer l'intersection ($S3$,$F2$) et ($S4$,$F2$). Cette partie est en cours de développement, elle ne sera donc pas détaillée.

5.2.5 Implémentation du G.E.M.G dans le système GOCAD

La mise en place du générateur de modèles géologiques a nécessité le développement des structures de stockage des informations concernant les différentes surfaces du modèle géologique, ainsi que les données initiales qui leur correspondent. Chaque modèle géologique est considéré comme une instance de la classe GOCMODEL qui dérive, au sens de la programmation orientée objet (cf chapitre 2), de la classe de base qui définit tous les objets. La classe GOCMODEL introduit de nouvelles données propres à cette classe, ainsi que des fonctions membres permettant de manipuler les objets du modèle. Parmi les données spécifiques à la classe GOCMODEL nous retrouvons :

- une liste de tous les objets composant le modèle,

- une structure qui permet d'associer à chaque surface S du modèle, les données V qui ont permis de la construire. Toute modification de V sera automatiquement répercutée sur H grâce au système de superviseur de projets présenté dans le chapitre 2.

Le paragraphe suivant présente les résultats obtenus sur un modèle réel.

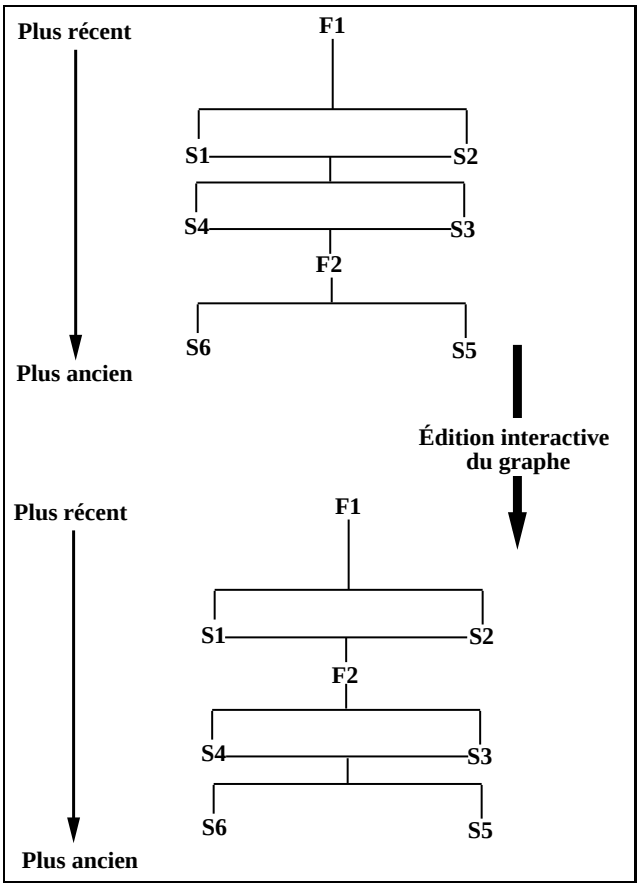

FIG. 5.7 – *Édition interactive d'un graphe d'intersection*

5.3 Étude de cas

Dans ce paragraphe, nous allons étudier la construction d'un modèle géologique contenant 1 horizon et 12 failles. Pour ne pas alourdir l'exposé nous présenterons les résultats obtenus sur la construction d'un horizon et d'une faille.

5.3.1 Modèle initial

Toutes les surfaces du modèle sont connues au travers de semis de points. Nous disposons aussi d'un fichier ASCII, fourni par le géologue, qui contient

un supplément de renseignements sur les surfaces du modèle (cf figure 5.8). Il s'agit de l'âge des surfaces, de leurs types, $\cdots$.

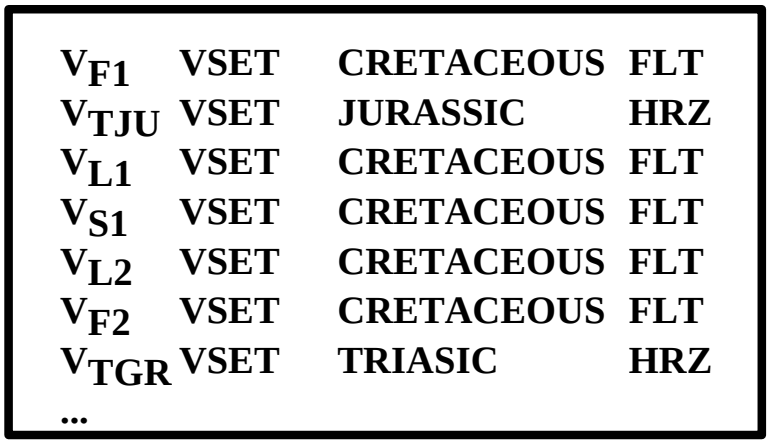

FIG. 5.8 – *Fichier de description des surfaces du modèle*

Pour ce modèle, les surfaces seront toutes construites selon la même méthode. Le principe de cette méthode, est présenté pour l'horizon TJU :

- Le semis de points (V_{TJU}) correspondant à TJU est représenté sur la figure 5.9 par les points rouges. Nous calculons les trois directions principales d'inertie (u, v, w).
- Après avoir projeté les points de V_{TJU} sur le plan (u, v), nous calculerons la ligne polygonale L_{TJU} correspondant à l'enveloppe convexe des points projetés (cf figure 5.9), selon le procédé développé par Y. Chipot et P. Lavest ([31]). Cette enveloppe convexe est matérialisée sur la figure 5.9 par une ligne polygonale jaune.
- Une surface TJU sera calculée à partir de la courbe L_{TJU}, en utilisant la procédure de construction de surface à partir d'une courbe polygonale fermée. L'étape suivante consiste à ajuster la surface créée au semis de points.
- La surface TJU sera ajustée aux points du semis en utilisant la contrainte $Fuzzy - Control - Point$ présentée dans le chapitre 2.
- Pour un meilleur ajustement de la surface TJU au semis de points V_{TJU}, nous avons mis en place un procédé automatique de densification

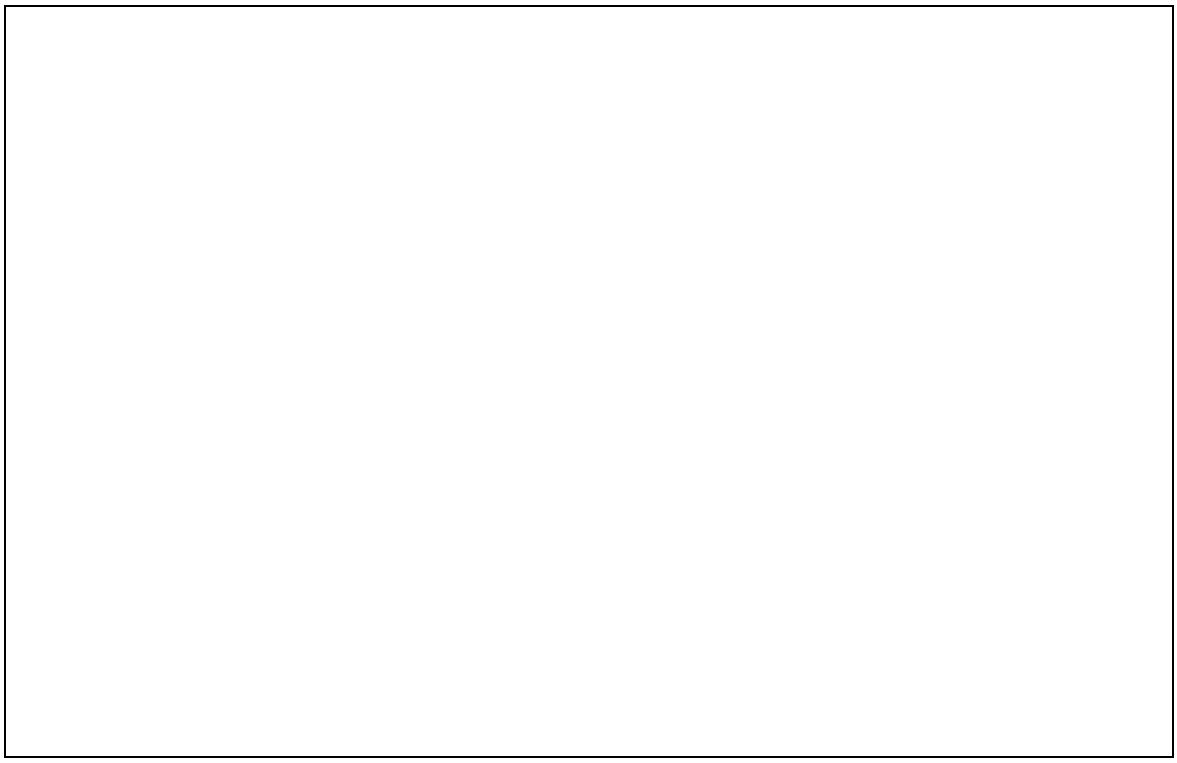

FIG. 5.9 – *Semis de points et enveloppe convexe*

locale du maillage de la surface. Ce procédé est basé sur le calcul de l'erreur d'ajustement d'une surface à un semis de points (cf chapitre 2). Ce procédé se décompose en trois étapes :

1. Nous calculons $Vmin$ et $Vmax$ qui sont respectivement les erreurs minimales et maximales d'ajustement de la surface au semis de points.
2. Tous les triangles au niveau desquels l'erreur d'ajustement est non nulle seront subdivisés en quatre triangles. Cela permet d'augmenter localement la souplesse de la surface, l'ajustement aux données sera donc plus facile à obtenir.
3. La surface sera ensuite interpolée à l'aide de DSI.

Le procédé sera répété itérativement tant que les erreurs minimales et maximales ne seront pas nulles. Pratiquement, plus le nombre des tri-

angles de la surface à ajuster est important plus le volume de calcul engendré pour la mise à jour des contraintes et le lissage de la surface, est important (cf figure 5.10). Les tests de performance effectués montrent qu'au bout d'un certain nombre d'itérations (proche de 3), le gain de précision obtenu par la densification du maillage de la surface est négligeable en regard de la lenteur des procédures de densification du maillage et de la mise à jour des contraintes. Il est donc souhaitable que l'utilisateur fixe une valeur de l'erreur d'ajustement pour laquelle la surface peut être supposée correcte.

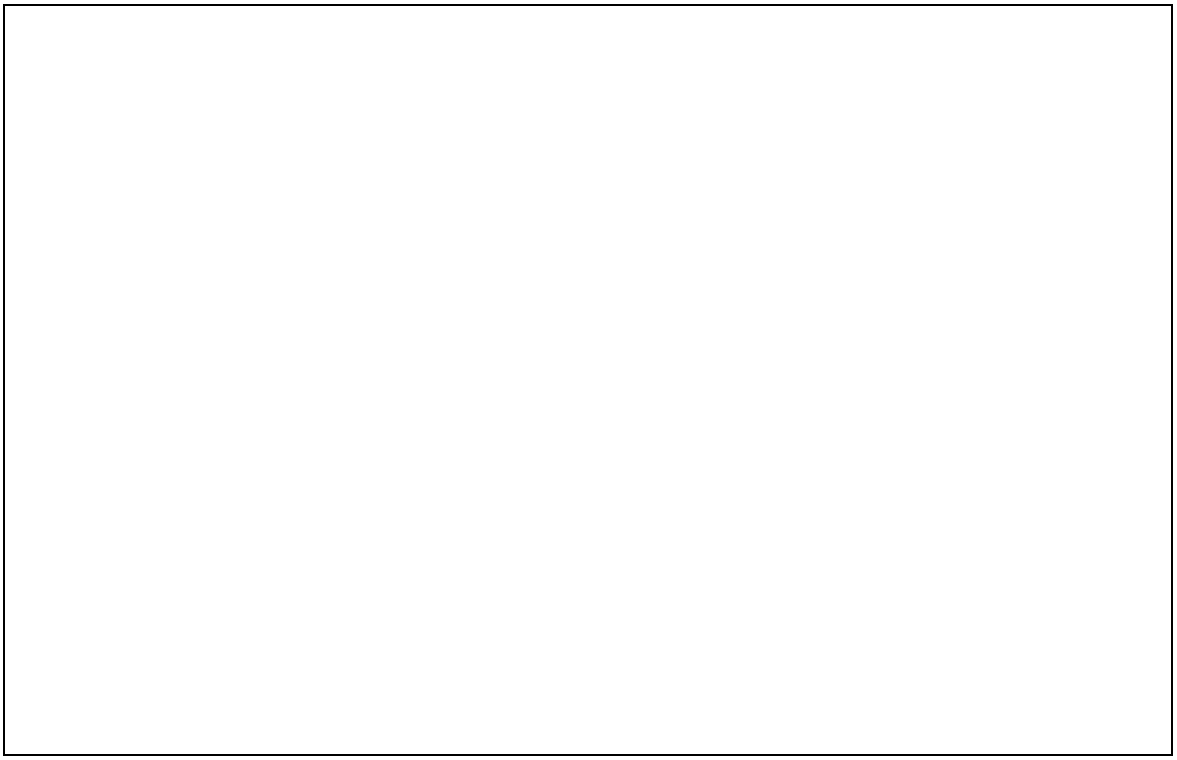

FIG. 5.10 – *Surface TJU après densification du maillage*

La deuxième partie de la construction du modèle consiste à calculer les intersections entre les différentes surfaces géologiques.

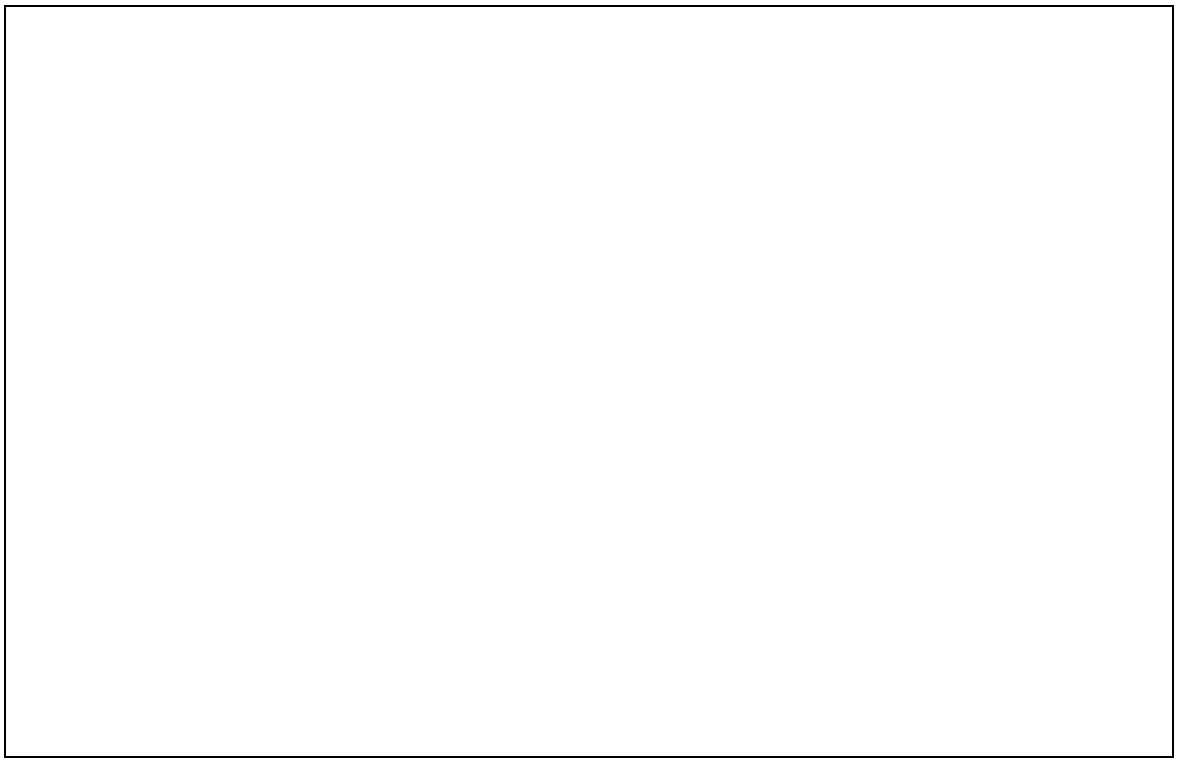

FIG. 5.11 – *Surface TJU*

5.3.2 Calcul des intersections entre les surfaces

Comme nous l'avons dit précédemment, un graphe de priorités d'intersection peut être mis en place pour permettre à l'utilisateur de détecter certaines incohérences dans le modèle initial. Dans notre cas, ce graphe sera simple car le modèle ne comporte que deux types de surfaces : des failles et des horizons. Le calcul d'intersections revient donc à couper, en utilisant la procédure $TSURF\_Cut\_by\_Tsurf()$ ([24]), chacun des deux horizons géologiques par toutes les failles, les contraintes $OnTsurf$ et $OnBorder$, décrites dans le chapitre 4, seront mises en place automatiquement au niveau des lèvres des failles. Je rappelle que l'utilisateur peut intervenir pour contrôler les déplacements des lèvres des failles en mettant en place la contrainte $VecLink$ décrite dans le chapitre 4. La figure 5.12 montre le résultat obtenu après le découpage de l'horizon TJU par l'ensemble des surfaces.

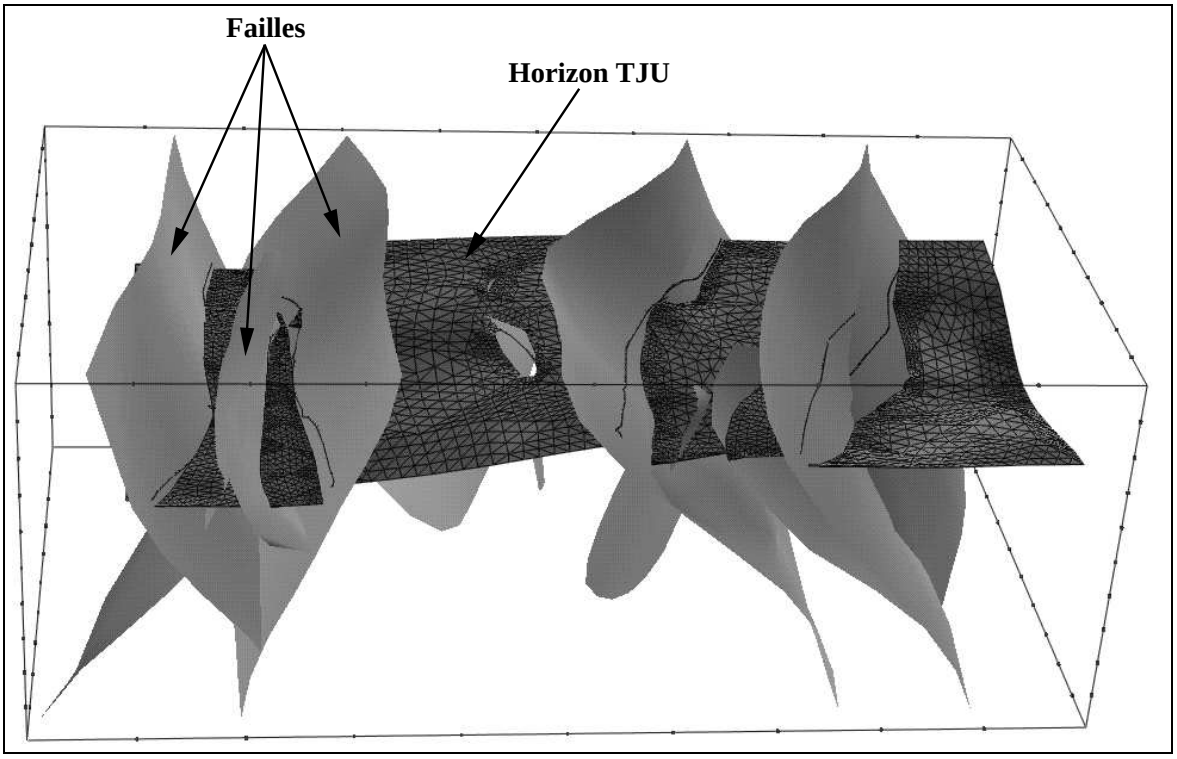

FIG. 5.12 – *Modèle final*

5.4 Conclusion

L'originalité de l'approche adoptée réside comme je l'ai signalé dans son interactivité. En effet, de nombreux outils d'édition des surfaces sont proposés dans le logiciel GOCAD pour permettre à l'utilisateur d'intervenir à tout moment au cours de la construction du modèle. Le principe de ce prototype étant général, il est possible de l'implémenter facilement dans un autre système informatique.

Chapitre 6

Conclusions

Depuis son lancement en 1989, le projet GOCAD tente d'apporter des solutions dans le domaine de la modélisation géométrique en trois dimensions et notamment de résoudre les problèmes liés au monde de la géologie. Pour répondre le plus efficacement possible aux problèmes caractéristiques de la modélisation d'objets géologiques, je me suis proposé de mettre au point *un prototype de générateur automatique de modèles géologiques*. Pour ce faire j'ai été amené à développer :

- Une méthode rapide et efficace pour le calcul des courbes d'isovaleurs. L'approche que j'ai proposée prend en compte le fait que pour une surface, les informations géométriques et physiques sont connues au niveau des nœuds de cette surface.

- Des contraintes géométriques DSI, afin de modéliser avec un maximum de précision le contact entre une faille et un horizon géologique.

A partir de ces outils ainsi que d'autres outils existant dans le système GOCAD, j'ai proposé un prototype de générateur automatique de modèles géologiques qui se base sur des règles géologiques précises pour construire les surfaces du modèle. Ce prototype permet à l'utilisateur d'intervenir à tout moment au cours de la construction automatique du modèle géologique.

L'utilisateur peut donc gouverner la construction du modèle géologique.

Le travail de cette thèse se présente sous trois aspects : Le premier est géologique et consiste à exprimer géologiquement le problème à résoudre. Le deuxième est mathématique et consiste à exprimer le problème sous forme de relations mathématiques. Par exemple, pour modéliser la relation faille-horizon j'ai exprimé cette relation géologique sous forme d'équations liées à DSI. Le troisième aspect est informatique puisque l'ensemble des notions développées furent intégrées au sein du logiciel GOCAD.

Le travail effectué présente quelques points faibles qu'il sera souhaitable d'étudier dans le futur :

- Il serait intéressant de pouvoir prendre en compte, lors de la modélisation de la relation faille-horizon, le comportement mécanique (coefficient de cisaillement,$\cdots$) de la faille et de l'horizon, cela permettra de calculer la direction de déplacement des compartiments de la faille.
- Le principe du prototype de générateur automatique de modèles géologiques étant fixé, il serait intéressant :
  - d'optimiser la procédure de densification du maillage afin de réduire l'erreur d'ajustement, des surfaces construites, aux données d'origine,
  - de développer l'éditeur du graphe d'intersection.

Enfin j'espère que la lecture de ce mémoire permettra au lecteur d'avoir une idée sur l'approche de la modélisation des surfaces géologiques adoptée dans GOCAD.

Annexe

La conception de la base de données du système GOCAD, réalisée en langage C, a été fortement influencée par les principes de la programmation orientée objet. Une notion *d'héritage virtuel* a été installée, pour éviter la duplication des fonctions. Ainsi tous les objets (courbes,surfaces,solides,$\cdots$) dérivent d'une racine appelée *GOBJ*. La structure *GOBJ_t* qui lui correspond est la suivante:

```
typedef struct GOBJ_t
  {
    STRING_c name;
    GSTYLE_c p_gstyle;
    void (*Delete)(GOBJ_t*);
    void (*loadBin)(GOBJ_t*,FILE*);
    void (*saveBin)(GOBJ_t*,FILE*);
    void (*loadAscii)(GOBJ_t*,FILE*);
    void (*saveAscii)(GOBJ_t*,FILE*);
    ...
  }GOBJ_t;
```

- *name* est une chaîne de caractères qui représente de façon univoque un objet.
- *p_gstyle* est un pointeur vers une structure qui définit les attributs graphiques de l'objet.
- *Delete* est une fonction virtuelle (au sens de la programmation orientée objet) qui permet de détruire l'objet et de le retirer de la base de données.
- *loadBin* est une fonction virtuelle qui charge un objet à partir d'un fichier au format binaire.

- *saveBin* est une fonction virtuelle qui sauvegarde un objet dans un fichier au format binaire.

- *loadAscii* est une fonction virtuelle qui permet de charger un objet à partir d'un fichier au format ASCII.

- *saveAscii* est une fonction virtuelle qui sauvegarde un objet dans un fichier au format ASCII.

La structure ci-dessus permet de décrire des objets graphiques tels que les caméras, et les objets géométriques tels que les surfaces. Dans la suite, je ne présenterai que les objets géométriques.

Notion du point du vertex et du vset

Dans GOCAD un *point* définit les coordonnées dans l'espace. La structure qui lui correspond est :

```
typedef struct POINT3_t
  {
    float x,y,z ;
  }POINT3_t;
```

Cette structure est utilisée pour définir des vecteurs ou des coordonnées d'un point dans un repère.
Un *vertex* est une sous-classe du point, elle caractérise la position dans l'espace d'un point appartenant à un objet.

```
typedef struct VRTX_t
  {
    POINT3_t pos ;
    long movable ;
    ...
  }VRTX_t;
```

- *pos* ce champ définit les coordonnées du vertex,

- *movable* définit le degré de liberté du point selon les directions Ox, Oy, Oz.

Un *vset* est un objet qui regroupe l'ensemble des vertex. La structure correspondant à un vset est la suivante :

```
typedef struct VSET_t
  {
    DERIVED_FROM(GOBJ_t);
    long nb_vrtx;
    SETK_c p_vrtx_setk;
    BOX3_t box;
    ...
  }VSET_t
```

- *DERIVED_FROM(GOBJ_t)* est une macro-définition qui spécifie qu'un vset hérite de toutes les caractéristiques d'un objet.
- *nb_vrtx* donne le nombre de vertex d'un vset.
- *p_vrtx_setk* est une liste qui contient tous les vertex d'un vset,
- *box* est la boîte qui englobe tous les vertex d'un vset.

Plusieurs objets peuvent partager des vertex. Une notion d'importation et d'exportation a donc été installée ([36]).

Notion de l'atome et de l'atomique

Un objet géométrique peut être décrit par un vset qui correspond aux nœuds. Cependant, le maillage ne peut être décrit correctement que par une structure qui permet à chaque nœud d'accéder à ses voisins, d'où l'introduction de la notion d'atome. Il s'agit d'un nœud connecté à un certain nombre de voisins directs appelés *satellites*. La structure correspondante est la suivante:

```
typedef struct ATOM_t
  {
    ATOMIC_t *p_owner;
    RECORD_t *p_record;
    ATOM_t **p_sat;
    VRTX_t *p_vrtx;
    CNSTRNODE_t *p_cnstrnode;
    ...
  }ATOM_t;
```

- *p_owner* permet à un atome d'accéder à l'objet auquel il appartient,
- *p_record* est un pointeur vers une structure qui permet d'attacher des propriétés physiques à un atome,
- *p_sat* désigne la liste des satellites de l'atome,
- *p_vrtx* permet à un atome d'accéder au vertex qui contient les coordonnées de l'atome,
- *p_cnstrnode* est une liste chaînée de toutes les contraintes attachées à un atome.

L'atome est la structure essentielle de la base de données car il contient les caractéristiques géométriques et topologiques, de tous les nœuds d'un objet. Un atomique contient tous les atomes d'un objet.

```
typedef struct ATOMIC_t
  {
    DERIVED_FROM(VSET_t);
    SET_c p_atom_set;
    ...
  }ATOMIC_t;
```

- *DERIVED_FROM(VSET_t)* signifie que l'objet atomique hérite des propriétés d'un vset,
- *p_atom_set* est la liste de tous les atomes qui composent l'objet atomique.

On peut déduire que tout objet géométrique (ligne,surface,solide) peut être entièrement défini par un réseau de type atomic. Cependant, ce dernier ne permet pas de représenter toutes les caractéristiques de ces objets. Les notions de courbe, de surface et de solide ont donc été introduites.

Notion du segment et de la courbe

L'élément de base d'une ligne, dans le système GOCAD, est le segment. Il est défini par la structure suivante :

```
typedef struct SEG_t
  {
```

```
    SEG_t **p_seg;
    ATOMPLINE_t *p_atom[2];
    ...
  }TRGL_t;
```

- *p_seg* est la liste des segments voisins,
- *p_atom* est un tableau des deux nœuds qui forment le segment.

Une courbe est représentée par la structure suivante :

```
typedef struct PLINE_t
  {
    DERIVED_FROM( ATOMIC_t );
    SET_t *p_simplex_set;
    ...
  }
```

- *DERIVED_FROM(ATOMIC_t)* est une "macro-définition" qui signifie qu'une ligne est un objet atomique.
- *p_simplex_set* est une liste de tous les segments de la courbe.

Notion du triangle et de la surface

L'élément de base d'une surface est le triangle dont la structure est la suivante :

```
typedef struct TRGL_t
  {
    TRGL_t **p_trgl;
    ATOMTSURF_t *p_atom[3];
    POINT3_t *bezier;
    ...
  }TRGL_t;
```

- *p_trgl* est un pointeur vers les triangles voisins,
- *p_atom* est un tableau des trois nœuds qui forment le triangle,
- *bezier* est un pointeur vers les points de controle de la facette de Bézier correspondante.

Une surface est représentée par la structure suivante :

```
typedef struct TSURF_t
  {
    DERIVED_FROM( ATOMIC_t );
    SET_t *p_simplex_set ;
    ...
  }
```

- *DERIVED_FROM(ATOMIC_t)* est une "macro-définition" qui montre qu'une surface est un objet atomique.
- *p_simplex_set* est la liste de tous les triangles de la surface.

Bibliographie

[1] T. Aït Ettajer, J.L. Mallet, *Automatic Modelling of Geological Model* Expanded Abstracts of Papers of the European Association of Petroleum Geoscientists and Engineers, 6th Conference and Technical Exhibition, Vienne 6-10 Juin 1994.

[2] T. Aït Ettajer, J.L. Mallet, *Génération automatique de modèles géologiques* Réunion des Sciences de la Terre, Nancy, 1994

[3] T. Aït Ettajer, J.L. Mallet, *Modelling the Fault-Horizon relationship* Extended Abstracts of Papers of the European Association of Geoscientists and Engineers, 7th Conference and Technical Exhibition, Glasgow 29 May-2Juin 1995.

[4] R. Banks, *Modeling Geological and Geophysical Surfaces*, GEOBYTE, Octobre 1990.

[5] R. Banks, *Contouring algorithms*, GEOBYTE, Octobre 1991.

[6] J.D. Boissonnat, *Shape reconstruction from planar cross-sections.* Computer Vision, Graphics, and Image Processing, num 44, pp 1-29, 1988.

[7] T. Budd, *Introduction à la programmation par objets*, Version française, Addison-Wesley Publishing Company, Inc. 1991.

[8] S. Collet Cases, *Gestion automatique des relations entre modules: Application à la C.A.O*, Thèse I.N.P.L, 1995.

[9] Y. Chipot, *Modélisation par éléments triangulaires*, Rapport de DEA de l'Université de NancyI, 1987.

[10] Y. Chipot, *Génération et modification des surfaces triangulées*, Thèse INPL, Nancy I, 1991.

[11] Y. Chipot, *Smoothing a surface with DSI*, Revue Internationale de CFAO et d'Infographie, vol. 4, num. 2.

[12] G. Christakos, *Random Field Models in Earth Sciences*, Academic Press, 1992.

[13] R. Cognot, T. Aït Ettajer, J.L. Mallet, *Modelling continuity throw discontinuities*, Article soumis en 1994.

[14] R. Cognot, *Définition, manipulation et interpolation de propriétés physiques*, Thèse I.N.P.L en cours

[15] R. Cognot, S. Cases Collet, J.L. Mallet, *Atomic-Controller model for DSI*, Article soumis en 1994.

[16] M. P. Do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice Hall, 1976.

[17] G. Dresen, U. Gwildis, Th. Kluegel *Numerical and analogue modelling of normal fault geometry*, The geometry of normal faults, Geological Society Special Publication N^o 56, pp 207-217, 1991.

[18] G. Farin, *Triangular Bernstein-Bézier patches*, Computer Aided Geometric Design, Vol. 3, No. 2, pp 83-126, 1986.

[19] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, 1988.

[20] T.R. Fisher, R.Q. Wales, *Rational splines and multidimensional geologic modeling*, Lecture Notes in Earth Sciences 41, Springer-Verlag 1992

[21] M. Gidon, *Les structures tectoniques*, Editions du B.R.G.M, 1987.

[22] J.A. Gregory, *Curves and Surfaces for Computer Aided Geometric Design*, Computer Graphics and Image

[23] J. Hamburger, *Intelligence artificielle et géologie. Interprétation géohistorique et diagnostic de conception* Thèse Université de Paris VI, Paris, 1990.

[24] Y. Huang, *Modélisation et manipulation des surfaces triangulées*, Thèse INPL, Nancy, 1990.

[25] G. Henry, *Géophysique des bassins sédimentaires* Éditions Technip, 1994

[26] M. Ismail, *Méthodes de construction d'une géométrie à exploiter dans le cadre d'une planification à court terme des mines souterraines*, Thèse Ecole des Mines de Paris, 1992.

[27] R.J. Knipe, *Faulting processes and fault seal*, Structural and Tectonic Modelling and its Application to Petroleum Geology, Norwegian Petroleum Society, 1992.

[28] P. Laffitte, *Traité d'informatique géologique*, MASSON & CIE, 1972. Processing. Vol. 13. 1980.

[29] K. Lamboglia, *Modélisation volumique de surfaces non-manifold*, Thèse I.N.P.L, 1994.

[30] S. Lanoé, *Modélisation sur ordinateur d'un compartiment géologique et applications pédagogiques*, Thèse Ecole des Mines de Paris, 1981.

[31] P. Lavest, Y. Chipot, *Builiding Complex Horizons for 3D seismics* Program and Expanded Abstract, Annual SEG meeting, Washington, 1993.

[32] M. Léger, J.M. Morvan, M. Thibaut, *Least-Squares Optimization of Fault Surfaces Using the Rogid Block Approximation* Program and Expanded Abstract, Annual SEG meeting, Washington, 1993.

[33] M. Léger, J.M. Morvan, M. Thibaut, *Least-Squares Optimization of Thread Surfaces*, Curves and Surfaces II, 1991.

[34] C. H. Scholz, *The Mechanics of Earthquakes and Faulting*, Cambridge University Press, 1990.

[35] M. Thibaut, *Géométrie des surfaces de faille et dépliage 3D. Méthodes et applications*, Thèse de doctorat de l'Université Joseph Fourrier-Grenoble1, Décembre 1994.

[36] P. Le Mélinaire, *Modélisation des relations géométriques par la méthode DSI -Application à la Géologie-*, Thèse INPL, Nancy, 1992.

[37] T. Lyche, L. L. Schumaker, *Mathematical methods in computer aided geometric design*, Academic Press, 1989.

[38] J.L Mallet, *Présentation d'un ensemble de méthodes et techniques de la cartographie automatique numérique*, Ann.Ec.Nat.Sup.Geo, N^o4 Oct 1974.

[39] J.L. Mallet, *Automatic contouring in presence of discontinuities*, Geostatistics for natural resources characterization, Vol 122, Part 2, pp. 669-677, 1984.

[40] J. L. Mallet, *Three-Dimensional Graphic Display of Disconnected Bodies*, Mathematical Geology, Vol 20, No 8, 1988.

[41] J. L. Mallet, P. Le Melinaire, *Modelling complex faults. The* GOCAD *approach* EAEG meeting 54, Paris, 1992.

[42] J. L. Mallet, *Discrete Smooth Interpolation in geometric modeling*, Computer Aided Design Journal, Vol 24, No. 4, 1992.

[43] J. L. Mallet, *GOCAD Report* 1993.

[44] O. Mariez, *Génération de courbes et modélisation de surfaces*, Rapport DEA, Université Nancy I, 1993.

[45] O. Mariez, *Modélisation de solides : vers une synthèse des modèles à base de surfaces non-manifold et de l'analyse d'images 3D*, Thèse I.N.P.L en cours.

[46] M.E Mortenson, *Geometric Modeling*, John Wiley, 1985.

[47] G.M. Nielson, *Transfinite, Visually Continuous, Triangular Interpolant* Geometric Modeling: Algorithms and new trends, pp235-246, S.I.A.M, 1987.

[48] J. Pouzet, *Estimation d'une surface faillée pour un tracé automatique d'isovaleurs*, proceedings of the 26th International Geological Congress, Sciences de la Terre, série Informatique géologique, pp. 163-174, 1980.

[49] H. Rakotoarisoa, *Modélisation géométrique et optimisation de structures géologiques 3D*, Thèse Université Claude Bernard, Lyon1, 1992.

[50] A.M. Roberts, G. Yielding, B. Freeman, *The Geometry of Normal Faults*, Geological Society Special Publication N^o 56, 1991.

[51] P. Sablonniere, *Composite finite elements of class* C^k, Journal of Computationnal and Applied Mathematics 12&13, pp. 541-550, 1985.

[52] P. Samson, J.L. Mallet, *Curvature Analysis of Triangulated Surfaces : Application to Structural Geology*, Article soumis en 1994.

[53] J.Y. Talon, *Génération et amélioration de maillages pour éléments finis en deux et trois dimensions*, Thèse INPG, Grenoble, 1989.

[54] Turner, *Three-Dimensional Modeling with Geoscientific Information Systems*, NATO ASI Series, Mathematical Physical Sciences, Vol. 354, 1989.

[55] J.J. Walsh, J. Watterson, *Analysis of the relationship between displacement and dimensions of faults*, Journal of Structural Geology, Vol 10. N^0 3, 1988.

[56] J.J. Walsh, J. Watterson, *Displacement gradients on faults surfaces*, Journal of Structural Geology, Vol 11. N^0 3, 1989.

[57] J.J. Walsh, J. Watterson, *Modelling of bed contours and cross-sections adjacent to planar normal faults*, Journal of Structural Geology, Vol 11. N^0 3, 1989.

[58] N. White, G. Yielding, *Calculating normal fault geometries at depth: theory and examples*, The geometry of normal faults, Geological, Society Special Publication N^o 56, pp 251-260.

[59] L. Wietzerbin, *Modélisation et paramétrisation d'objets naturels de formes complexes en trois dimensions. Application à la simulation stochastique de la distribution d'hétérogénéités au sein des réservoirs pétroliers*, Thèse I.N.P.L, 1994.

Table des figures