

Thèse présentée pour obtenir le grade de
Docteur de l'Université Louis Pasteur
Strasbourg I

Discipline : informatique
par Marc Grunberg

Conception d'une méthode de
maillage 3D parallèle pour la
construction d'un modèle de Terre
réaliste par la tomographie sismique.

Soutenue publiquement le 29 septembre 2006

Membres du jury

Directeurs : Mme Catherine Mongenet, professeur
Université Louis Pasteur de Strasbourg
M. Michel Granet, physicien
Université Louis Pasteur de Strasbourg

Rapporteur interne : M. Jean-Jacques Lévêque, directeur de recherche CNRS
Université Louis Pasteur de Strasbourg

Rapporteurs externes : M. Denis Trystram, professeur
Institut National Polytechnique de Grenoble
M. Thierry Priol, directeur de recherche INRIA
Institut de Recherche en Informatique et Systèmes Aléatoires de Rennes

Examineur : M. Stéphane Genaud, maître de conférences
Université Robert Schuman de Strasbourg

Remerciements

En tout premier lieu, je tiens à remercier mes deux co-directeurs de thèse, Catherine Mongenet et Michel Granet, pour m'avoir dirigé et soutenu dans ce travail, et s'être montrés aussi disponibles malgré leurs nombreuses activités.

Toute ma reconnaissance va à Stéphane Genaud, pour m'avoir encadré durant cette thèse, et avec qui j'ai eu énormément de plaisir à travailler.

Je suis très reconnaissant à Jean-Jacques Lévêque d'avoir accepté la tâche ardue de rapporter mon travail ainsi que de présider mon jury de thèse, ainsi qu'à Thierry Priol et Denis Trystram d'avoir également accepté d'être rapporteurs de ma thèse. Leurs commentaires et leurs questions m'ont permis de clarifier certains points du manuscrit et m'ont donné de nouvelles pistes de réflexion.

Mes remerciements à toute l'équipe de l'ICPS pour l'excellente ambiance de travail et à mes collègues du ReNaSS, Christiane, Christophe, et les transfuges Alain et Thibaud.

Merci enfin à mes parents, ma grand-mère, toute ma famille, et notamment Isabelle et Rebecca, pour m'avoir encouragé inconditionnellement et aidé pendant toute cette période.

Résumé

Depuis 30 ans de nombreux travaux ont été menés en tomographie sismique, pour imager la structure interne de la Terre. Les évolutions des moyens de calcul, tant en puissance des processeurs qu'en capacité mémoire, ont permis d'obtenir des images de plus en plus précises. Mais on sait depuis longtemps que la résolution des modèles tomographiques est limitée par un certain nombre de facteurs, notamment par les répartitions hétérogènes des séismes et des stations sismologiques. Ceci est dû aux différents contextes sismotectoniques qui font que certaines régions du globe sont plus sismiques que d'autres, ainsi qu'au placement des capteurs plutôt situés sur les continents que dans les océans. De cette répartition résulte une distribution hétérogène des rais sismiques à l'intérieur de la Terre, conduisant à un échantillonnage spatial du globe extrêmement variable. La conséquence directe est que les régions faiblement échantillonnées ont une résolution plus faible que les régions bien échantillonnées. L'idée développée ces dernières années est de construire un *modèle irrégulier* qui puisse tirer parti au mieux des données disponibles en s'adaptant localement à la densité des informations sismiques, et ainsi avoir une résolution fine là où les données le permettent.

L'objectif du travail présenté dans cette thèse est de construire un maillage irrégulier afin de réaliser un modèle tomographique global. Ce maillage est constitué de cellules hexaédriques dont la taille est adaptée localement à la quantité et à la qualité de l'information apportée par les rais sismiques. La construction d'un tel maillage nécessite l'utilisation de très nombreuses données qui proviennent de la sismicité mondiale des années 1964 à 1995. Ceci représente environ 82000 séismes et 12 millions de rais. L'exploitation d'une telle masse de données dans le cadre d'une modélisation adaptative de la Terre nous a conduit à proposer de nouveaux algorithmes, ainsi que leur parallélisation, pour faire face, d'une part aux besoins en ressources de calcul, et d'autre part à l'importante consommation en ressources mémoire. Afin d'évaluer les performances de ces algorithmes, des expérimentations sur différentes plates-formes parallèles (machine parallèle, cluster et grille de calcul) ont été menées sur les applications de tracé de rais sismiques et sur la construction du maillage irrégulier.

Abstract

Since the last 30 years, many works have been conducted in seismic tomography to reveal the structure of Earth interior. Because of the computer architectures evolution, both in power of the processors and in memory size, it is possible to obtain increasingly precise images. However it is known for a long time that resolution of the tomographic models is limited by a certain number of factors, in particular by the heterogeneous distributions of the seisms and the seismological stations. This is due primary to the various tectonic contexts which make certain areas more seismic than others, and secondly to the placement of the sensors rather located on the continents than in the oceans. This implies an heterogeneous distribution of the seismic rays inside the Earth, leading to an extremely variable sampling of the Earth. The direct consequence is that the slightly sampled areas have a resolution weaker than the well sampled areas. The idea developed these last years is to build a *irregular model* which can, as well as possible, benefit from the available data, while adapting locally to the density of seismic information, and result in a fine resolution where the data allow it.

The objective of the work presented in this thesis is to build an irregular mesh in order to carry out a global tomographic model. This mesh consists of hexahedral cells whose size is adapted locally to the quantity and the quality of the information brought by the seismic rays. The construction of such a mesh requires a huge amount of data which comes from the seismicity of the world from the years 1964 to 1995. This represents approximately 82000 seisms and 12 millions of rays. Carrying out such huge data within the framework of an adaptive modeling of the Earth led us to propose new algorithms, and their parallelization, to face the requirements in CPU and memory resources. In order to evaluate their performances, we have tested the seismic ray tracing application and the irregular mesh construction on various parallel platforms (parallel machine, cluster and computational grid).

Table des matières

1	Introduction	1
2	Contexte du problème	7
2.1	La tomographie sismique à ondes de volume	7
2.1.1	Introduction	7
2.1.2	Tomographie à ondes de volume	11
2.1.3	Paramétrisation	13
2.1.4	Méthodes de tracé de rais	16
2.1.5	Solution tomographique	19
2.1.6	Régularisation	19
2.1.7	Résolution du problème numérique	21
2.2	Les architectures parallèles et les grilles	21
2.2.1	Classification des architectures parallèles	21
2.2.2	Les grilles	23
2.2.3	Programmation des applications parallèles	24
2.3	Les maillages	26
2.3.1	Définition d'un maillage en 3D	26
2.3.2	Génération de maillage structuré	27

2.3.3	Génération de maillage non structuré	28
2.3.4	Adaptation de maillages conformes	30
2.3.5	Parallélisation de maillage	32
3	Tracé de rai dans un maillage	35
3.1	Construction du maillage	36
3.1.1	Le maillage et les cellules	36
3.1.2	Description du maillage	38
3.1.3	Construction du maillage	39
3.2	Le tracé de rais	42
3.2.1	Analyse de la signature du rai	42
3.2.2	Tracé de rai en 3D	48
3.2.3	Intégration du rai dans le maillage	50
3.2.4	Calcul des scores	51
3.3	La parallélisation	52
3.3.1	Utilisation de la mémoire : maillage allégé	56
3.3.2	Distribution des rais dans le maillage	57
3.3.3	Description de l'algorithme parallèle	60
3.4	Résultats expérimentaux	61
3.4.1	Implémentation	65
3.4.2	Les plates-formes utilisées	66
3.4.3	Test de scalabilité	68
3.4.4	Série de tests avec 10 processeurs	73
3.4.5	Série de tests avec 16 processeurs	74
3.5	Conclusion	77

4	Le maillage adaptatif	79
4.1	Construction du maillage irrégulier	81
4.1.1	Évaluation de l'illumination	83
4.1.2	Comparaison des scores	84
4.1.3	Calcul des configurations	84
4.1.4	Sélection des configurations	88
4.2	Algorithme parallèle	96
4.2.1	Subdivision du maillage	96
4.2.2	Les zones de recouvrement	96
4.2.3	Construction de la liste $\mathcal{L}_{i,j}$	99
4.2.4	Sélection et nettoyage	99
4.3	Résultats expérimentaux	101
4.3.1	Implémentation	103
4.3.2	Jeu de données et paramètres de l'application	103
4.3.3	Les résultats	104
4.4	Conclusion	110
5	Résultats tomographiques	113
5.1	Construction du maillage initial	114
5.2	Constitution du jeu de données à tracer	114
5.2.1	Jeu de données initial	115
5.2.2	Sélection des rais	116
5.2.3	Ray bundles	116
5.3	Tracé de rais	117
5.4	Génération du maillage irrégulier	118

5.5	Résolution du problème inverse	119
5.5.1	Construction de la matrice	119
5.5.2	Calcul de la solution	122
5.6	Résultats	122
5.6.1	Conclusion	126
6	Conclusion et perspectives	129
A	Nomenclature des phases sismiques	139
B	Liste des publications	145

Table des figures

2.1	Propagation des ondes de volume P et S dans un milieu élastique. Le sens de propagation est indiqué par les flèches.	8
2.2	Sismogrammes d'un séisme et les différentes phases sismiques P et S associées. L'échelle horizontale est le temps écoulés en seconde depuis l'heure origine du séisme.	9
2.3	Représentation des interfaces majeures du globe et du modèle de vitesse de référence $ak135$ pour les ondes P et S	10
2.4	Réflexion et réfraction d'un rai sismique sur une interface séparant deux milieux de vitesses différentes.	11
2.5	Représentation d'un rai à travers un modèle 2D composé de blocs.	17
2.6	Exemple de maillages obtenus en résolvant le système de Laplace et celui de Poisson.	28
2.7	Construction d'un maillage en utilisant la méthode frontale.	29
2.8	Construction d'un maillage à partir d'un nuage de points en utilisant la méthode de Delaunay-Voronoi.	29
2.9	Construction du maillage d'un disque par la méthode des quadrees.	30
2.10	Exemple en 2D de déplacement d'un sommet (en blanc).	31
2.11	Exemple de raffinement et de déraffinement d'arêtes en 2D.	32
3.1	Les trois étapes de la décomposition d'une sphère lors de la construction du maillage.	37
3.2	Orientation et géométrie des faces des cellules du maillage.	37

3.3	Description XML du maillage des principales enveloppes de la Terre.	39
3.4	Description XML du maillage de la région euro-méditerranéenne	39
3.5	Maillage global et régional.	40
3.6	Les structures de données constituant le maillage.	41
3.7	Étape <i>topologique</i> dans la construction de la couche n du maillage. Les relations de voisinage, Nord/Sud et Est/Ouest, entre les cellules sont mises en place.	42
3.8	Exemple des différents chemins empruntés par les ondes sismiques, correspondant à un séisme, entre la source (S) et la station sismologique (R). Les trajets en bleu (resp. rouge) correspondent à une propagation en compression (resp. cisaillement).	44
3.9	Exemples de signature de rai. La distance épacentrale d est définie par l'angle \widehat{SOR} , où S est la source du séisme, O le centre de la Terre, et R le récepteur.	46
3.10	Analyse de la phase PcPSKS.	47
3.11	Chaque point P du rai est défini par sa profondeur r , un angle d'incidence i , ainsi qu'un angle $\theta = (\widehat{P_0OP})$. Le premier point du rai P_0 (la source) est défini par $(r_0, i_0, \theta_0 = 0)$. O est le centre de la Terre et P_{bas} le point bas du rai.	48
3.12	Distribution du nombre d'itérations réalisées pour la construction des rais P, pP et PP provenant du jeu de données de [ENGDAHL <i>et al.</i> 98].	50
3.13	Structure de données associée à l'information apportée par un rai dans une cellule.	51
3.14	Structure de données associée à l'information apportée par l'ensemble des rais dans une cellule.	51
3.15	(Partie 1/2) – Scores élémentaires permettant de caractériser l'illumination des cellules du maillage par les rais sismiques entre 760 et 2740 km de profondeur.	53
3.15	(Partie 2/2) – Scores élémentaires permettant de caractériser l'illumination des cellules du maillage par les rais sismiques entre 760 et 2740 km de profondeur.	54
3.16	Score composite résultant du produit des quatre scores élémentaires entre 760 et 2740 km de profondeur.	55

3.17	Occupation mémoire minimale et maximale de l'application avant et après la phase d'échange, dans le cas d'une distribution des rais en mode <i>maître/esclave</i> .	58
3.18	(Partie 1/2) – Schéma de la parallélisation de l'application pour une distribution de données en mode <i>maître/esclave</i> .	62
3.18	(Partie 2/2) – Schéma de la parallélisation de l'application pour une distribution de données en mode <i>maître/esclave</i> .	63
3.19	Distribution par signatures du nombre de rais calculés à partir de 348 401 rais extraits des bases de données de l'ISC.	64
3.20	Description XML du maillage global utilisé lors des expérimentations (712 800 cellules réparties sur 11 couches).	64
3.21	Organisation des briques logiciels.	66
3.22	Représentation graphique de l'efficacité du tracé de rais sismiques e en fonction du nombre de processeurs.	69
3.23	Performances du tracé de rais sismiques dans un maillage global de la Terre, obtenues sur la machine parallèle SGI Origin 3800 du CINES.	71
3.24	Performances du lancé de rais sismiques dans un maillage global de la Terre, obtenues sur le cluster HPC.	72
3.25	Performances sur un cluster de PC et une SGI Origin 3800 (10 processeurs).	73
3.26	Performances de la configuration type grille métropolitaine de Strasbourg (10 processeurs).	74
3.27	Partie de la dorsale nationale Renater empruntée et évolution des débits entre Renater 2bis et Renater 3.	75
3.28	Performances avec 16 processeurs sur la grille nationale avec Renater 2 puis Renater 3.	76
3.29	Performances sur une Origin 3800 (16 processeurs).	77
4.1	La sismicité mondiale de 1928 à 2004. En vert les contours des plaques tectoniques.	80
4.2	Répartition des stations sismologiques en activité à la fin de l'année 2004.	80
4.3	Organisation des briques logicielles pour la construction du maillage irrégulier.	81

4.4	Représentation de la meta-cellule $M_{x,y}^{4,2}$ de taille $p \times q = 4 \times 2$ ayant pour cellule de tête $C_{x,y}$ dans le maillage Ω	83
4.5	Représentation de la fonction de calcul du meta-score $F(s)$ en fonction du score s d'une meta-cellule, et du score optimal s_{optim} . Le maximum de $F(s)$ (<i>ie.</i> le meilleur meta-score) est obtenu pour $s = s_{optim}$	85
4.6	Exemple de configurations des meta-cellules constituant le début de la liste \mathcal{L} , juste après sa création.	87
4.7	La création de la meta-cellule $M_{x,y}^{p,q}$ ayant pour cellule de tête $C_{x,y}$ dans le maillage Ω nécessite de nettoyer la liste \mathcal{L} des meta-cellules devenues incompatibles avec cette création. Une première phase de nettoyage <i>en avant</i> (a), travaille sur la zone de recherche définie par $M_{x,y}^{p,q}$, tandis qu'une deuxième phase de nettoyage <i>en arrière</i> (b), travaille sur la zone de recherche <i>Area</i>	89
4.8	(Partie 1/2) – Exemple de la construction du maillage irrégulier par <i>sélection</i> et <i>nettoyage</i> sur la liste \mathcal{L} (le nettoyage arrière induit des modifications des éléments en aval de la liste non représentés sur le schéma).	90
4.8	(Partie 2/2) – Exemple de la construction du maillage irrégulier par <i>sélection</i> et <i>nettoyage</i> sur la liste \mathcal{L} (le nettoyage arrière induit des modifications sur des éléments en aval de la liste non représentés sur le schéma).	91
4.9	Nombre cumulé de configurations de meta-cellules supprimées en fonction de $p_{max} \times q_{max}$ et du nombre d'appels à la procédure SelectClean	94
4.10	Subdivision d'une couche Ω du maillage régulier, en 16 sous-domaines.	97
4.11	Identification et construction de la zone de recouvrement ($Z_{i,j}^1 \cup Z_{i,j}^2$) du maillage $\Omega_{i,j}$	98
4.12	Sous-domaine $D_{i,j}$ ainsi que les zones de recouvrement liées aux domaines voisins.	99
4.13	Performances de l'application obtenues sur le cluster Itanium2 et le cluster Xeon pour $p_{max} \times q_{max} = 10 \times 10$ en fonction du nombre de processeurs.	106
4.14	Performances de l'application obtenues sur le cluster Itanium2 et le cluster Xeon pour $p_{max} \times q_{max} = 20 \times 20$ en fonction du nombre de processeurs.	107
4.15	Maillage irrégulier de la couche 7 (de 760 à 2740 km) constitué de 17 827 meta-cellules avec $p_{max} \times q_{max} = 10 \times 10$. La fonction de score utilisée (f) est la densité de rais, et le score optimal est 2.5 (<i>ie.</i> le nombre de rais souhaité dans les meta-cellules est $10^{2.5}$).	109

4.16	Distribution des scores associés aux meta-cellules du maillage final.	110
5.1	Organisation des briques logicielles pour la tomographie complète.	113
5.2	Description du maillage utilisé pour la tomographie qui est constitué de 18 couches de la surface jusqu'à l'interface manteau-noyau à 2889 km de profondeur.	115
5.3	Exemples types de trajets en 2D des rais tracés.	117
5.4	Description XML du maillage irrégulier.	119
5.5	(Partie 1/3) – Visualisation des 18 couches du maillage irrégulier.	120
5.5	(Partie 2/3) – Visualisation des 18 couches du maillage irrégulier.	121
5.5	(Partie 3/3) – Visualisation des 18 couches du maillage irrégulier.	122
5.6	Résultats tomographiques obtenus dans cette thèse à 500 km et 1300 km de profondeur.	124
5.7	Résultats tomographiques obtenus par [VAN DER HILST <i>et al.</i> 97] à 500 km et 1300 km de profondeur. Pour un meilleur résultat visuel, les blocs initialement de taille $2^\circ \times 2^\circ$, ont été ré-échantillonnés à posteriori par les auteurs par des blocs de $1^\circ \times 1^\circ$	125
5.8	Résultats tomographiques obtenus par [BIJWAARD <i>et al.</i> 98] à 1325 km de profondeur.	126
5.9	Résultats tomographiques obtenus par [SAMBRIDGE & FALETIC 03] à 500 km et 1300 km de profondeur.	127

Liste des tableaux

3.1	Caractéristiques des ressources de calcul utilisées dans les expériences. . . .	66
3.2	Performance du tracé de rais sismiques sur la machine parallèle du CINES.	69
4.1	Performances obtenues pour $p_{max} \times q_{max} = 10 \times 10$	104
4.2	Performances obtenues pour $p_{max} \times q_{max} = 20 \times 20$	105
4.3	Pourcentage représentant la part du nombre de cellules contenues dans zone de recouvrement $(Z_{i,j}^1 \cup Z_{i,j}^2)$ par rapport au nombre total de cellules du maillage $D_{i,j}$, en fonction du nombre de processeurs et de la taille maximale $p_{max} \times q_{max}$ d'une meta-cellule.	108
5.1	Répartition des rais tracés en fonction du type de phases	118
5.2	Tableau comparatif des paramètres de tomographie utilisés.	123

Chapitre 1

Introduction

En enregistrant à la surface de la Terre les mouvements du sol provoqués par les séismes, il est possible de recueillir des informations sur la structure interne de notre planète. L'enjeu essentiel de la tomographie¹ sismique consiste, à partir de ces enregistrements, à modéliser de façon réaliste les caractéristiques physiques des hétérogénéités de l'intérieur de la Terre.

L'énergie libérée par un séisme se propage dans toutes les directions sous forme d'ondes sismiques. On distingue deux types d'ondes : les ondes de volume qui traversent la Terre (en compression ou en cisaillement) et les ondes de surface qui se propagent parallèlement à sa surface avec une pénétration verticale qui dépend de la fréquence de l'onde. Leur vitesse de propagation et leur amplitude sont modifiées par les structures géologiques qu'elles traversent. Les stations sismologiques, réparties à la surface de la Terre, enregistrent les signaux correspondant à ces ondes, appelés sismogrammes. Ces signaux dépendent directement des milieux traversés.

Les paramètres qui permettent de déduire les structures géologiques sont les vitesses de propagation des ondes sismiques. Tant qu'il n'y a pas de changement physico-chimique dans le milieu traversé par ces ondes, leur vitesse de propagation reste constante. C'est par une analyse des contrastes de vitesse que l'on arrive à déduire les différentes structures géologiques de la Terre, comme par exemple les limites des plaques tectoniques ou les points chauds dans le manteau.

Il existe principalement deux méthodes de tomographie. De façon schématique, la première méthode dite à *temps de trajet* (travel-time tomography) utilise les temps de trajet des différentes ondes de volume, obtenus à partir des sismogrammes enregistrés. Elle tente de trouver un modèle de Terre dans lequel, en simulant la propagation de ces fronts d'onde

¹Du préfixe grec *tomo* morceau coupé et de *graphe* image.

par des *rais sismiques*², on aurait les mêmes temps de trajet que ceux observés. La deuxième méthode (waveform tomography) consiste à trouver un modèle de Terre dans lequel un sismogramme synthétique serait semblable au véritable sismogramme enregistré. Dans cette dernière méthode le nombre d'inconnues est beaucoup plus important que pour la méthode à temps de trajet, et les équations reliant les inconnues du modèle aux données conduisent à la construction d'un système non linéaire. C'est pourquoi la tomographie à temps de trajet est la plus couramment utilisée, et le travail présenté dans cette thèse s'appuie sur cette méthode.

Depuis 30 ans de nombreux travaux ont été menés en tomographie sismique, pour imaginer la structure interne de la Terre. Les principales méthodes employées sont caractérisées par :

- la zone géographique d'étude : locale ou globale avec une profondeur variable,
- l'utilisation d'ondes de surface, permettant d'imager plutôt la lithosphère³ ou de volume pour imaginer le manteau et le noyau,
- une modélisation fondée soit sur une décomposition en blocs réguliers, soit sur un développement en harmoniques sphériques. Le choix de l'une ou l'autre de ces méthodes est adapté à l'objectif que l'on souhaite atteindre. Par exemple [VAN DER HILST *et al.* 97] ont utilisé une modélisation fine par blocs pour imaginer la convection dans le manteau, alors que [DZIEWONSKY 84] a plutôt utilisé des harmoniques sphériques pour étudier les variations latérales du manteau inférieur, notamment au niveau de l'interface noyau-manteau.

Les diverses améliorations apportées à ces méthodes ainsi que les évolutions des moyens de calcul, tant en puissance des processeurs qu'en capacité mémoire, ont permis d'obtenir des images de plus en plus précises. Mais on sait depuis longtemps que la résolution⁴ des modèles tomographiques est limitée par un certain nombre de facteurs, notamment par les répartitions inhomogènes des séismes et des stations sismologiques. Ceci est dû en particulier aux différents contextes sismotectoniques qui font que certaines régions du globe sont plus sismiques que d'autres et aussi que les capteurs sont plutôt placés sur les continents que dans les océans. De cette répartition résulte une distribution hétérogène des rais sismiques à l'intérieur de la Terre, conduisant à un échantillonnage spatial du globe extrêmement variable. La conséquence directe est que les régions faiblement échantillonnées ont une résolution plus faible que les régions bien échantillonnées. L'idée développée ces dernières années est de construire un *modèle irrégulier* qui puisse tirer parti au mieux des données disponibles en s'adaptant localement à la densité des informations sismiques, et ainsi avoir une résolution fine là où les données le permettent. Pour réaliser cet objectif [BIJWAARD *et al.* 98] ont construit à partir d'une cellule de base hexaédrique, des cel-

²Intuitivement le rai sismique correspond au trajet emprunté par l'onde, de la source du séisme au capteur sismologique.

³La lithosphère a une épaisseur comprise entre 100 et 300 kilomètres, elle est constituée de la croûte et d'une partie du manteau supérieur, et correspond grossièrement à la notion de plaques tectoniques.

⁴La résolution indique la taille de la plus petite structure géologique observable.

lules irrégulières de forme quelconque, alors que [SAMBRIDGE & FALETIC 03] ont procédé plutôt par raffinement d'un maillage tétraédrique.

La construction d'un tel modèle nécessite l'utilisation de données nombreuses et de qualité, caractérisées par une localisation précise des séismes et par une bonne identification des différents temps d'arrivée des ondes sismiques sur les sismogrammes. Un tel jeu de données a été constitué par [ENGDAHL *et al.* 98] à partir de la base de données de la sismicité mondiale de l'ISC (International Seismic Center) pour les années de 1964 à 1995. Il contient une très grande quantité d'informations, 82000 séismes et environ 12 millions de temps d'arrivée d'ondes sismiques. L'exploitation d'une telle masse de données dans le cadre d'une modélisation adaptative de la Terre nécessite l'implémentation de nouveaux algorithmes, ainsi que leur parallélisation, pour faire face, d'une part aux besoins en ressources de calcul, et d'autre part à l'importante consommation en ressources mémoire.

Objectifs de la thèse

Le travail présenté dans cette thèse s'inscrit dans le cadre des méthodes de tomographie sismique à temps de trajet. L'objectif est de réaliser une tomographie globale dans laquelle on manipule des ondes de volume. La méthode consiste à construire un maillage hexaédrique irrégulier dont la taille des cellules est adaptée localement à la quantité et à la qualité de l'information apportée par les rais sismiques. Ces informations caractérisent ce que les sismologues appellent couramment *l'illumination* d'une cellule.

La construction du maillage irrégulier est automatique et ne dépend que de la géométrie des rais sismiques propagés à l'intérieur de la Terre, ainsi que de certains paramètres fixés par l'expert sismologue. Celui-ci définit notamment les tailles minimale et maximale raisonnables pour les cellules du modèle (qui ait un sens du point de vue géophysique), ainsi qu'un ensemble de règles qui permet de caractériser le degré d'illumination d'une région du globe. Ces règles combinent un ensemble d'informations calculées dans chaque cellule comme par exemple :

- le nombre de rais,
- la longueur des rais,
- le nombre d'impacts de rais par face de la cellule,
- les coordonnées du point d'entrée et de sortie de chaque rai qui traverse la cellule,
- l'angle de pénétration de chaque rai dans la cellule.

Méthode

La méthode développée dans cette thèse se décompose comme suit :

1. La construction d'un maillage initial régulier hexaédrique

Le maillage initial de la Terre est obtenu en décomposant une sphère en un certain nombre de couches concentriques, dont les épaisseurs sont adaptées au contexte géo-

physique, de la surface jusqu'au centre. Chaque couche est ensuite décomposée, à partir du centre de la Terre, en secteurs angulaires, à la fois en latitude et en longitude. Les volumes élémentaires ainsi créés peuvent être approximés par des hexaèdres et constituent les cellules du maillage, souvent appelé maillage sphérique. On affecte ensuite à chaque cellule une vitesse de propagation des ondes selon un modèle de vitesse de référence, le modèle ak135 [KENNETT *et al.* 95], qui ne dépend que de la profondeur.

2. Le tracé de rais sismiques dans le maillage initial

Il est possible de modéliser dans un maillage la propagation d'un rai, du foyer du séisme à la station sismologique, à partir d'un modèle de vitesse et d'informations extraites des bases de données sismologiques. Ces informations sont :

- les coordonnées géographiques du foyer et de la station,
- la signature du rai (ou phase).
- l'angle d'incidence initial du rai au foyer du séisme, pour un modèle de vitesse donné.

(a) Analyse de la signature du rai

Cette signature est formée de symboles qui précisent chacun des changements qui influent sur la propagation du rai lorsqu'il rencontre une interface géologique majeure (à la surface de la terre, ou à l'interface manteau/noyau par exemple). Ces changements peuvent concerner le mode de propagation de l'onde qui peut passer d'un mode en compression à un mode en cisaillement et réciproquement. Il peut aussi s'agir d'une réflexion ou d'une réfraction au niveau d'une interface. L'analyse des ces signatures a été automatisée en implémentant un analyseur avec l'outil YACC.

(b) Tracé de rai dans le maillage

La construction du rai en trois dimensions est un processus itératif qui consiste à propager le rai pas à pas dans le maillage. L'algorithme basé sur la loi de Snell-Descartes est piloté par l'analyse de la signature du rai, pour décider des éventuelles réflexions ou réfractions, ou bien d'un changement de mode de propagation lorsque le rai atteint une interface géologique. Au fur et à mesure que le rai est tracé, son temps de propagation entre foyer et station est calculé, en utilisant le modèle de vitesse de référence. A l'issue du tracé d'un rai, il arrive que le point final ne coïncide pas exactement avec les coordonnées de la station. Cela provient soit d'une imprécision sur l'angle d'incidence initial associé, soit de l'utilisation d'un modèle de vitesse de référence différent de celui qui est spécifié dans les bases de données. Lorsque la distance entre ces deux points est supérieure à un certain seuil de précision, on procède à un calcul itératif du rai en faisant varier de façon dichotomique l'angle d'incidence jusqu'à obtenir une précision suffisante.

3. Construction du maillage adaptatif

Le maillage initial étant constitué de couches concentriques, la construction du mail-

lage adaptatif peut être menée indépendamment sur chaque couche. Cette construction est réalisée par fusion d'un ensemble rectangulaire de cellules pour former une *meta-cellule*⁵. Le processus de fusion vise à construire un maillage irrégulier dans lequel un maximum de meta-cellules ont une bonne illumination. Si l'illumination d'une cellule n'est pas suffisante par rapport à un certain seuil fixé, l'algorithme de construction du maillage irrégulier tendra à l'agréger avec ses cellules voisines si l'illumination de la meta-cellule résultante est meilleure que celle de la plupart des cellules prises individuellement.

4. Inversion tomographique

L'objectif final est de raffiner le modèle de Terre obtenu, en modifiant la vitesse de chaque cellule du maillage irrégulier, afin que pour chaque rai, le temps de propagation calculé soit le plus proche possible du temps de propagation observé, lequel est mémorisé dans les bases de données sismologiques. Cette étape, appelée inversion tomographique, consiste à résoudre un système d'équations linéaires dont les inconnues sont les lenteurs (*ie.* l'inverse de la vitesse) et les données sont les longueurs des segments de rais dans les meta-cellules. Dans cette thèse, la méthode utilisée pour la résolution de ce système d'équations est la méthode LSQR⁶ [PAIGE & SAUNDERS 82].

Approche parallèle

L'ambition de cette thèse est de réaliser une tomographie globale en utilisant les bases de données de sismicité existantes contenant une très grande quantité de rais sismiques à modéliser (plusieurs dizaines de millions). Les algorithmes développés ont donc été parallélisés pour éviter des temps de calcul prohibitifs, sur une architecture séquentielle classique, et pour s'affranchir des contraintes de mémoire. A titre d'exemple pour un jeu de données de 1 million de rais, et avec un maillage initial composé de 300000 cellules, le calcul nécessite plus de 10 heures de temps CPU sur un bi-processeurs Intel Xeon 3.2 Ghz, pour le seul tracé de rais en mode itératif dans le maillage régulier, et la taille des données produites est de 6 Go environ.

La parallélisation de l'algorithme du tracé de rais dans le maillage initial a été réalisée en distribuant les données sismiques sur différents processeurs. La longueur des rais est très variable, et le temps de calcul d'un rai dépend, d'une part de sa longueur, d'autre part du temps de convergence du processus itératif du tracé d'un rai. Pour garantir un bon équilibrage de charge, dépendant du type d'architecture exploitée, deux stratégies de distribution de données ont été étudiées et implémentées. La première effectue une distribution calculée statiquement, tandis que la deuxième réalise une distribution dynamique de type maître/esclave.

La construction du maillage adaptatif consiste à paver le maillage régulier initial en zones géographiques, et à affecter chaque zone à un processeur. Pour calculer les meta-

⁵La taille maximum d'une meta-cellule est un paramètre de l'application fixé par le sismologue.

⁶LSQR : Least Square Reconstruction.

cellules sur un processeur, une zone de recouvrement des données avec les processeurs adjacents est nécessaire. Par ailleurs lorsqu'un processeur sélectionne une meta-cellule, une communication peut-être nécessaire avec l'un ou plusieurs de ses voisins, afin de garantir la cohérence du maillage adaptatif final.

D'un point de vue logiciel, l'implémentation des programmes s'est faite de façon modulaire et portable, en langage C, et la bibliothèque de communication utilisée pour la parallélisation est MPI [MPI 95]. L'ensemble des programmes tourne sous différents systèmes (Linux, IRIX, SunOS), et la visualisation en 3D utilise VTK [SCHROEDER *et al.* 97].

Expérimentations

Une analyse des comportements de ces algorithmes a été menée sur diverses architectures matérielles telles que :

- des machines parallèles (SGI Origin 2000 et 3800),
- des clusters (Pentium4, IA-64 Itanium2),
- des grilles de calcul (métropolitaine et nationale),

et une série de tests a permis d'étudier l'équilibrage de charge, la scalabilité, et l'impact du réseau lorsque l'application est lancée sur une grille de calcul.

Par ailleurs, en utilisant le même jeu de données que [SAMBRIDGE & FALETIC 03], [BIJWAARD *et al.* 98] et [VAN DER HILST *et al.* 97], une tomographie globale a été réalisée, et la comparaison des résultats montre une grande similitude entre les modèles obtenus.

Chapitre 2

Contexte du problème

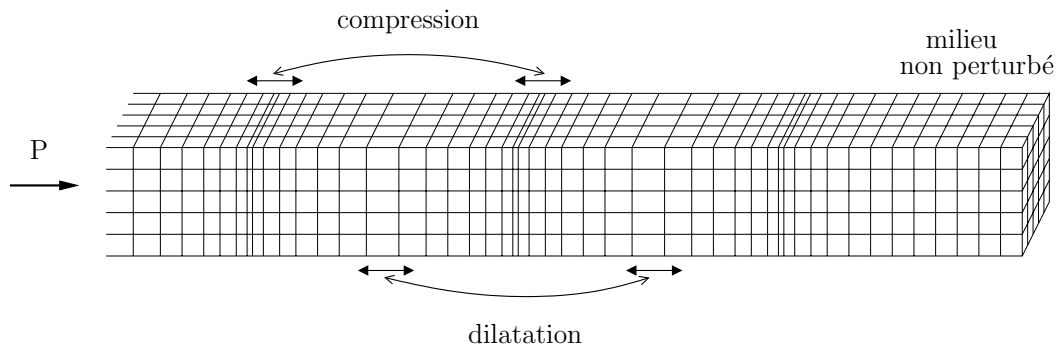
Dans ce chapitre nous présentons les éléments fondamentaux qui permettent d'aborder les problèmes de tomographie sismique, en nous intéressant en particulier aux applications de tomographie globale qui nécessitent une grande quantité de données ainsi que des ressources considérables en calcul et en mémoire. C'est pourquoi, après une présentation du problème de la tomographie sismique, nous exposons brièvement les différentes architectures parallèles disponibles aujourd'hui et susceptibles de pouvoir exécuter ce type d'application. Nous passons également en revue les différentes techniques de construction de maillage ainsi que les méthodes de maillage adaptatif qui permettent de résoudre numériquement le problème tomographique.

2.1 La tomographie sismique à ondes de volume

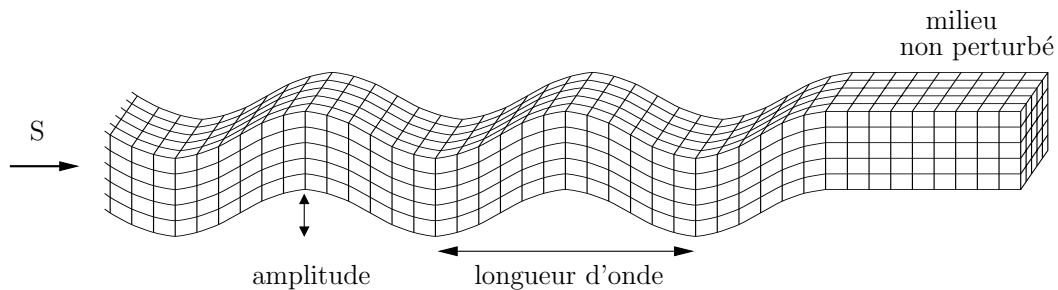
2.1.1 Introduction

Lorsqu'un séisme se produit, des ondes sismiques se propagent depuis le foyer du séisme dans toutes les directions en déformant de façon élastique les milieux traversés. On distingue deux types d'ondes (cf. figure 2.1) en fonction du type de déformation engendrée : les ondes de compression, dites ondes P , et les ondes de cisaillement, dites ondes S . Les vitesses de propagation de ces deux types d'ondes sont différentes et dépendent principalement des paramètres physico-chimiques du milieu traversé (en composition, pression et température). Tant que celui-ci ne présente aucun changement, ces vitesses restent constantes. Pour un milieu donné dans lequel ces ondes se propagent, la vitesse des ondes P est toujours plus élevée que celle des ondes S . De plus, contrairement aux ondes S qui ne peuvent se propager que dans les milieux solides, les ondes P se propagent également dans les milieux liquides

comme le noyau externe. Enfin, une onde P peut donner naissance lors d'une réflexion ou d'une transmission, à une onde S , et réciproquement. On parle alors d'onde convertie.



(a) Onde P : la déformation du milieu, en compression et dilatation, se fait dans le sens de la propagation de l'onde (onde longitudinale).



(b) Onde S : la déformation du milieu, en cisaillement, se fait perpendiculairement au sens de propagation de l'onde (onde transversale).

FIG. 2.1 – Propagation des ondes de volume P et S dans un milieu élastique. Le sens de propagation est indiqué par les flèches.

En disposant d'un ensemble de capteurs sismologiques à la surface de la Terre, les différents organismes de surveillance sismique enregistrent, sous forme de sismogrammes, les mouvements du sol qui résultent de l'arrivée des différentes ondes sismiques, appelées phases. Celles-ci correspondent non seulement aux différentes ondes de volume, mais aussi aux ondes qui se propagent le long de la surface de la Terre (les ondes de surface). La figure 2.2 représente un sismogramme sur lequel sont identifiées les différentes ondes sismiques ainsi que leur temps d'arrivée respectif.

Étant donné un modèle de vitesse de la Terre, l'interprétation de ces sismogrammes, où l'on identifie les différentes phases, notées i , ainsi que leur temps d'arrivée, noté T_i , permet

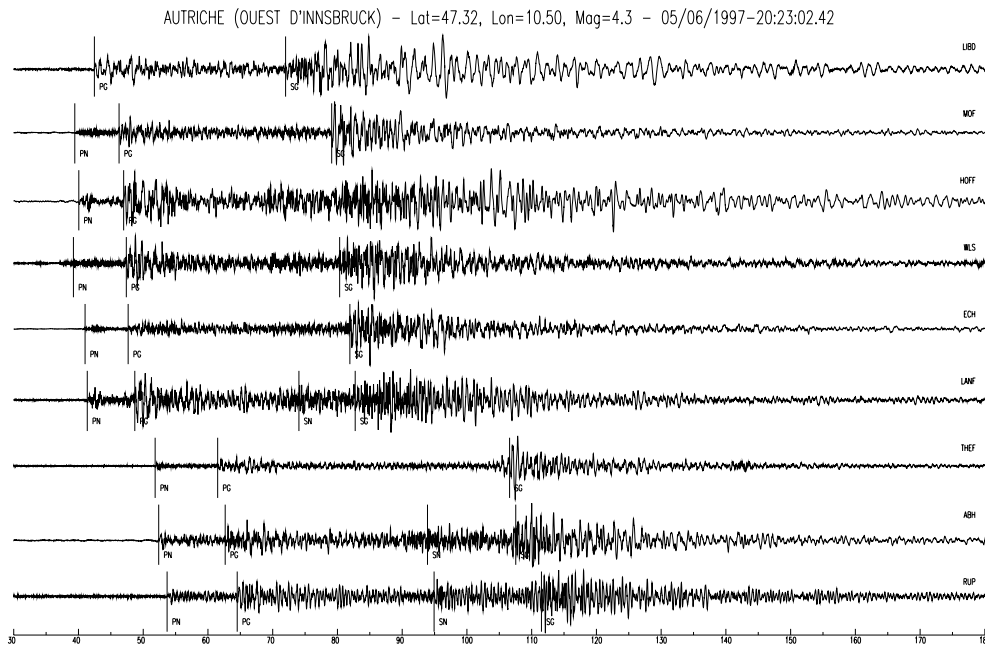


FIG. 2.2 – Sismogrammes d'un séisme et les différentes phases sismiques P et S associées. L'échelle horizontale est le temps écoulés en seconde depuis l'heure origine du séisme.

de localiser la source du séisme dans l'espace (latitude, longitude, profondeur) et dans le temps (on note le temps origine, ou heure du séisme, T_o).

On appelle *le temps de trajet* d'une phase i , son temps de propagation associé $t_i = T_i - T_o$, depuis la source du séisme vers le capteur sismologique. Ce temps est plus ou moins long en fonction du trajet emprunté par l'onde et dépend des différentes structures géologiques traversées.

La tomographie sismique par ondes de volume est une méthode géophysique qui permet d'affiner un modèle de vitesse de propagation des ondes sismiques de référence en un nouveau modèle. Dans la pratique le modèle de référence utilisé est généralement un modèle radial, qui ne dépend que de la profondeur (notée r), alors que le modèle affiné est à 3 dimensions. Dans la suite $v_o^P(r)$ et $v_o^S(r)$ désignent respectivement les vitesses des ondes P et S dans le modèle de référence et $v^P(x)$ et $v^S(x)$ les vitesses dans le modèle affiné, où x correspond aux coordonnées d'un point de l'espace 3D de la Terre (latitude, longitude, profondeur).

Cette méthode de tomographie utilise, pour les différentes ondes sismiques identifiées sur les sismogrammes, à la fois les temps de trajet observés (t_i^{obs}) et les temps de propagation théoriques¹ (t_i^{calc}) correspondants. La différence de temps mesurée ($t_i^{obs} - t_i^{calc}$), appelée

¹Le temps de propagation théorique d'une onde est calculé dans le modèle de vitesse de référence v_o .

résidu, est dûe à la présence de structures géologiques sur le chemin parcouru par l'onde dont les propriétés physico-chimiques diffèrent de celle du modèle de référence.

A partir de ces résidus, et du modèle de vitesse de référence (en P ou en S), on obtient le modèle affiné (respectivement en P ou en S) en réalisant une inversion matricielle, comme cela est présenté dans la suite. Dans le modèle affiné, l'analyse locale des contrastes de vitesse permet d'identifier la géométrie des structures géologiques présentes dans la Terre. De plus, en considérant les deux modèles de vitesses affinés P et S , le rapport des vitesses $v^P(x)/v^S(x)$ caractérise le type de roche.

La figure 2.3 représente un modèle de vitesse de référence couramment utilisé en géophysique, le modèle *ak135* [KENNETT *et al.* 95] qui ne dépend que de la profondeur. Elle montre que chaque interface majeure du globe induit un changement brutal de la vitesse de propagation des ondes. On constate notamment qu'à l'interface manteau-noyau la vitesse des ondes P chute quasiment d'un facteur deux, et que celles des ondes S est nulle. En effet les ondes S ne se propagent pas dans le noyau externe puisque celui-ci est liquide. Les ondes S peuvent néanmoins se retrouver dans le noyau interne lorsqu'elles sont le résultat d'une conversion de phase P en S à l'interface noyau externe / noyau interne.

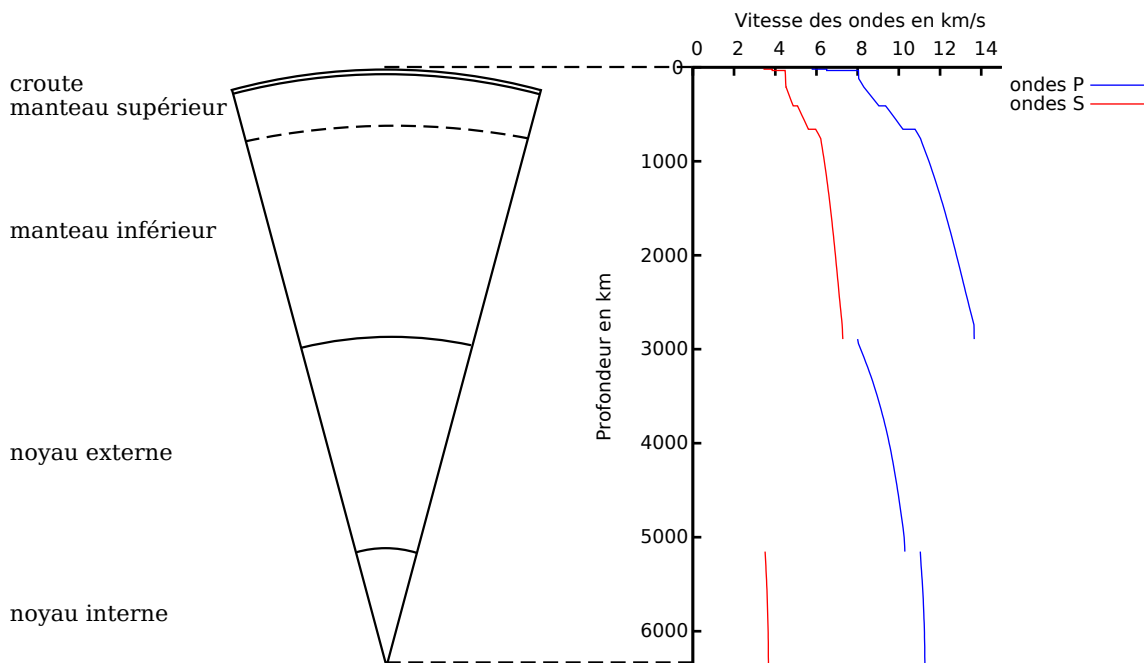


FIG. 2.3 – Représentation des interfaces majeures du globe et du modèle de vitesse de référence *ak135* pour les ondes P et S .

2.1.2 Tomographie à ondes de volume

La théorie des rais est une simplification de l'équation de propagation des ondes dans un milieu élastique. Celle-ci devient, en supposant que la solution est une onde plane et en faisant une approximation haute fréquence², l'équation de l'*eikonal*³ [CERVENY *et al.* 77] qui s'écrit :

$$\left(\frac{\partial t}{\partial x}\right)^2 + \left(\frac{\partial t}{\partial y}\right)^2 + \left(\frac{\partial t}{\partial z}\right)^2 = \frac{1}{v^2(x, y, z)} \quad (2.1)$$

où t est le temps de propagation du front d'onde et v sa vitesse de propagation dans un modèle en 3 dimensions.

Ceci permet de modéliser la propagation d'une onde de volume à l'intérieur de la Terre (*ie.* son trajet et son temps de propagation), de la source du séisme jusqu'à un capteur, à l'aide d'un rai qui est construit comme étant en tout point perpendiculaire au front d'onde. Ce rai suit la loi de réflexion et de réfraction de Snell-Descartes dès lors qu'il atteint une interface géologique.

Cette loi (illustrée par la figure 2.4) donne, pour un rai P ou S arrivant sur une interface séparant deux milieux de vitesses différentes la relation entre l'angle d'incidence i (angle entre le rai et la normale à l'interface) et :

1. l'angle de réflexion θ_1 entre le rai (P ou S) réfléchi et la normale,
2. l'angle de réfraction θ_2 entre le rai (P ou S) réfracté et la normale.

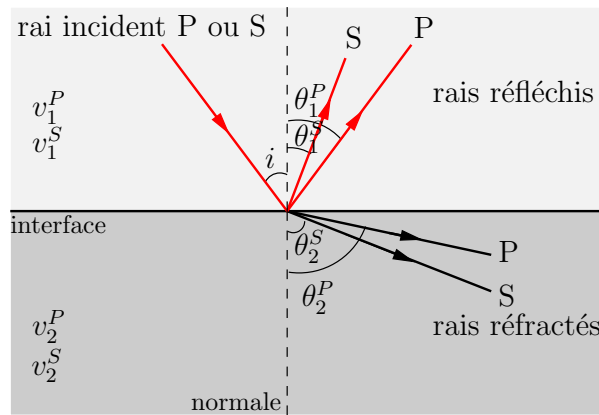


FIG. 2.4 – Réflexion et réfraction d'un rai sismique sur une interface séparant deux milieux de vitesses différentes.

²Dans l'approximation haute fréquence, la longueur d'onde de l'onde de volume est beaucoup plus petite que les structures rencontrées par celle-ci lors de sa propagation.

³Le terme *eikonal* vient du grec et signifie image.

La loi de réflexion et de réfraction s'écrit :

$$\frac{\sin i}{v_1^{P|S}} = \frac{\sin \theta_{milieu}^{phase}}{v_{milieu}^{phase}} \quad (2.2)$$

où *phase* prend la valeur *P* ou *S* et *milieu* la valeur 1 ou 2 selon que le rai est réfléchi ou réfracté.

On considère ici un ensemble de n phases sismiques déterminées par l'analyse de sismogrammes provenant de l'enregistrement de plusieurs séismes par plusieurs stations sismologiques.

Soit $t_1^{obs}, \dots, t_n^{obs}$ les temps de propagation *observés* de ces n phases, depuis la source d'un séisme vers un récepteur. La vitesse de propagation des ondes dans la Terre est donnée par la fonction $v(x)$, où $x \in \mathbb{R}^3$ est la position à l'intérieur de la Terre⁴. Le temps de propagation t_i^{obs} de l'onde i est l'intégrale de l'inverse de la vitesse $v(x)$ le long du chemin Γ_i parcouru par le rai, et il s'écrit :

$$t_i^{obs} = \int_{\Gamma_i} \frac{1}{v(x)} dl \quad (2.3)$$

où dl est un déplacement élémentaire sur le chemin Γ_i du rai i .

Soit $t_1^{calc}, \dots, t_n^{calc}$ les temps de propagation *calculés* de ces mêmes n phases dans le modèle de vitesse de référence $v_o(r)$, qui ne dépend que de la profondeur r . On note Γ_{oi} le chemin parcouru par le rai i dans ce modèle, son temps de propagation t_i^{calc} s'écrit :

$$t_i^{calc} = \int_{\Gamma_{oi}} \frac{1}{v_o(r)} dl \quad (2.4)$$

où dl est un déplacement élémentaire sur Γ_{oi} .

Les équations (2.3) et (2.4) sont non linéaires puisqu'elles sont respectivement en $1/v(x)$ et en $1/v_o(r)$. C'est pourquoi on définit la *lenteur* $s(x)$ comme étant l'inverse de $v(x)$, et $s_o(r)$ l'inverse de $v_o(r)$. Les deux relations précédentes deviennent alors linéaires en fonction de $s(x)$ et $s_o(r)$:

$$t_i^{obs} = \int_{\Gamma_i} s(x) dl \quad (2.5)$$

$$t_i^{calc} = \int_{\Gamma_{oi}} s_o(r) dl \quad (2.6)$$

Soit δs la perturbation de lenteur entre le modèle de référence et celui à déterminer. On suppose que cette perturbation est petite par rapport à la lenteur s (*ie.* $\delta s \ll s$), on a alors

⁴L'espace de tous les modèles de vitesse 3D $v : E = \{x \in \mathbb{R}^3 \mid x \in Terre\} \rightarrow \mathbb{R}$ est un sous espace de Hilbert (la généralisation en dimension quelconque d'un espace euclidien) noté $L_2(E)$.

$s \approx s_o + \delta s$. Ceci implique que la perturbation le long du trajet Γ_i (notée $\delta\Gamma_i$) par rapport à Γ_{oi} est également petite, et la différence de temps de propagation entre celui observé et celui calculé pour une phase i donnée s'écrit⁵ alors :

$$\delta t_i = t_i^{obs} - t_i^{calc} \quad (2.7)$$

$$\delta t_i = \int_{\Gamma_i} s \, dl - \int_{\Gamma_{oi}} s_o \, dl \quad (2.8)$$

$$\approx \int_{\Gamma_i} (s_o + \delta s) \, dl - \int_{\Gamma_{oi}} s_o \, dl \quad (2.9)$$

Comme $\delta\Gamma_i \ll \Gamma_i$, on a $\Gamma_i \approx \Gamma_{oi} + \delta\Gamma_i$, il s'en suit que :

$$\int_{\Gamma_i} (s_o + \delta s) \, dl \approx \int_{\Gamma_{oi}} (s_o + \delta s) \, dl + \int_{\delta\Gamma_i} (s_o + \delta s) \, dl \quad (2.10)$$

$$\approx \int_{\Gamma_{oi}} s_o \, dl + \int_{\Gamma_{oi}} \delta s \, dl + \int_{\delta\Gamma_i} s_o \, dl + \int_{\delta\Gamma_i} \delta s \, dl \quad (2.11)$$

Le dernier terme de l'expression (2.11) est de second ordre, il est par conséquent négligeable. De plus d'après le principe de Fermat⁶, l'intégrale le long de la petite perturbation du chemin de l'onde est nulle (avant dernier terme), et l'on obtient :

$$\delta t_i \approx \int_{\Gamma_{oi}} s_o \, dl + \int_{\Gamma_{oi}} \delta s \, dl - \int_{\Gamma_{oi}} s_o \, dl \quad (2.12)$$

$$= \int_{\Gamma_{oi}} \delta s \, dl \quad (2.13)$$

Ainsi, étant donné un modèle de vitesse de référence $v_o(r)$, le problème consistant à trouver la fonction $v(x)$ dans l'équation (2.3) se ramène à un problème où la perturbation de vitesse δs est l'inconnue de l'équation (2.13).

Dans une phase dénommée *paramétrisation*, on fait passer l'équation (2.13) d'un système physique continu à un modèle discret.

2.1.3 Paramétrisation

La paramétrisation nécessite de choisir une base orthonormale constituée de m fonctions $e_j(x)$ avec $j = 1, \dots, m$ qui permettent de définir un modèle quelconque de lenteur dans la

⁵Pour éviter de surcharger les équations, les paramètres x et r des fonctions $s()$ et $s_o()$ sont omis dans la suite du texte.

⁶Selon le principe de Fermat, le temps de propagation de l'onde est stationnaire par rapport à une petite perturbation de son trajet.

Terre. Celui-ci s'écrit alors :

$$\delta s(x) = \sum_{j=1}^m c_j e_j(x) \quad (2.14)$$

où le coefficient⁷ $c_j = \langle \delta s(x), e_j(x) \rangle$ représente le poids de la fonction $e_j(x)$ dans la perturbation de lenteur $\delta s(x)$. L'équation (2.13) devient donc :

$$\delta t_i = \int_{\Gamma_{oi}} \sum_{j=1}^m c_j e_j dl \quad (2.15)$$

$$= \sum_{j=1}^m c_j \int_{\Gamma_{oi}} e_j dl \quad (2.16)$$

En définissant la matrice A_{ij} par :

$$A_{ij} = \int_{\Gamma_{oi}} e_j dl \quad (2.17)$$

On obtient :

$$\delta t_i = \sum_{j=1}^m A_{ij} c_j \quad (2.18)$$

Soit en notation matricielle :

$$\delta t = A c \quad (2.19)$$

$$\begin{pmatrix} \delta t_1 \\ \vdots \\ \delta t_n \end{pmatrix} = A \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix} \quad (2.20)$$

où n est le nombre total de rais et m le nombre de fonctions de base utilisées.

La matrice A est de dimension $n \times m$. Elle est appelée matrice des dérivées de Fréchet [YAMAMURO 74]. Notons qu'en tomographie sismique le nombre de rais n est beaucoup plus important que le nombre de fonctions de bases m utilisées.

On distingue deux types principaux de paramétrisation :

- la paramétrisation par polyèdre,
- et celle par harmonique sphérique.

⁷La notation $\langle f, g \rangle$ représente le produit scalaire de $f(x)$ par $g(x)$ dans $L_2(E)$, tel que $\langle f, g \rangle = \int_E f(x)g(x) dx$, où $x \in \mathbb{R}^3$.

Chacune des ces méthodes est adaptée à un contexte bien particulier. De façon générale, la paramétrisation par polyèdre est plutôt utilisée lorsqu'il s'agit d'étudier des variations de vitesse relativement localisées. Pour des variations à plus grande échelle, il est plus approprié d'utiliser les harmoniques sphériques. Ainsi [VAN DER HILST *et al.* 97] ont utilisé une paramétrisation par polyèdre (de forme hexaédrique) de façon à étudier les mouvements de convection dans le manteau, alors que [DZIEWONSKY 84] utilise plutôt les harmoniques sphériques pour étudier les variations latérales de vitesse dans le manteau inférieur.

Paramétrisation par harmoniques sphériques

Les harmoniques sphériques $Y_l^m(\theta, \phi)$ sont des fonctions définies sur la sphère unité (noté S), telles que :

$$Y_l^m(\theta, \phi) = k_l^m P_l^m(\cos \theta) e^{im\phi} \quad (2.21)$$

où $\theta \in [0, \pi]$ est l'angle correspondant à la colatitude⁸, $\phi \in [0, 2\pi]$ celui correspondant à la longitude, P_l^m est le polynôme associé de Legendre de degré l et d'ordre m avec $|m| \leq l$, k_l^m est un facteur de normalisation, et $e^{im\phi} = \cos(m\phi) + i \sin(m\phi)$.

Les harmoniques sphériques forment une famille orthonormale qui permet de construire une base sur $L_2(S)$. Ainsi, de façon similaire aux séries de Fourier, une fonction *surfactive* $f : S = \{(\theta, \phi) \in \text{surface de la sphère}\} \rightarrow \mathbb{R}$ peut-être exprimée comme étant une combinaison linéaire d'harmoniques sphériques :

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l C_{l,m} Y_l^m(\theta, \phi) \quad (2.22)$$

où $C_{l,m}$ est une constante.

Le passage à une fonction *volumique* $f : E = \{(r, \theta, \phi) \in \text{Terre}\} \rightarrow \mathbb{R}$ requiert l'introduction d'une base supplémentaire de fonctions radiales (qui ne dépendent que de la profondeur r), notées $H_u(r)$. Dans le cas qui nous intéresse, la fonction à développer en harmoniques sphériques est la variation de lenteur δs par rapport à un modèle de référence. En approximant δs par un nombre fini d'harmoniques sphériques on obtient :

$$\delta s(r, \theta, \phi) = \sum_{u=0}^U \sum_{l=0}^L \sum_{m=-l}^l C_{u,l,m} H_u(r) P_l^m(\cos \theta) e^{im\phi} \quad (2.23)$$

où $C_{u,l,m}$ est une constante. La fonction radiale $H_u(r)$ est habituellement un polynôme de Legendre ou de Chebyshev⁹ de degré u .

⁸En coordonnées sphérique, la colatitude θ est l'angle complémentaire de la latitude λ , $\theta = 90^\circ - \lambda$.

⁹Le polynôme de degré n de Chebyshev est défini par la fonction $T_n(x) = \cos(n \arccos(x))$, pour $x \in [-1, 1]$.

Notons que plus les valeurs de U et de L sont grandes plus le nombre de coefficients $C_{u,l,m}$ est élevé $[(U + 1) \sum_{m=0}^L (2m + 1)]$ et meilleure est la précision du modèle. Les coefficients de la matrice des dérivées de Fréchet (définis par la relation 2.17) s'obtiennent alors en calculant l'intégrale suivante pour chaque rai i :

$$\int_{\Gamma_{oi}} H_u(r) P_l^m(\cos \theta) e^{im\theta} dl \quad (2.24)$$

Paramétrisation par polyèdres

Elle décompose la Terre, ou le cas échéant la zone d'étude locale, en une série de polyèdres, appelés aussi cellules, qui ne s'intersectent pas. Les polyèdres les plus communément utilisés sont les hexaèdres, ou blocs, [VAN DER HILST *et al.* 97] et les tétraèdres [SAMBRIDGE & FALETIC 03]. Les fonctions de base e_j peuvent alors prendre, dans le cas le plus simple, la forme suivante :

$$e_j(x) = \begin{cases} 1 & \text{si } x \in \text{cellule } j \\ 0 & \text{sinon} \end{cases} \quad (2.25)$$

Dans un tel cas les éléments de la matrice A sont les longueurs des segments de rais dans chaque cellule du modèle. Ainsi, A_{ij} qui définit la longueur du $i^{\text{ème}}$ rai à travers la $j^{\text{ème}}$ cellule s'écrit :

$$A_{ij} = \int_{\Gamma_{oi} \cap \text{cell}_j} dl \quad (2.26)$$

On peut remarquer que pour un rai i donné, la plupart des coefficients A_{ij} (pour $j = 1, \dots, m$) sont nuls, puisque le rai i ne traverse qu'un petit nombre de cellules du modèle (cf. figure 2.5 où chaque cellule est caractérisée par sa longueur s_j).

Dans ce travail, nous utilisons une paramétrisation par polyèdre, basée sur un maillage *hexaédrique* dont la construction est présentée au chapitre 3.

2.1.4 Méthodes de tracé de rais

Afin de pouvoir calculer les différents coefficients A_{ij} pour un rai i donné, il est nécessaire d'avoir les coordonnées géographiques d'un certain nombre de points de son trajet reliant la source du séisme au capteur sismologique. Cette opération, appelée *tracé de rais sismiques* ou encore *problème direct*, est une étape indispensable de la tomographie. Elle est généralement coûteuse en temps de calcul.

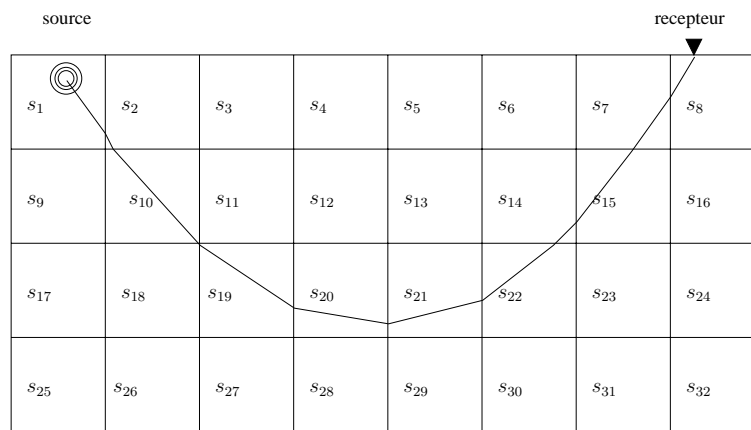


FIG. 2.5 – Représentation d’un rai à travers un modèle 2D composé de blocs.

Dans les programmes de tracé de rais utilisés en tomographie, on trouve principalement quatre types d’algorithmes :

1. le *ray shooting*,
2. le *shortest path ray tracing*, ou méthode du plus court chemin dans un graphe,
3. le *ray bending*,
4. les différences finies,

qui sont basés sur la loi de Descartes (pour la méthode 1) [DESCARTES 37], le principe de Fermat (méthodes 2 et 3) [FERMAT 91] et de Huygens (méthode 4) [HUYGENS 90].

Toutes ces méthodes nécessitent d’avoir, pour un rai à calculer, un modèle de vitesse de propagation des ondes sismiques, ainsi que les coordonnées géographiques de la source du séisme et du capteur.

Le *ray shooting*

Dans la méthode de *ray shooting*, à partir d’un angle d’incidence initial situé au foyer du séisme, le rai est propagé pas à pas en utilisant la loi de Descartes. Cependant, rien ne permet de dire que le point final du rai tracé coïncide avec le récepteur. C’est pourquoi il peut être nécessaire de faire varier l’angle d’incidence initial (par exemple par une approche dichotomique), jusqu’à ce que le point final du rai coïncide avec le récepteur (dans une certaine marge d’erreur acceptable).

La méthode du *ray shooting* est celle utilisée dans ce travail, et elle est présentée plus en détail dans le chapitre 3.2.

Le *shortest path ray tracing*

D'après le principe de Fermat, le trajet parcouru par un rai entre deux points est celui qui minimise ou qui maximise le temps de parcours (principe de stationnarité). En sismologie le trajet du rai est celui qui a le temps de parcours minimum.

Le *shortest path ray tracing* est une méthode dont l'origine provient de la théorie des graphes [DIJKSTRA 59] qui a été employée par [NAKANISHI & YAMAGUCHI 86] et [MOSER 91]. On construit un graphe, dont les nœuds sont distribués dans le modèle. Chaque nœud est connecté via un arc à un nombre restreint de nœuds de son entourage immédiat. Il est alors possible de se déplacer d'un nœud à un autre par les arcs. On affecte comme poids associé à un arc, le temps de parcours du rai calculé entre ses deux nœuds.

Le *shortest path ray tracing* consiste alors, à trouver le chemin le plus court dans ce graphe. Le degré de précision dans le tracé du rai augmente avec la densité de nœuds du graphe.

Le *ray bending*

La méthode de calcul du trajet d'un rai par *ray bending* consiste, à partir d'un rai initial non optimal dont on a fixé l'origine et l'extrémité (comme par exemple une ligne droite ou un arc de cercle joignant la source et le récepteur), à le perturber par itérations successives, de façon à minimiser son temps de trajet [W. A. PROTHERO *et al.* 88]. Les perturbations du rai sont réalisées en utilisant une famille de fonctions sinus dont les amplitudes sont déterminés à chaque itération par l'algorithme de minimisation du simplexe [NELDER & MEAD 65].

Le tracé de rais par différences finies

La méthode par différences finies [VIDALE 90] se base sur l'équation de l'*eikonal* (2.1), et sur une grille rectilinéaire où l'on attribue à chaque nœud une lenteur. La source du séisme, considérée comme ponctuelle, est située sur l'un des nœuds.

La propagation de l'onde se fait en deux étapes. Elle démarre du nœud source en allant vers les nœuds adjacents à qui l'on affecte un temps de propagation de l'onde par interpolation linéaire de la lenteur. Ensuite, le temps de propagation associé à chaque nœud du voisinage du nœud source se calcule à partir de l'équation (2.1) et des temps déjà calculés de ses nœud adjacents.

A l'issue de cette phase, chaque nœud ainsi traité est considéré comme une nouvelle source et la construction de l'onde se poursuit sur le même schéma que précédemment.

Ainsi de façon itérative le front d'onde est propagé de proche en proche sur l'ensemble des nœuds.

2.1.5 Solution tomographique

En pratique en tomographie sismique, le système d'équations à résoudre (2.19), $\delta t = A c$, est surdéterminé puisque dans la matrice A le nombre de rais (*ie.* le nombre de lignes n) est beaucoup plus grand que le nombre de fonctions de base utilisées (*ie.* le nombre de colonnes m). De plus la matrice A est creuse puisque certaines régions du globe ne sont pas échantillonnées par les rais. C'est pourquoi ce système d'équations est mal posé, et doit être résolu par approximation. La méthode couramment utilisée est celle des moindres carrés qui détermine une solution c en minimisant l'expression :

$$\min \|A c - \delta t\|^2 \quad (2.27)$$

En dérivant par rapport à c l'expression (2.27), et en considérant que l'expression résultante est nulle à son minimum, on obtient le résultat suivant (cf. [TREFETHEN & BAU 97]) :

$$A^T A c = A^T \delta t \quad (2.28)$$

connu sous le nom d'*équations normales*.

Dans le cas où la matrice $(A^T A)$ n'est pas singulière, on a :

$$c = (A^T A)^{-1} A^T \delta t \quad (2.29)$$

où $(A^T A)^{-1}$ est appelée la matrice *inverse généralisée* de A .

Notons toutefois que dans la plupart des problèmes inverses de tomographie, la matrice $(A^T A)$ est presque singulière, ce qui signifie que de petites perturbations de A peuvent provoquer de grandes variations dans $(A^T A)$, et donc dans la solution c . Dans un tel cas, une phase de *régularisation* est nécessaire pour déterminer une solution stable.

2.1.6 Régularisation

La régularisation consiste à enrichir le système d'équations initial (2.19) par un ensemble d'équations qui visent à contraindre la solution afin qu'elle soit pertinente du point de vue géophysique. Ces équations ajoutées correspondent à deux objectifs distincts.

Le premier consiste à faire tendre la solution vers zéro, c'est-à-dire vers un modèle de vitesse sans aucune perturbation par rapport au modèle de vitesse de référence. Cette étape, appelée *minimum norm damping* se traduit par l'ajout du système d'équations suivant :

$$\lambda_1 I c = 0, \quad (2.30)$$

où λ_1 est une constante et I est la matrice identité de dimension $(m \times m)$.

Le deuxième objectif vise à garantir l'obtention d'un modèle de vitesse réaliste. Pour cela on impose, dans une phase de *gradient damping*, qu'il ne peut y avoir de variation "trop" brusque de la lenteur (et donc de la vitesse) localement.

Dans un modèle paramétrisé par des polyèdres, cette contrainte exprime l'absence de variation brusque de la lenteur entre deux cellules adjacentes du modèle. Le système d'équations associé à cette contrainte s'écrit :

$$\lambda_2 D c = 0 \quad (2.31)$$

où λ_2 est une constante et D la matrice reliant la lenteur de chaque cellule à celle de l'un de ses voisins telle que :

$$D_{ij} = \begin{cases} 1 & \text{si } j = i \\ -1 & \text{si } j \text{ est voisin de } i \\ 0 & \text{autrement} \end{cases} \quad (2.32)$$

Si N_i est le nombre de voisins de la cellule i (au sens de la connexité par les faces on a $N_i \leq 4$ pour un tétraèdre et $N_i \leq 6$ pour un hexaèdre), alors la taille de la matrice D est $(\sum_{i=1}^m N_i) \times m$, m étant le nombre de cellules.

Le système d'équations résultant s'écrit alors :

$$\begin{pmatrix} A \\ \lambda_1 I \\ \lambda_2 D \end{pmatrix} c = \begin{pmatrix} \delta t \\ 0 \\ 0 \end{pmatrix} \quad (2.33)$$

Habituellement les valeurs des constantes λ_1 et λ_2 sont déterminées de façon empirique. Plus ces constantes sont importantes et plus les relations linéaires associées prennent du poids lors de la résolution du système.

Étant donné que la régularisation dépend des paramètres λ_1 et λ_2 , plusieurs inversions sont à réaliser, en faisant varier ces deux paramètres, avant de considérer une solution comme étant satisfaisante. Il est donc nécessaire d'avoir un critère qui permette de sélectionner la bonne solution parmi celles calculées. On définit pour cela, la *réduction de variance* vr par :

$$vr = 1 - \frac{\|A c_f - \delta t\|^2}{\|\delta t\|^2} \quad (2.34)$$

où c_f est la solution finale. Cette réduction de variance estime la distance entre les deux vecteurs $A c_f$ et δt . Elle est exprimée en pourcent, et plus cette valeur est importante plus la solution c_f est proche de la solution optimale. Notons que dans la pratique, en tomographie sismique, la réduction de variance est habituellement comprise entre 10% et 50%.

2.1.7 Résolution du problème numérique

Plusieurs méthodes sont potentiellement utilisables pour résoudre un système d'équation linéaire tel que le système (2.33). On trouve notamment les méthodes directes comme Cholesky [BOSCHI 03] ou la décomposition SVD [MARQUES *et al.* 03], ou les méthodes itératives comme l'*Algebraic Reconstruction Technique* (ART) [GORDON *et al.* 70], la *Simultaneous Iterative Reconstruction Technique* (SIRT) [GILBERT 72], ou le gradient conjugué [HESTENES 80].

Nous utilisons dans ce travail la méthode itérative LSQR¹⁰[PAIGE & SAUNDERS 82], une variante de la méthode du gradient conjugué, pour résoudre ce système d'équations. Avec cette méthode, la qualité de la solution augmente avec le nombre d'itérations en tendant vers la solution exacte des moindres carrés. Elle permet en outre, à chaque itération, d'évaluer si la solution calculée est acceptable en se basant sur un critère d'arrêt.

2.2 Les architectures parallèles et les grilles

La définition des architectures parallèles couvrent un spectre plutôt large, allant du parallélisme au niveau des microprocesseurs¹¹ jusqu'aux immenses réseaux d'ordinateurs hétérogènes distribués à l'échelle d'internet : les grilles. Nous nous intéressons ici plus particulièrement aux architectures de machines parallèles, définies comme étant un ensemble de processeurs qui coopèrent et communiquent.

L'objectif de cette partie est donc de présenter de façon générale les différentes architectures parallèles et les grilles que l'on peut trouver aujourd'hui pour exécuter différents type d'applications notamment en calcul scientifique.

2.2.1 Classification des architectures parallèles

Selon la classification de Flynn [FLYNN 66], qui est couramment utilisée, on distingue quatre types principaux de parallélisme : SISD, SIMD, MISD et MIMD. Cette classification est basée sur les notions de flot de contrôle (renseignée par les deux premières lettres, le I voulant dire *Instruction*) et flot de données (deux dernières lettres, le D voulant dire *Data*).

¹⁰LSQR : Least Square avec factorisation QR

¹¹La plupart des micro-processeurs modernes sont à eux seuls de véritables machines parallèles. Par exemple il n'est pas rare (Pentium, PowerPC etc.) d'avoir plusieurs unités de calcul arithmétique pouvant fonctionner en parallèle (dans le cas du Pentium, d'unités de calcul flottant, ou dans le cas des processeurs MIPS, plusieurs additionneurs, multiplieurs, etc.).

Au delà de cette classification qui n'intègre pas certaines architectures récentes on peut classifier les différentes architectures selon le mode d'agencement de la mémoire : les architectures à mémoire partagée, celles à mémoire distribuée et enfin les architectures à mémoire hybride. Les architectures à mémoire partagée présentent l'avantage de permettre un partage immédiat des données, facilitant de ce fait la programmation. Cependant dans cette solution, le nombre de processeurs associés à une même mémoire est assez limité pour des raisons de performances, puisqu'ils se partagent le bus de communication et sont en concurrence pour les accès mémoire. En revanche les architectures à mémoire distribuée ou hybride ont l'avantage de permettre une hausse des performances processeurs/mémoires plus intéressante que dans le cas des architectures à mémoire partagée, mais en contrepartie c'est à la charge du programmeur de gérer les détails de la communication entre les unités de calcul.

Dans les *architectures à mémoire partagée*, on distingue celles qui le sont physiquement, et celles qui le sont logiquement. Dans les *architectures à mémoire physiquement partagée*, l'accès à la mémoire est le même pour chaque processeur (ils sont tous identiques). On parle alors de mémoire à accès uniforme (UMA¹²). Un contrôleur de cache s'assure que la mémoire cache de chaque processeur est cohérente avec les données présentes en mémoire centrale. Ce type de mémoire est présent dans la plupart des systèmes SMP¹³, ou dans les systèmes à processeurs multi-cores. Dans les *architectures à mémoire logiquement partagée*, chaque processeur dispose de sa propre mémoire qui lui est rattachée directement, mais il peut accéder via un réseau dédié et de manière tout à fait transparente pour l'utilisateur, à la mémoire vive de chacun des autres processeurs. Les accès mémoire peuvent donc être de vitesse différente selon que les données sont présentes dans la mémoire locale ou non. On parle alors de *mémoires à accès non uniforme* (NUMA¹⁴). Comme pour les architectures de type UMA, on peut y rajouter un système de cohérence de cache (ccNUMA). On trouve par exemple ce type de mémoire dans les machines Origin de Silicon Graphics.

Dans le cas de la *mémoire distribuée* (ou systèmes répartis), chaque processeur possède sa propre mémoire locale privée. La modification de celle-ci par son processeur n'a pas d'influence directe sur celles des autres processeurs. Cela suppose donc de mettre en place une communication explicite entre les processeurs, souvent par l'intermédiaire d'un réseau, pour transférer de l'information.

Les *architectures à mémoire hybride* utilisent à la fois de la mémoire partagée et de la mémoire distribuée. Ces architectures sont constituées de plusieurs groupes de processeurs, partageant de la mémoire, qui communiquent grâce à un réseau. Par exemple, les clusters de machines SMP font partie de cette catégorie.

¹²UMA : Uniform Memory Access

¹³SMP : Symetric Multi-Processor

¹⁴NUMA : Non Uniform Memory Access

2.2.2 Les grilles

L'accroissement important des performances des processeurs, des réseaux et des disques a fait progresser, dans la dernière décennie, l'idée de relier des ressources de calcul ou de stockage distantes géographiquement, dans un objectif de mutualisation de ces ressources. Cette idée a été popularisée sous le terme de *grille*, notamment par [FOSTER & KESSELMAN 98].

Aujourd'hui, de nombreux projets proposent des infrastructures de type grille, aux caractéristiques souvent hétérogènes sur lesquelles le comportement des applications est difficile à prédire. Plusieurs projets ont été développés ces dernières années pour déployer des infrastructures de grille dédiées à grande échelle : DAS-3 [DAS] aux Pays-Bas, TeraGrid [TER] aux USA et Grid'5000 [CAPPELLO 05] en France. Ces projets permettent l'accès à des milliers de CPU via des réseaux haut-débit.

Actuellement les grilles proposent des services permettant de vérifier les identités des utilisateurs du service, de placer automatiquement des exécutables sur les ressources de calcul distantes (*executable staging*), d'unifier les différents flots d'entrées/sorties, et de synchroniser le déclenchement des exécutables sur les ressources de calcul. Dans le futur, on disposera probablement en standard d'ordonnanceurs à l'échelle d'une grille, capables d'utiliser un service d'annuaire pour localiser les ressources les plus adéquates et de planifier l'exécution parmi d'autres travaux.

D'un point de vue technique, l'exécution efficace d'une application parallèle sur une grille amène des questions difficiles, singulièrement plus compliquées que pour l'exécution sur les machines parallèles citées précédemment. En effet, obtenir le temps d'exécution le plus court implique de choisir les "meilleurs" processeurs parmi un ensemble hétérogène, tant du point de vue des processeurs, de la RAM et des disques, que de leurs liaisons réseaux.

Dans le cadre de cette thèse nous avons utilisé une grille de test et nous avons pu dès 2002, procéder à des comparaisons et tirer des conclusions sur les performances que nos différentes applications pouvaient atteindre à l'heure actuelle sur ce type d'infrastructure. Cette grille est construite sur le réseau Renater et l'intergiciel Globus [FOSTER & KESSELMAN 97]. Les applications utilisent pour MPI l'implémentation mpich-g2 [FOSTER & KARONIS 98]. Les ressources de calcul mutualisées sont des machines parallèles et des stations de travail.

Les observations faites, concernant le comportement de nos applications sur cette grille de test, ne peuvent pas avoir de valeur générale. En effet, les ressources d'une grille peuvent être variées à l'infini, et pour un ensemble fixé de ressources, leurs caractéristiques varient dans le temps. Cependant, l'évolution des performances des réseaux notamment, au cours de nos expériences, nous ont montré que les grilles sont aujourd'hui des plate-formes concurrentes des machines parallèles dans certains cas.

2.2.3 Programmation des applications parallèles

Des approches variées ont été proposées pour écrire un programme capable d'exploiter de telles architectures parallèles. Plus encore que pour la programmation des machines séquentielles, le choix d'un modèle de programmation dépend du compromis que le programmeur est prêt à accepter entre performance et facilité d'expression des programmes. Depuis le début du parallélisme coexistent des approches opposées : d'une part des modèles de programmation proches du fonctionnement sous-jacent de la machine, permettent l'exécution efficace d'un programme mais au prix d'une conception, mise au point et maintenance difficiles. Par exemple, le langage OCCAM [HOARE 85] qui implémente la théorie de CSP (Communicating Sequential Processes) [HOARE 78], permet l'exécution concurrente de processus séquentiels se synchronisant par rendez-vous, envoi et réception de messages. L'implémentation d'OCCAM sur des machines de type MIMD est donc très simple, car le modèle de programmation est presque le modèle d'exécution. D'autre part, des modèles de programmation plus abstraits ont également été proposés très tôt. Ils permettent au programmeur de s'abstraire de certains détails liés à l'architecture de la machine, permettant d'améliorer la portabilité du code mais en contrepartie se reposant sur le compilateur pour générer du code efficace. Le langage Linda [GELERTER 85] en est un exemple précoce : les processus en parallèle communiquent et se synchronisent via des écritures ou lectures sur un support commun, le *tuple space* (ou tableau noir), qu'on peut considérer comme une mémoire partagée logique évoluée (mémoire associative, lectures destructives ou non).

Dans les dix dernières années, les modèles de programmation les plus utilisés pour la programmation parallèle n'ont pas beaucoup évolué vers plus d'abstraction¹⁵. Les modèles proches du modèle d'exécution ont prédominé, avec le succès de PVM [SUNDERAM 90] puis MPI [MPI 95], qui sont devenus le standard pour la programmation par passage de messages. Le succès de MPI tient à la façon dont il a été élaboré : un forum d'utilisateurs a défini une spécification, faisant l'objet d'une implémentation simultanée pour en vérifier la faisabilité. L'existence d'une spécification et la simplicité du modèle de programmation a permis son développement sur de multiples plate-formes et garantit donc la portabilité des programmes l'utilisant. Cependant, si MPI garantit la pérennité des développements qui l'utilisent, ce mode de programmation est difficile. Parmi ces difficultés, les plus flagrantes sont la correction des échanges de messages qui est difficile à établir (pouvant mener à des verrous mortels), et la gestion de la mémoire distribuée qui est lourde (obligation d'écrire explicitement les communications correspondantes).

Le modèle de programmation *data-parallèle*, plus abstrait que le modèle à passage de message, a donc logiquement suscité beaucoup d'intérêts comme modèle de programmation alternatif. Dans ce modèle, le programmeur déclare dans son programme des *variables parallèles*, dont l'utilisation dans des instructions déclenche implicitement des traitements

¹⁵On peut noter que même les langages comme Linda ont été mis au goût du jour à travers les JavaSpaces [FREEMAN *et al.* 99].

parallèles sur chaque composante de la variable parallèle. Le compilateur, aidé ou non d'indications du programmeur, distribue les composantes des variables parallèles sur la ou les mémoires disponibles. Après plusieurs propositions de langages data-parallèles (comme C* [CST 90]), un forum a été créé pour définir, à l'instar du procédé d'élaboration de MPI, la spécification HPF [HPF 93, KOELBEL *et al.* 94], basé sur Fortran 90. Cependant, de nombreuses difficultés sont apparues pour compiler ce langage, et le succès escompté n'a pas été au rendez-vous.

Des approches alternatives ont été proposées pour simplifier les développements parallèles tout en permettant une compilation efficace du code. C'est le cas d'OpenMP [DAGUMI & MENON 98], standardisé en 1997. Pour le programmeur, l'avantage est de pouvoir partir d'un code séquentiel et d'ajouter petit à petit ses directives¹⁶ en donnant des indications de parallélisation au compilateur. Ainsi, ajouter une directive disant que les itérations d'une boucle sont indépendantes permet d'exécuter les itérations en parallèle. Le parallélisme se traduit par l'exécution de threads. Cependant, OpenMP repose sur l'existence d'une mémoire partagée. Or, même si des mémoires partagées peuvent être émulées logiciellement sur des architectures à mémoire distribuée, l'extensibilité de celles-ci n'est pas acquise et l'utilisation d'OpenMP se restreint aujourd'hui principalement à des machines parallèles à mémoire partagée. Notons que les travaux de recherche actifs pour l'exécution d'OpenMP sur mémoire distribuée, ont débouché récemment sur des produits prometteurs comme Cluster OpenMP [HOEFLINGER 06] d'Intel.

Enfin, UPC [EL-GHAZAWI *et al.* 05] est le dernier langage parallèle en date à avoir suscité un large engouement. Plusieurs universités et entreprises majeures de l'industrie informatique (dont IBM, HP, Cray) ont écrit des compilateurs UPC. UPC tente de faire le pont entre la facilité de programmation qu'offre la mémoire partagée, et la possibilité de contrôler la répartition des données dans une mémoire distribuée. Comme avec OpenMP, le modèle d'exécution d'UPC est basé sur des threads potentiellement distribués. Les variables déclarées partagées (shared) par le programmeur, notamment les tableaux, sont alors distribués à la compilation selon la distribution des threads. La mémoire est ainsi vue comme un espace global partitionné, chaque thread ayant une "affinité" (localité) avec une partition, et donc certaines composantes des variables partagées. À ce point de vue très proche du modèle data-parallèle, UPC ajoute des primitives inspirées du modèle à passage de messages, comme les barrières, pour contrôler le flux d'exécution.

Plus récemment, la gamme des modèles de programmation s'est élargie avec la possibilité d'exécuter une application parallèle sur un ensemble distribué à large échelle de processeurs, les grilles de calcul. De nombreux prototypes ont été proposés ces dernières années pour programmer ce type d'architecture. Pour la plupart, ils utilisent des modèles de programmation issus des systèmes distribués. La plupart des prototypes proposés sont basés sur le modèle *Remote Procedure Call* (RPC). Certains intergiciels (middlewares) de

¹⁶Les directives apparaissent sous la forme syntaxique de commentaires. Ainsi, elles sont ignorées par un compilateur ordinaire, autorisant une exploitation séquentielle du même code.

grille (par exemple DIET [CARON *et al.* 02] ou XtremWeb [FEDAK *et al.* 01]) proposent, soit des modèles basés sur des surcouches de Corba [COR 03] ou JavaRMI, soit des adaptations de RPC, comme GridRPC [NAKADA *et al.* 02]. Les applications de ce type sont plus robustes (un module distribué dépend peu d'autres modules, et peut être remplacé en cas de panne), et sont plus faciles à maintenir et à déployer (principe des composants). En revanche, ils n'ont pas l'expressivité du parallélisme et donc la performance des langages parallèles. D'autres propositions tentent de remédier à la perte d'expressivité du parallélisme en intégrant le parallélisme au sein d'objets distribués communiquant avec Corba, comme dans GridCCM [PÉREZ *et al.* 03]. Enfin, d'autres approches proposent de conserver un modèle de programmation de type passage de message, tout en assignant à un système sous-jacent (par exemple l'intergiciel) la tâche de rendre l'exécution de l'application robuste ([BOUTEILLER *et al.* 03] ou [GENAUD & RATTANAPOKA 05] en sont deux illustrations).

L'exploitation de programmes parallèles sur des systèmes distribués à large échelle pose de réels problèmes non-encore résolus à l'heure actuelle, dans lesquels l'intergiciel et le modèle de programmation doivent coopérer pour offrir au programmeur un environnement viable de développement.

2.3 Les maillages

Nous présentons dans cette partie, de façon assez générale, les principales techniques qui permettent de construire un maillage 3D. Pour une présentation complète du sujet, on se référera aux ouvrages de [FREY & GEORGE 99], [THOMPSON *et al.* 85], et [CAREY 97]. Après quelques définitions, nous aborderons notamment les maillages structurés et non structurés, ainsi que les méthodes de maillage adaptatif. Enfin nous verrons quel type d'approche il est possible d'avoir pour paralléliser la construction d'un maillage ainsi que la résolution du problème numérique associé.

2.3.1 Définition d'un maillage en 3D

Avant de poursuivre plus en avant, il est nécessaire de définir ce qu'est un maillage. Un maillage est constitué d'un ensemble d'éléments géométriques formés en 3D, de sommets, d'arêtes et de faces, qui partitionne un domaine délimitant les données à analyser. Il est défini par sa métrique et sa topologie. La métrique est l'ensemble des coordonnées des sommets de ses éléments, auxquels sont associées, le cas échéant, une ou plusieurs valeurs physiques. La topologie correspond à la définition de la connectivité entre les différents sommets, ainsi qu'aux relations de voisinage entre les éléments constitutifs du maillage.

On dit qu'un maillage est *conforme* lorsque l'ensemble de ses éléments constitutifs recouvre le domaine, et que toute intersection entre deux éléments du maillage est réduite soit à l'ensemble vide, soit à un sommet, soit à une arête, soit à une face, pour un maillage en 3D.

2.3.2 Génération de maillage structuré

Un maillage structuré est construit de telle sorte que la connectivité est implicite, ce qui signifie que chaque sommet possède le même nombre de voisins et que ceux-ci peuvent être facilement retrouvés grâce à un système d'index. On distingue dans la littérature plusieurs méthodes pour obtenir des maillages structurés.

Les maillages structurés réguliers sont basés directement sur des systèmes de coordonnées classiques tels que les systèmes cartésien ou polaire par exemple. Le principal problème de ces maillages est qu'ils peuvent, dans certains cas, ne pas épouser les frontières particulières d'un domaine.

Les méthodes algébriques permettent de construire le maillage d'un domaine *réel* à partir du maillage structuré d'un domaine *logique* facile à construire. Cette opération se fait en utilisant une fonction de projection déterminée par une méthode de type interpolation transfinie, qui peut être vue comme une distorsion topologique, dont une synthèse est présentée dans [PERRONNET 98]. Cette fonction de projection est en général polynomiale.

D'autres méthodes existent pour générer des maillages structurés. Par exemple celle basée sur la résolution d'un système d'équations aux dérivées partielles [THOMPSON *et al.* 85]. L'équation de Laplace avec les conditions aux limites de Dirichlet, est fréquemment utilisée dans ce cas. Les maillages obtenus par cette technique possèdent alors certaines propriétés intéressantes telles que le non chevauchement de leurs mailles ainsi que l'aspect lisse et régulier de leurs frontières. Dans le même registre, une autre possibilité consiste à utiliser l'équation de Poisson qui permet de contrôler la densité des mailles dans certaines régions critiques par exemple. La figure 2.6 montre l'exemple de deux maillages obtenus respectivement avec le système de Laplace et celui de Poisson.

Enfin, une autre méthode couramment utilisée, consiste à diviser une géométrie complexe en plusieurs blocs plus simples. Chacun de ces blocs sera ensuite maillé à l'aide de l'une des méthodes évoquée précédemment. Ce genre de technique produit des maillages appelés multi-blocs. Une attention particulière doit être portée aux frontières entre blocs afin que les mailles situées à la frontière "correspondent entre elles" (*ie.* préservent la conformité et la connectivité), faute de quoi le maillage global ne serait pas structuré.

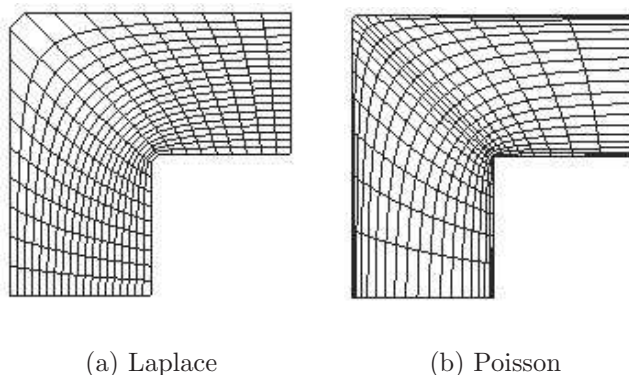


FIG. 2.6 – Exemple de maillages obtenus en résolvant le système de Laplace et celui de Poisson.

2.3.3 Génération de maillage non structuré

Un maillage est dit non structuré si sa connectivité est quelconque. Les maillages de ce type offrent une liberté supplémentaire par rapport aux maillages structurés puisqu'ils permettent de contrôler localement la concentration des mailles en fonction de la nature des données à mailler. Cependant, la connectivité entre les sommets doit être explicitement fournie ce qui implique de mettre en place des structures de données plus complexes que dans le cas des maillages structurés. Il existe plusieurs méthodes pour générer des maillages non structurés, ceux-ci pouvant être conformes ou non conformes.

Maillages conformes

On distingue principalement deux méthodes : la méthode frontale et celle de Delaunay-Voronoi.

La *méthode frontale* (*advancing front method*) consiste d'abord à discrétiser les arêtes frontières du domaine à mailler, puis à ajouter graduellement, au maillage en cours de construction (*ie.* le front), des sommets et des arêtes en avançant vers l'intérieur du domaine. La figure 2.7 montre la progression d'un front pour un cas en 2D.

La méthode de *Delaunay-Voronoi* est également très utilisée. Celle-ci permet de générer à partir d'un nuage de points, un maillage constitué de triangles en 2D et de tétraèdres en 3D. Dans cette méthode, la conformité est garantie par la propriété suivante, appelée *critère de Delaunay*, qui assure que pour chaque triangle (resp. tétraèdre) le cercle (resp. sphère) circonscrit ne contient aucun autre sommet du maillage. La figure 2.8 montre pour

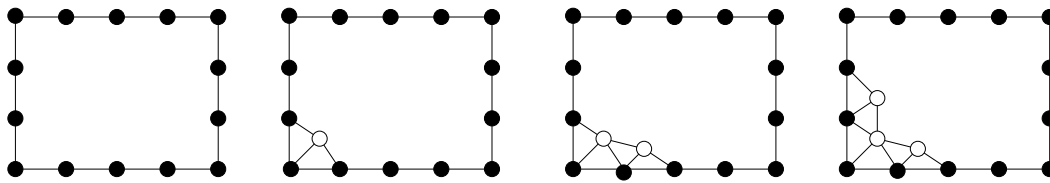


FIG. 2.7 – Construction d'un maillage en utilisant la méthode frontale.

un nuage de points donné, la construction de deux maillages, l'un ne respectant pas le critère de Delaunay, l'autre le respectant.

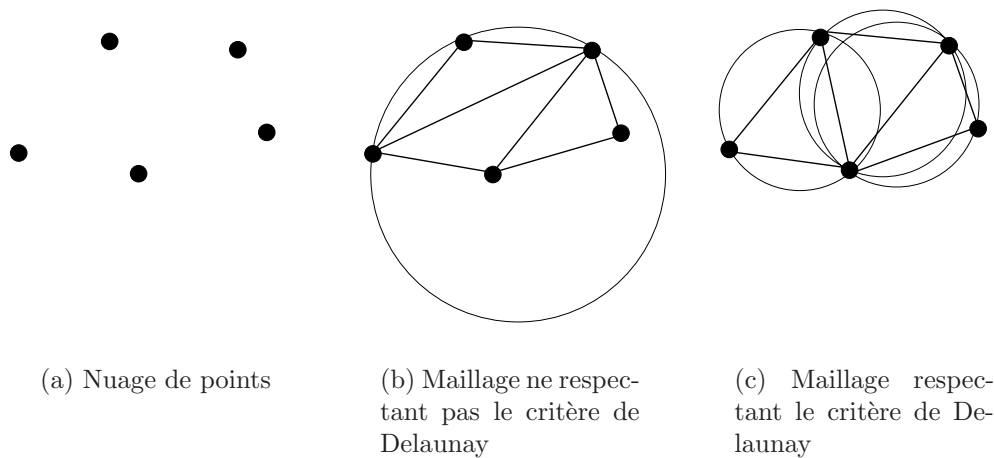


FIG. 2.8 – Construction d'un maillage à partir d'un nuage de points en utilisant la méthode de Delaunay-Voronoi.

Maillages non conformes

La méthode la plus répandue pour construire un maillage non structuré non conforme est la méthode de *décomposition spatiale*, telle que celle des *octrees* en 3D (*quadtrees* en 2D). Dans cette méthode le maillage est généré par l'union de boîtes, appelées aussi cellules, disjointes et de tailles variables constituant une partition du domaine (cf. figure 2.9). Ces cellules sont obtenues par un découpage récursif d'une cellule mère englobant l'objet à mailler en huit cellules filles en 3D (quatre en 2D). Le principe de base de la méthode est qu'au fur et à mesure que la subdivision s'affine, la propriété physique contenue dans la cellule, se simplifie.

De part sa construction par décomposition spatiale, un maillage non conforme est bien adapté au type de données qu'il manipule. A contrario, les maillages conformes obtenus par

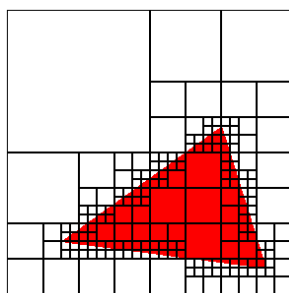


FIG. 2.9 – Construction du maillage d’un disque par la méthode des quadtrees.

les méthodes frontales ou Delaunay-Voronoi peuvent être modifiés pour mieux s’adapter localement aux données, on parle alors d’adaptation de maillage.

2.3.4 Adaptation de maillages conformes

L’adaptation d’un maillage conforme aux propriétés physiques du problème traité est une technique qui permet d’améliorer la précision numérique des solutions et de réduire les temps de calcul. Elle est liée à la distribution des sommets du maillage, ainsi qu’à la forme et à la taille de ses éléments.

Toutes les méthodes adaptatives sont basées plus ou moins sur la même approche. Celle-ci consiste tout d’abord à construire un maillage initial du domaine d’étude, selon l’une des méthodes décrites précédemment. Ensuite, une solution est calculée et analysée au moyen d’un estimateur d’erreur. Le résultat de cette analyse est traduit en spécifications de taille et servira à gouverner l’adaptation du maillage. Ainsi en tenant compte de ces spécifications, un nouveau maillage considéré comme plus adapté au problème physique traité, est construit en utilisant l’une des méthodes présentées dans ce chapitre. Une solution est alors calculée et le processus adaptatif se poursuit ou non en fonction de la nouvelle estimation de l’erreur.

Une façon de classer les procédés de modification de maillage est d’examiner si la topologie du maillage de départ est modifiée. En effet on peut distinguer les méthodes adaptatives à topologie constante et celles à topologie variable.

Méthodes à topologie constante

Elles sont aussi appelées *déformations* car les sommets sont déplacés de façon explicite (cf. figure 2.10), par résolution de système d’équations. De telles méthodes préservent la

connectivité des sommets. Cependant de sérieuses difficultés peuvent apparaître telles que des recouvrements entre les mailles, lorsque le déplacement de certains sommets est trop "important" et que la nouvelle position du sommet se trouve en dehors de l'espace défini par l'ensemble de ses cellules (*ie.* problème de conformité).

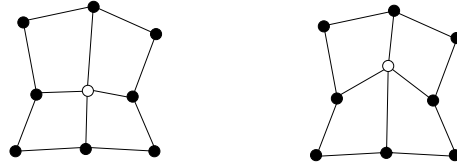


FIG. 2.10 – Exemple en 2D de déplacement d'un sommet (en blanc).

Méthodes à topologie variable

Dans les méthodes à topologie variable, des sommets peuvent être ajoutés ou supprimés, la connectivité du maillage résultant est donc modifiée. On distingue principalement trois approches :

- la *reconstruction globale* s'applique lorsque l'on dispose d'un générateur de maillage rapide. Celui-ci peut alors être intégré dans le processus d'adaptation de maillage. À chaque modification du maillage, lors d'un ajout ou une suppression de sommets, le maillage est globalement reconstruit ;
- le *raffinement* est réalisé en introduisant de nouveaux sommets dans les parties où il est nécessaire d'avoir un maillage plus fin. Ceci peut poser des problèmes de raccord avec la partie du maillage inchangée. En utilisant des méthodes locales de re-maillage ces raccords peuvent être soit conformes (Voronoi ou *advancing front* par exemple), soit non conformes (méthodes de type octree). La figure 2.11(a) montre, dans le cas 2D, un exemple de raffinement : l'arête en pointillés a été sélectionnée pour être raffinée. Un sommet est donc ajouté en son milieu et de nouvelles arêtes sont créées pour former quatre nouveaux éléments ;
- le *déraffinement* consiste à supprimer un élément du maillage lorsque localement un maillage plus grossier est suffisant. On retrouve les mêmes problèmes de reconnexion (conforme ou non conforme) que pour le raffinement. Dans l'exemple de déraffinement présenté dans la figure 2.11(b), les deux sommets de l'arête sélectionnée (en pointillés) sont fusionnés, et la position du nouveau sommet est arbitrairement choisie au milieu de l'arête déraffinée.

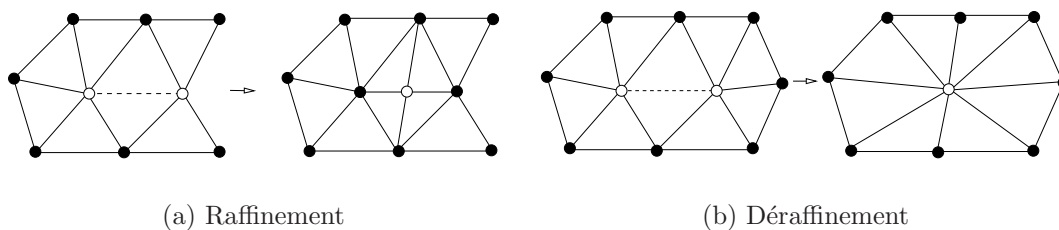


FIG. 2.11 – Exemple de raffinement et de déraffinement d’arêtes en 2D.

2.3.5 Parallélisation de maillage

Il est devenu courant de voir dans de nombreux domaines d’applications, tels que celui de la dynamique des fluides, ou bien celui de la propagation d’ondes dans un milieu complexe, des maillages de très grande taille qui comprennent plusieurs millions d’éléments. La génération de tels maillages, ainsi que la résolution du problème numérique sur celui-ci, pose des difficultés notamment en termes de rapidité et de capacité mémoire en raison de la quantité et du volume de données à traiter.

Devant la multiplication des ressources de calculs disponibles, telles que celles présentées dans le paragraphe 2.2, il est nécessaire de développer de nouveaux algorithmes de maillage parallèles pour tirer parti de ces différentes architectures. On trouvera une présentation des méthodes les plus couramment utilisées dans [CHRISOCHOIDES 05].

Les méthodes de parallélisation de maillage se basent le plus souvent sur des algorithmes de maillage séquentiel qui ont déjà fait leurs preuves (ré-utilisation de code). Elles consistent à décomposer le problème initial de la génération du maillage en plusieurs sous-problèmes, qui peuvent être résolus par plusieurs processeurs. Pour cela, il est nécessaire d’effectuer au préalable un partitionnement du domaine, afin d’affecter par la suite, chacune de ses parties à un processeur. Concrètement, ce partitionnement est réalisé en décomposant le domaine en plusieurs sous-domaines, par une méthode de type *octree* par exemple [COUGNY & SHEPHARD 99], ou bien en utilisant une approximation du domaine par un maillage grossier qui est décomposé en plusieurs sous-maillages par une technique éventuellement parallèle de partitionnement de graphe telle que celle de [OLIKER & BISWAS 98]. L’union de l’ensemble des sous-domaines, ou de l’ensemble des sous-maillages, donne un recouvrement du domaine complet. On appellera *interfaces*, les zones frontières entre les différents sous-domaines ou sous-maillages.

Les méthodes de maillage en parallèle sont classées par [GEORGE & BOROUCHAKI 98] en trois catégories :

1. les méthodes qui maillent en premier lieu, en parallèle ou en séquentiel, les différentes interfaces, pour ensuite mailler individuellement chaque sous-domaine,

2. les méthodes qui vont tout d'abord mailler en parallèle les sous-domaines, pour ensuite mailler les interfaces, de façon à ce que le maillage global soit conforme si nécessaire,
3. et enfin les méthodes qui simultanément maillent les sous-domaines tout en affinant les interfaces.

Les analyses de performances d'une application de maillage en parallèle montrent que celle-ci dépendent de plusieurs facteurs. Il est notamment nécessaire d'avoir un bon équilibrage de charge des processeurs sur les différents sous-domaines, ce qui nécessite une "bonne" méthode de partitionnement. De plus, un aspect décisif dans le choix d'une méthode de parallélisation, est le maillage des interfaces entre sous-domaines voisins, qui conditionne souvent le nombre de communications entre processeurs, et donc les performances de l'application.

La méthode de maillage adaptatif, développée dans le cadre de cette thèse, est présentée au chapitre 4. Elle s'appuie sur la construction d'un maillage initial structuré, par couche, qui est ensuite adapté, en fusionnant un certain nombre de cellules adjacentes, de façon à optimiser localement la distribution de l'information apportée par les rais sismiques. La propriété de conformité de ce nouveau maillage adapté, non structuré et non conforme, n'a pas à être respectée puisque celle-ci n'est pas nécessaire dans la résolution du problème géophysique (inversion tomographique). Concernant le type de parallélisation employé pour la génération du maillage adaptatif, l'algorithme développé dans cette thèse fait partie des méthodes de maillage parallèle de la troisième catégorie où les sous-domaines et les interfaces sont maillés simultanément.

Chapitre 3

Tracé de rai sismique dans un maillage régulier de la Terre

L'objectif de ce chapitre est de caractériser l'illumination d'une région quelconque du globe, par les rais sismiques dont les paramètres proviennent des bases de données de la sismicité mondiale. Cette illumination s'appuie sur un ensemble de critères géométriques qui dépendent de la distribution 3D des rais sismiques dans cette région.

Pour y parvenir, dans la première partie, un maillage régulier du globe est construit, subdivisant la Terre en cellules hexaédriques. Le calcul de la trajectoire des rais sismiques dans un tel maillage est présenté dans la deuxième partie, et permet pour chacune de ses cellules, de collecter les informations relatives à la géométrie des rais qui la traversent. Ainsi, l'utilisation de règles combinant ces informations géométriques, définies par l'expert sismologue, permet de caractériser le degré d'illumination, ou *score*, de chaque cellule du maillage.

Devant la quantité de rais contenus dans les bases de données de la sismicité mondiale, dont il faut calculer la trajectoire, l'algorithme du tracé de rais dans le maillage a été parallélisé pour éviter des temps de calcul prohibitifs et pour repousser les limitations liées à la mémoire. Cet algorithme, présenté dans la troisième partie de ce chapitre, a été parallélisé selon deux stratégies de distribution de données qui permettent un bon équilibrage de charge.

Enfin, la dernière partie de ce chapitre présente, à partir d'un ensemble d'expérimentations, une analyse des performances obtenues sur différentes plates-formes : une machine parallèle, un cluster de PC, ainsi qu'une grille de calcul métropolitaine et nationale.

L'apport du travail réalisé dans ce chapitre, par rapport aux travaux existants, concerne essentiellement :

- le pilotage du tracé de rais par une analyse de phase sismique (en implémentant un analyseur avec l'outil YACC),
- le calcul d'un score évolué prenant en compte différents paramètres, alors que classiquement seule la densité de rais par cellule est prise en compte,
- la parallélisation du tracé de rais dans un maillage.

3.1 Construction d'un maillage initial régulier hexaédrique

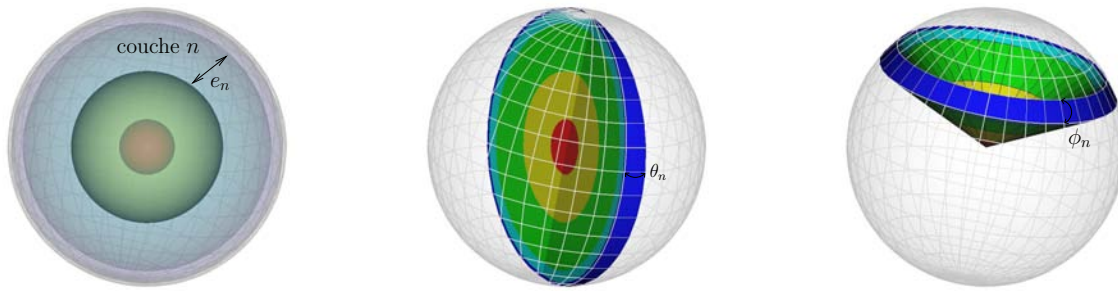
3.1.1 Le maillage et les cellules

Comme exposé précédemment, le choix retenu dans ce travail est de construire un maillage initial par couche dont les cellules peuvent être approximées par des hexaèdres. La Terre étant à symétrie sphérique la construction du maillage du globe s'appuie sur une série de décompositions géométriques simples de la sphère. Dans le cas où la zone géographique d'étude ne concerne qu'une partie du globe, le maillage associé est une sous-partie du maillage global.

La construction de ce maillage global est réalisée en trois étapes :

1. tout d'abord la sphère est décomposée, de la surface jusqu'au centre, en N couches concentriques, ou enveloppes, d'épaisseur variable e_n , avec $n \in [1, N]$ désignant l'indice de chaque couche (cf. figure 3.1(a)) ;
2. ensuite chaque couche n est régulièrement découpée, par pas de θ_n degrés selon les longitudes, par des grands cercles passant par les deux pôles, donc le long des méridiens (cf. figure 3.1(b), où le découpage est présenté pour la couche $n = 1$ pour des raisons de lisibilité) ;
3. enfin, pour chaque couche, un dernier découpage, en secteurs angulaires réguliers de ϕ_n degrés, est effectué selon les latitudes (cf. figure 3.1(c) pour $n = 1$). Tous ces secteurs angulaires ont leur origine au centre de la Terre.

Les volumes élémentaires ainsi créés constituent les cellules du maillage et peuvent être approximés par des hexaèdres (figure 3.2). Par construction, les faces haut/bas d'une cellule, qui correspondent aux faces supérieures et inférieures dans une couche donnée, sont en fait des carreaux de sphère conformément au découpage de la sphère, en couches concentriques, à l'étape 1. Les faces Est/Ouest sont des plans délimités par deux grands cercles à l'étape 2, et les faces Nord/Sud des carreaux de cône, c'est-à-dire une partie de la surface d'un cône défini par le processus décrit ci-dessus à l'étape 3. Notons qu'il existe une singularité aux

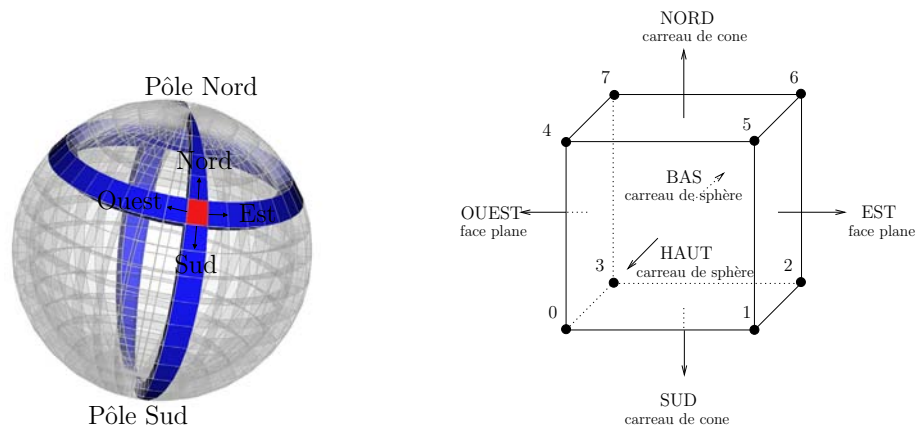


(a) Décomposition de la sphère en N enveloppes concentriques d'épaisseur variable e_n , pour $n \in [1, N]$.

(b) Décomposition de la couche n par des grands cercles en longitude par pas de θ_n degrés.

(c) Découpage de la couche n , en secteurs angulaires en latitude par pas de ϕ_n degrés.

FIG. 3.1 – Les trois étapes de la décomposition d'une sphère lors de la construction du maillage.



(a) Orientation des faces d'une cellule dans le maillage.

(b) Géométrie des faces d'une cellule.

FIG. 3.2 – Orientation et géométrie des faces des cellules du maillage.

pôles et que pour une couche donnée et par rapport à une cellule située à l'équateur, la taille des cellules diminue au fur et à mesure qu'elles sont proches des pôles.

L'ensemble de ces paramètres, à savoir le nombre de couches N , l'épaisseur de chaque couche e_n ainsi que la taille des secteurs angulaires θ_n et ϕ_n utilisés pour la décomposition de chaque couche n de la sphère, est fixé par l'expert sismologue. Les valeurs de θ_n et ϕ_n augmentent avec la profondeur de façon à ce que les dimensions d'une cellule en latitude et en longitude soient à peu près similaires sur l'ensemble des couches. En effet l'utilisation d'un même angle, de la surface jusqu'au centre de la Terre, conduirait à des cellules très petites dans les couches profondes, qui ne contiendraient de ce fait pas d'informations pertinentes du point de vue géophysique. Par ailleurs la taille minimale d'une cellule est contrainte par la longueur d'onde des ondes sismiques utilisées. Procéder à un découpage en cellule de taille inférieure à cette longueur d'onde n'aurait pas de sens au plan géophysique, et l'information mémorisée dans de telles cellules ne serait pas pertinente. De plus, quelque soit le nombre de couches utilisées, les limites de certaines couches coïncident avec les interfaces géologiques majeures (manteau-noyau externe, noyau externe-noyau interne, ...).

3.1.2 Description du maillage

La description complète du maillage se fait par l'intermédiaire d'un fichier XML, dans lequel est défini l'ensemble des paramètres cités précédemment.

En préambule se trouvent les informations suivantes, exprimées en degrés :

- la description de la zone d'étude, définie par les latitudes minimale et maximale (`lat-min` et `lat-max`) et par les longitudes minimale et maximale (`lon-min` et `lon-max`);
- la définition de la cellule unité, qui est la plus petite cellule pouvant exister dans le maillage, en précisant la taille des angles en longitude θ (`lon-unit-size`) et en latitude ϕ (`lat-unit-size`) qui définissent respectivement la taille de la cellule unité dans les directions Est/Ouest et Nord/Sud. La cellule unité sert d'élément de base pour la définition des cellules situées dans les N couches du maillage.

Chaque couche (`layer`) est ensuite décrite par ses limites en profondeur, en ayant respectivement `zstart` et `zend` pour les profondeurs minimale et maximale. La taille d'une cellule, dans une couche donnée, selon les directions Est/Ouest et Nord/Sud, est exprimée en fonction de la cellule unité, `lat-unit` \times `lat-unit-size` pour la taille en latitude et `lon-unit` \times `lon-unit-size` pour la longitude. La taille de la cellule dans la direction haut/bas correspond à l'épaisseur de la couche considérée.

Deux exemples de description de maillage sont présentés aux figures 3.3 et 3.4. Le premier décrit un **maillage global**, composé de cinq couches dont les profondeurs correspondent aux principales interfaces géologiques de la Terre : la lithosphère jusqu'à 150 km de profondeur, le manteau supérieur entre 150 et 700 km puis le manteau inférieur, le noyau

externe et finalement la noyau interne entre 5100 km et le centre de la Terre à 6371 km. Les cellules de ce maillage ont des tailles qui augmentent en latitude et en longitude en fonction de la profondeur, avec un facteur 2 dans le manteau supérieur, jusqu'à un facteur 20 dans le noyau interne. Le deuxième exemple décrit un **maillage local**, dont les caractéristiques proviennent du modèle tomographique réalisé par [GRANET & TRAMPERT 89] pour la région euro-méditerranéenne, située entre les latitudes 33° et 60°, et les longitudes -12° et 30°. Ce maillage, d'une profondeur maximale de 1500 km, est composé de huit couches, avec des cellules de tailles identiques en latitude et en longitude. La représentation de ces deux maillages est donnée figure 3.5.

```
<?xml version="1.0"?>
<mesh lat-unit-size="1" lon-unit-size="1"
      lat-min="-90"    lat-max="90"
      lon-min="0"     lon-max="360">

  <model>
    <layer name="lithosphère"  zstart="0"    zend="150"  lat-unit="1" lon-unit="1"/>
    <layer name="manteau sup." zstart="150" zend="700"  lat-unit="2" lon-unit="2"/>
    <layer name="manteau inf." zstart="700" zend="2900" lat-unit="2" lon-unit="2"/>
    <layer name="noyau ext."   zstart="2900" zend="5100" lat-unit="4" lon-unit="4"/>
    <layer name="noyau int."   zstart="5100" zend="6371" lat-unit="20" lon-unit="20"/>
  </model>
</mesh>
```

FIG. 3.3 – Description XML du maillage des principales enveloppes de la Terre.

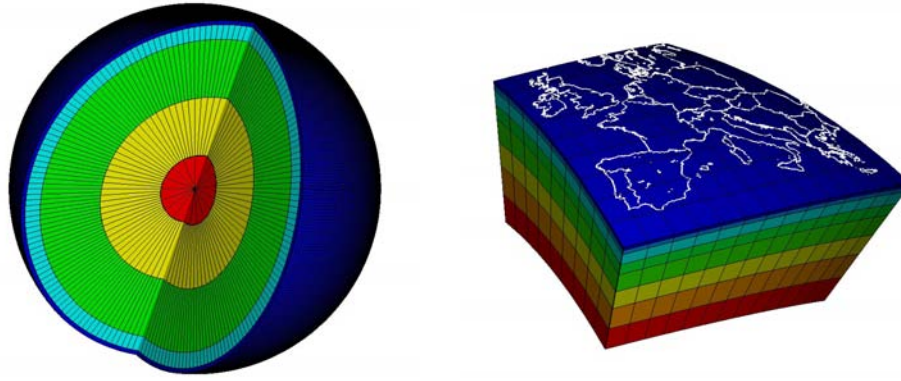
```
<?xml version="1.0"?>
<mesh lat-unit-size="3" lon-unit-size="3"
      lat-min="33"    lat-max="60"
      lon-min="-12"   lon-max="30">

  <model>
    <layer name="layer 0" zstart="0"    zend="40"   lat-unit="1" lon-unit="1" />
    <layer name="layer 1" zstart="40"   zend="100"  lat-unit="1" lon-unit="1" />
    <layer name="layer 2" zstart="100"  zend="250"  lat-unit="1" lon-unit="1" />
    <layer name="layer 3" zstart="250"  zend="400"  lat-unit="1" lon-unit="1" />
    <layer name="layer 4" zstart="400"  zend="650"  lat-unit="1" lon-unit="1" />
    <layer name="layer 5" zstart="650"  zend="900"  lat-unit="1" lon-unit="1" />
    <layer name="layer 6" zstart="900"  zend="1200" lat-unit="1" lon-unit="1" />
    <layer name="layer 7" zstart="1200" zend="1500" lat-unit="1" lon-unit="1" />
  </model>
</mesh>
```

FIG. 3.4 – Description XML du maillage de la région euro-méditerranéenne

3.1.3 Construction du maillage

La construction du maillage repose sur la prise en compte des éléments suivants : la cellule, le méridien et la couche. Les cellules sont assemblées en méridiens, les méridiens en couches, et les couches en maillage global de la Terre.



(a) Maillage global défini par un modèle à cinq couches (figure 3.3).

(b) Maillage de la région euro-méditerranéenne défini par un modèle à huit couches (figure 3.4).

FIG. 3.5 – Maillage global et régional.

Le maillage est représenté sous la forme d'une structure de données décrite par la figure 3.6. Cette description s'appuie sur la structure de données d'une couche (figure 3.6(b)), d'une cellule (figure 3.6(c)), et de l'ensemble des paramètres du maillage défini dans la section précédente (cf. figure 3.6(d)).

La construction du maillage, décrite par l'algorithme 3.1 (page 43), est réalisée en deux étapes principales :

1. une étape *topologique*, qui consiste à construire indépendamment toutes les couches du maillage (figure 3.7). Pour une couche donnée n , les cellules sont instanciées de façon à former des méridiens, et sont mémorisées dans *la table des cellules du maillage* (le tableau `cell_tab_t` dans la figure 3.6(a)). Les liens Nord/Sud entre deux cellules adjacentes, dans un même méridien, sont mis en place dans *la liste des cellules voisines* de chaque cellule (`cell_list_t` dans la figure 3.6(c)). Les méridiens, ainsi formés, sont assemblés en couche, et les cellules adjacentes appartenant à deux méridiens sont liées dans la direction Est/Ouest. Les liens Nord/Sud, entre deux méridiens opposés par rapport à un pôle, sont ensuite mis en place par l'intermédiaire des cellules situées sur ces pôles. Enfin les cellules de deux couches adjacentes, n et $n+1$, sont liées entre elles, par des liens haut/bas. Une cellule peut avoir plusieurs cellules adjacentes, dans la direction haut/bas, si les tailles angulaires, θ et ϕ , des cellules dans les deux couches n et $n+1$ ne sont pas identiques. Dans le cas d'un maillage global, pour permettre des chaînages corrects au niveau des pôles, le nombre de méridiens par couche doit être pair. Notons que chacune des couches ainsi construites est un maillage conforme, tel que défini au chapitre ??, et si pour tout $n \in [1, N]$, les angles θ_n et ϕ_n sont identiques alors le maillage global est lui aussi conforme ;

```

struct mesh_t {
    char *xml_filename;           /* nom du fichier de description XML */
    int ncells;                   /* nombre de cellules */
    int nlayers;                  /* nombre de couches */
    struct layer_t **layer;       /* les couches */
    struct mesh_parameter_t *parameter; /* paramètres du maillage */
    struct point3d_tab_t *allocated_points; /* tableau des sommets alloués */
    struct cell_tab_t *allocated_cells; /* tableau des cellules allouées */
};

```

(a) Structure de données du maillage

```

struct layer_t {
    char *name;                   /* nom de la couche */
    int number;                   /* numéro de la couche */
    int ncells;                   /* nombre de cellule dans la couche */
    int nlat;                     /* nombre de cellules en latitude */
    int nlon;                     /* nombre de cellules en longitude */
    int npoints;                 /* nombre de points alloués */
    struct cell_t *cell;          /* cellule initiale */
    int lat_unit;                /* nb of plat (ie cell min) */
    int lon_unit;                /* nb of plon (ie cell min) */
    float zstart;                /* profondeur du début de la couche */
    float zend;                  /* profondeur de la fin de la couche */
};

```

(b) Structure de données d'une couche.

```

struct cell_t {
    struct coron_z3_t id;         /* coordonnées de la cellule */
    unsigned int point[8];       /* indices des 8 sommets */
    struct cell_list_t *neighbour_list[6]; /* liste des cellules voisines par face */
    struct cell_info_t *cell_info; /* information sur les rais */
    float score;                 /* score ie. illumination */
};

```

(c) Structure de données d'une cellule.

```

struct mesh_parameter_t {
    double lat_min;              /* latitude minimale du maillage */
    double lat_max;              /* latitude maximale */
    double lon_min;              /* longitude minimale du maillage */
    double lon_max;              /* longitude maximale */
    double lat_unit_size;        /* taille cellule unité en latitude */
    double lon_unit_size;        /* taille cellule unité en longitude */
};

```

(d) Structure de données des paramètres du maillage.

FIG. 3.6 – Les structures de données constituant le maillage.

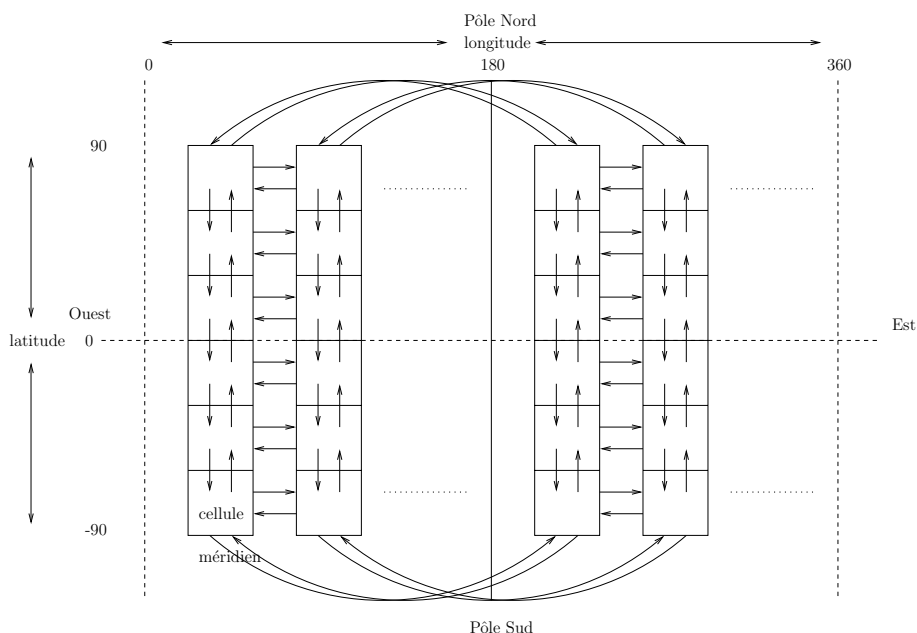


FIG. 3.7 – Étape *topologique* dans la construction de la couche n du maillage. Les relations de voisinage, Nord/Sud et Est/Ouest, entre les cellules sont mises en place.

2. une étape *métrique*, où l'on affecte, à chaque sommet des cellules, leurs coordonnées géographiques : latitude, longitude et profondeur. Pour cela, le maillage est parcouru, couche par couche, cellule après cellule, en utilisant les relations de voisinage (Nord, Sud, Est, Ouest). Les coordonnées des sommets de la cellule courante sont déterminées en fonction de ceux déjà présents dans les cellules voisines. Si le sommet courant est commun avec une cellule déjà traitée, l'instanciation est immédiate en utilisant les relations de voisinages. S'il s'agit d'un nouveau sommet, les coordonnées sont calculées en fonction de la position de la cellule. L'ensemble des sommets est mémorisé dans *la table des sommets du maillage* (le tableau `point3d_tab_t` dans la figure 3.6(a)).

3.2 Le tracé de rais sismiques dans le maillage initial

3.2.1 Analyse de la signature du rai

Lorsqu'un séisme se produit, des ondes sismiques se propagent dans la Terre dans toutes les directions, depuis le foyer du séisme vers les différentes stations sismologiques. Le temps de propagation de chacune de ces ondes dépend du chemin parcouru. Pour un séisme donné, une station peut capter plusieurs ondes, correspondant aux différents

```

Input : le fichier XML de description du maillage
Output : le maillage complet
Data : GlobalMesh est un booléen qui indique s'il s'agit ou non d'un maillage global.
foreach couche z do
  // re-initialisation du premier méridien FirstM
  FirstM ← non défini
  PreviousM ← non défini

  lon-step ← lon-unit × lon-unit-size
  lat-step ← lat-unit × lat-unit-size
  for longitude = lon-min à lon-max step lon-step do
    // re-initialisation de la cellule précédente PreviousC
    PreviousC ← non défini

    // construction d'un méridien
    for latitude = lat-min à lat-max step lat-step do
      // créer la cellule courante
      C ← CreateCell()
      // la lier avec la cellule précédente par un lien Nord/Sud
      if PreviousC est défini then
        | LinkCells(PreviousC, C, Nord/Sud)
      endif
      PreviousC ← C
    endfor

    // Récupération du méridien M qui vient d'être créé
    M = GetMeridianFromCell (C)

    // lier le méridien courant avec le méridien précédent par un lien Est/Ouest
    if FirstM est non défini then
      // le méridien précédent n'existe pas encore
      | FirstM ← M
    endif
    else
      | LinkMeridians(PreviousM, M, Est/Ouest)
    endif
    PreviousM ← M
  endfor

  // maillage global
  if GlobalMesh est vrai then
    // lier le dernier méridien au premier par un lien Est/Ouest
    | LinkMeridians(M, FirstM, Est/Ouest)
  endif

  // liens au niveau des pôles Nord et Sud
  foreach pôle p ∈ { Nord, Sud } do
    // lier entre elles les cellules extrémales des méridiens par un lien Nord/Sud
    | LinkPole(pôle)
  endfch
endfch

```

Algorithme 3.1 : Création du maillage régulier.

chemins empruntés par celles-ci à l'intérieur de la Terre (cf. figure 3.8, où la distance épacentrale entre la source et le récepteur, par rapport au centre O de la Terre définie par l'angle \widehat{SOR} , est de 130 degrés). On modélise chacun de ces chemins par un *rai sismique*. L'analyse des sismogrammes permet à l'expert de déduire les caractéristiques de chaque rai : le mode de propagation de l'onde qui peut passer d'un mode en compression (onde P) à un mode en cisaillement (onde S) et réciproquement, et les différentes réflexions ou transmissions au niveau des interfaces géologiques. Ces caractéristiques définissent la *signature* ou *phase sismique* du rai. Cette signature est un code alphanumérique qui indique chaque changement survenant aux différentes interfaces telles que :

- la surface (à 0 km de profondeur),
- le moho, *ie.* l'interface croûte–manteau (en moyenne à 30 km de profondeur),
- l'interface manteau–noyau externe (à 2889 km de profondeur environ),
- l'interface noyau externe–noyau interne (à 5153 km de profondeur environ),
- une interface située à une profondeur quelconque, précisée dans la signature (par exemple un fond océanique, ou une discontinuité connue telle que celle entre le manteau supérieur et le manteau inférieur à environ 700 km de profondeur).

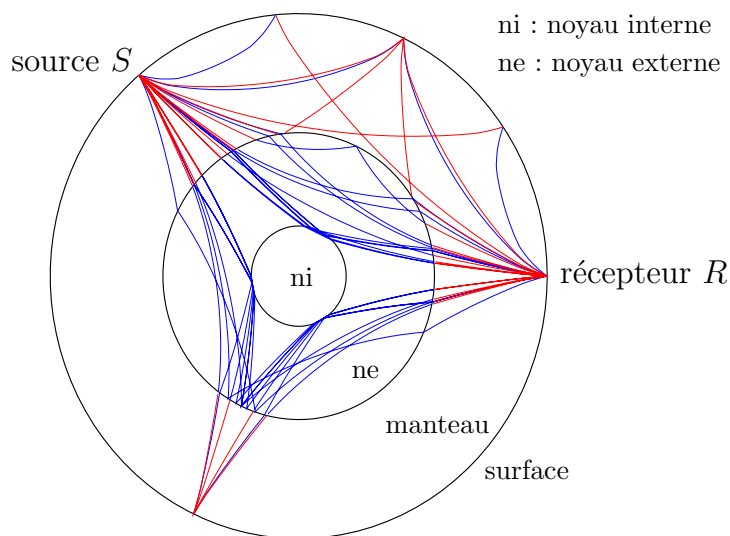


FIG. 3.8 – Exemple des différents chemins empruntés par les ondes sismiques, correspondant à un séisme, entre la source (S) et la station sismologique (R). Les trajets en bleu (resp. rouge) correspondent à une propagation en compression (resp. cisaillement).

Différents exemples de rais sismiques sont présentés figure 3.9. Ils illustrent le mode de fonctionnement des phases sismiques. Nous utilisons les tracés bleus pour désigner une propagation en onde de compression (P), et rouges pour une propagation en onde de cisaillement (S).

1. Le premier exemple, figure 3.9(a), concerne une onde dont la signature est P, indiquant qu'elle se propage directement de la source jusqu'au récepteur en mode de compression. Aucune modification ne vient perturber sa propagation et le code associé à son trajet est donc P. Une onde ne se propageant qu'en mode cisaillement aurait une signature S et se propagerait de façon analogue.
2. Dans l'exemple suivant, figure 3.9(b), il s'agit d'une onde PcP qui se propage en mode de compression (P), jusqu'à l'interface manteau/noyau où elle se *réfléchit* vers la surface (signifié par la présence de c dans la signature), en se propageant en compression (P) jusqu'à la surface.
3. L'onde ayant pour signature PKP, figure 3.9(c), se propage en mode de compression (P) jusqu'à l'interface manteau/noyau, où elle se *transmet* et poursuit son chemin en compression dans le noyau externe (K) jusqu'à ressortir à la même interface, et terminer son trajet jusqu'au récepteur en mode de compression (P).
4. Pour une onde SKS représentée dans la figure 3.9(d), par analogie avec l'exemple précédent (PKP), le trajet de l'onde est semblable, à la différence près que l'onde se propage en mode de cisaillement dans les deux parties de trajet dans le manteau. Ceci impliquant un double *changement de mode de propagation* (de S vers P pour la première transmission et de P vers S pour la deuxième) à l'interface manteau/noyau.

La description de l'ensemble des signatures possibles, [STORCHAK *et al.* 03], a été formalisée par une grammaire à partir de la nomenclature située en annexe B. L'analyse de ces signatures est effectuée par un analyseur engendré automatiquement à partir de cette grammaire par le générateur YACC. L'objectif de cette analyse de phase sismique est de fournir une structure de données permettant de piloter le moteur de tracé de rais de la source vers le récepteur de façon univoque.

Une représentation synthétique de cette structure de données est présentée figure 3.10. Elle illustre le fonctionnement de l'analyse d'une phase sismique complexe PcPSKS. A partir de la source, le rai se propage en mode de compression (P, trajet en bleu) jusqu'à l'interface manteau-noyau, où il se réfléchit (en raison de la présence de c dans la signature) vers la surface en compression (P). Arrivé à la surface, le rai se réfléchit à nouveau et repart en mode de cisaillement (S, trajet en rouge). Arrivé à l'interface manteau-noyau il se propage en compression dans le noyau externe (K, trajet en bleu) pour ressortir à la même interface et terminer son trajet jusqu'au récepteur en mode cisaillement (S).

L'ensemble de ces informations est mémorisé dans la structure de données comme indiqué figure 3.10. La première colonne indique s'il s'agit d'une onde réfléchie ou transmise. La deuxième colonne **sens** donne le sens de propagation du rai, vers la surface (UP) ou vers le centre de la Terre (DOWN). La troisième colonne **depth** précise la profondeur à laquelle a lieu le changement. La dernière colonne **phase** indique le mode de propagation (compression ou cisaillement).

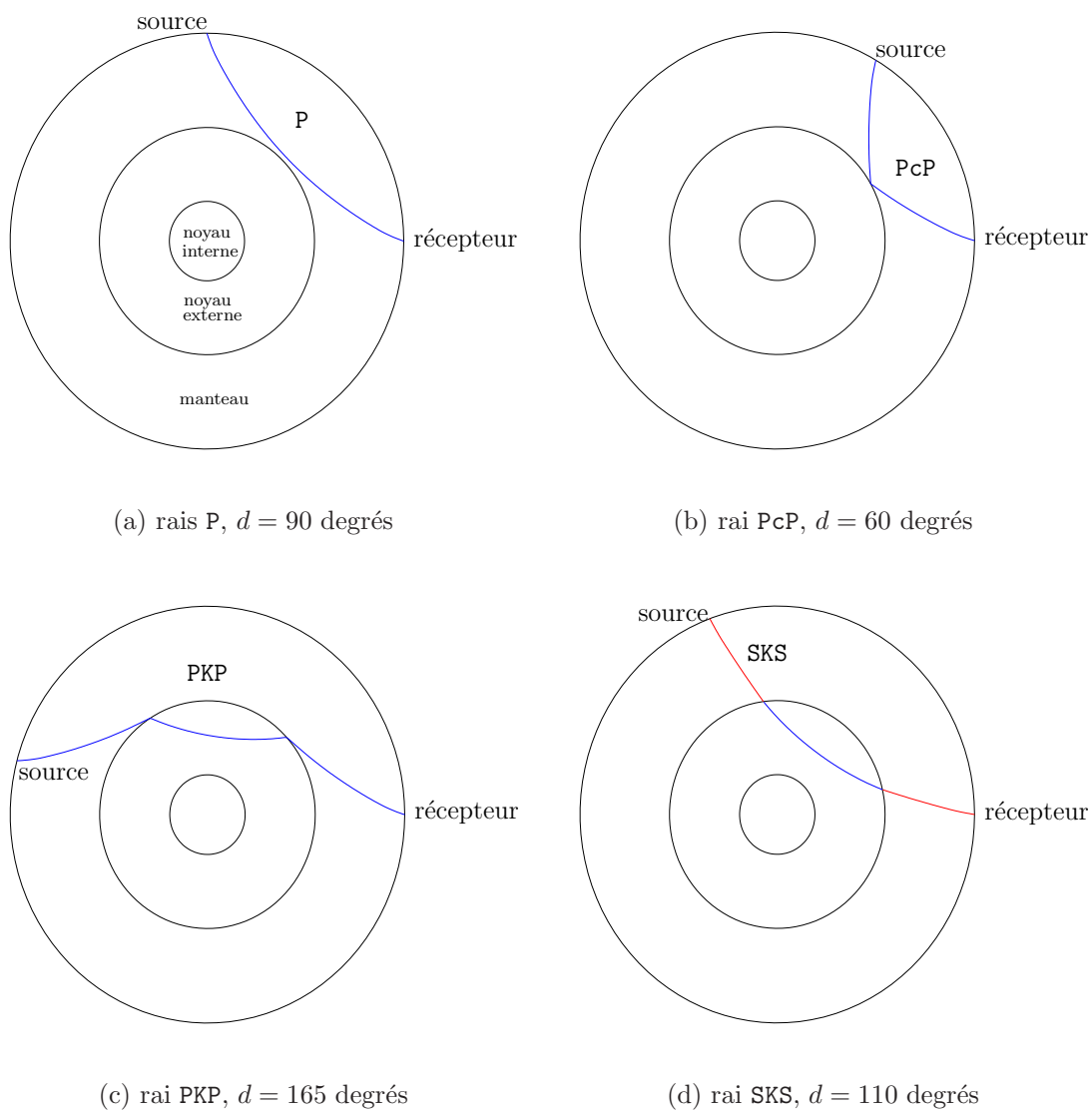
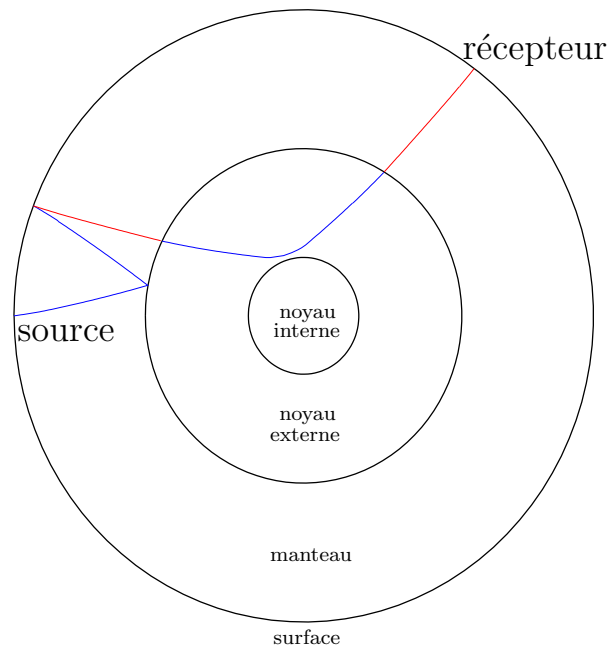


FIG. 3.9 – Exemples de signature de rai. La distance épacentrale d est définie par l'angle \widehat{SOR} , où S est la source du séisme, O le centre de la Terre, et R le récepteur.



PcPSKS	sens	depth	phase
START	DOWN	0.00	P
REFLECTION	UP	2889.00	P
REFLECTION	DOWN	0.00	S
TRANSMISSION	DOWN	2889.00	P
TRANSMISSION	UP	2889.00	S
STOP	-	0.00	-

FIG. 3.10 – Analyse de la phase PcPSKS.

3.2.2 Tracé de rai en 3D

La construction du rai en 3D est un processus itératif qui consiste à propager le rai pas à pas dans le maillage, de la source du séisme jusqu'à la station sismologique. Le rai est donc modélisé par une structure de données définissant l'ensemble de ses points de discrétisation. A chaque point on associe ses coordonnées géographiques ainsi que la distance au point de discrétisation précédent.

La construction du rai se base sur la loi de Snell-Descartes [DESCARTES 37], en géométrie sphérique dont l'équation est :

$$p = r \cdot \frac{\sin(i)}{v(r)} \quad (3.1)$$

où p est le paramètre de rai, constant en tout point du rai, $v(r)$ est la vitesse de propagation des ondes à la profondeur r , et i l'angle d'incidence du rai à cette même profondeur r (figure 3.11).

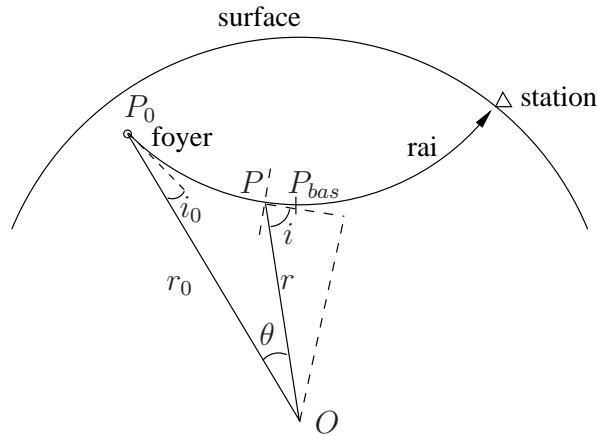


FIG. 3.11 – Chaque point P du rai est défini par sa profondeur r , un angle d'incidence i , ainsi qu'un angle $\theta = (\widehat{P_0OP})$. Le premier point du rai P_0 (la source) est défini par $(r_0, i_0, \theta_0 = 0)$. O est le centre de la Terre et P_{bas} le point bas du rai.

Le calcul de la trajectoire d'un rai nécessite donc de connaître les coordonnées de la source du séisme, du récepteur, sa phase, ainsi qu'un modèle de vitesse de propagation. C'est à partir de ce dernier que l'on détermine grâce à des bases de données (ou par le moyen de tables ou d'abaques) l'angle d'incidence initial du rai, i_0 , à la source du séisme.

Différents modèles de vitesse tels que $PREM^1$ de [DZIEWONSKI & ANDERSON 81] ou $ak135$ de [KENNETT *et al.* 95], ne dépendant que de la profondeur, sont très utilisés par la

¹PREM : Preliminary Reference Earth Model

communauté des sismologues. Ces modèles sont d'excellentes approximations de la structure radiale de la Terre. L'utilisation de tels modèles de vitesse, ne dépendant que de la profondeur, implique que la propagation du rai se fait dans un plan défini par la source, le récepteur, et le centre de la Terre. Un tel plan définit un grand cercle. Le calcul de la trajectoire du rai peut donc être effectué en 2D dans un plan.

Les coordonnées de chaque point du rai, calculées dans ce plan, doivent être ensuite replacées dans l'espace 3D. Pour cela, trois rotations successives sont appliquées. Les deux premières permettent de positionner le plan dans lequel les calculs du rai ont été effectués, sur le grand cercle contenant source et récepteur. La troisième rotation permet de positionner les points source et récepteur, définis sur le plan, sur la source et le récepteur effectifs du séisme sur la sphère.

La construction du rai en 2D est effectuée par un processus itératif qui propage le rai pas à pas en procédant soit par segments linéaires élémentaires (le pas d'échantillonnage du rai), soit par décalages angulaires élémentaires en fonction de l'angle d'incidence courant du rai. En effet, lorsque l'angle d'incidence du rai est de 90 degrés, le point "bas" du rai est atteint et sa trajectoire s'infléchit (passant d'une propagation DOWN à UP, cf. figure 3.11). Aussi, lorsque l'angle d'incidence courant est proche de 90 degrés, l'utilisation de segments linéaires produirait un tracé erroné. C'est pourquoi, lorsque l'on atteint un angle d'incidence compris entre $90-\alpha$ et $90+\alpha$, où α est une constante donnée de l'ordre de 5 centièmes de degré, l'algorithme passe d'un mode de construction du rai par segments linéaires à un mode utilisant des décalages angulaires.

L'analyse de la signature d'un rai, décrite dans la section précédente, intervient à chaque étape de sa propagation. Lorsqu'un segment atteint une interface préalablement référencée lors de l'analyse de phase, les modifications associées (changement du mode de propagation de l'onde en compression ou en cisaillement, réflexion ou transmission) sont appliquées sur le segment en cours de construction. Au fur et à mesure que le rai est calculé pas à pas, son temps de propagation entre foyer et station est calculé, en utilisant le modèle de vitesse de référence.

A l'issue du tracé d'un rai, il arrive que le point final ne coïncide pas exactement avec les coordonnées de la station. Une imprécision sur l'angle d'incidence initial du rai à la source, l'utilisation d'un modèle de vitesse différent de celui qui a permis la détermination de cet angle initial, ou un pas d'échantillonnage du rai mal adapté, peuvent provoquer une telle situation. C'est pourquoi, lorsque la distance entre ces deux points (celui correspondant à la station et le dernier point du rai calculé) est supérieure à un certain seuil de précision, on procède à un calcul itératif du rai, jusqu'à obtenir une précision suffisante.

Ce calcul itératif se décompose en deux étapes. La première consiste à faire varier de façon dichotomique l'angle d'incidence initial i_0 jusqu'à obtenir une précision suffisante. Dans certains cas pathologiques (utilisation d'un modèle de vitesse non compatible, information sur le rai peu fiable), cette précision ne peut être atteinte. C'est pourquoi l'utilisateur fixe

le nombre maximum d'itérations. Dans une deuxième étape, lorsque la précision n'est pas atteinte dans l'étape précédente, on procède à une modification du pas d'échantillonnage du rai, et on réitère la première étape décrite ci-dessus. En pratique, le seuil de précision utilisé dans ce travail, a été fixé à 10 mètres pour la distance entre le point de sortie du rai et le point visé.

Des expérimentations ont été réalisées sur 1 171 197 rais P, pP et PP provenant d'un jeu de données extraits des bases de données de l'ISC par [ENGD AHL *et al.* 98]. Ce jeu de données est couramment utilisé en sismologie, et nous nous en servons pour faire une tomographie globale au chapitre 5. Ces expérimentations montrent que tous les rais sont tracés avec une précision inférieure au seuil. La figure 3.12 présente un histogramme de la distribution du nombre d'itérations réalisé sur ce jeu de données. Le pas d'échantillonnage initial du rai a été fixé à 4 km et le nombre maximum d'itérations pour la recherche dichotomique de la première phase fixé à 30. Le résultat montre tous les rais sont tracés avec la précision souhaitée, et nécessitent en moyenne un total de 12 itérations.

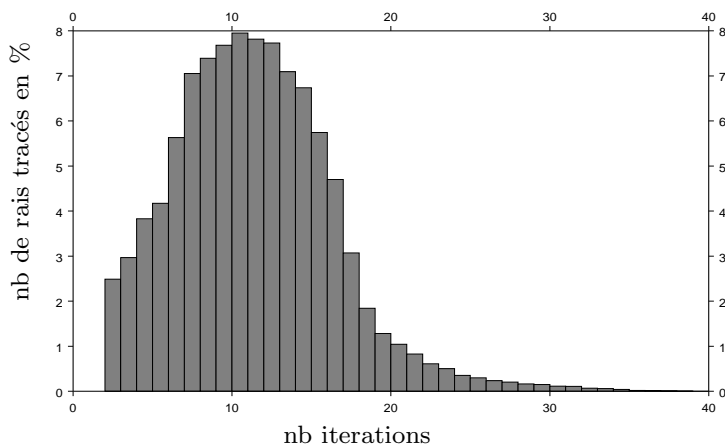


FIG. 3.12 – Distribution du nombre d'itérations réalisées pour la construction des rais P, pP et PP provenant du jeu de données de [ENGD AHL *et al.* 98].

3.2.3 Intégration du rai dans le maillage

Dès que le rai est calculé avec la précision nécessaire, il est "plongé" dans le maillage, et chaque cellule qu'il traverse mémorise les informations qui le caractérisent localement (cf. 3.13), à savoir :

- les points d'entrée et de sortie du rai dans la cellule. Ces deux points sont calculés en déterminant l'intersection du segment de rai avec les faces concernées de la cellule. Ce calcul prend en compte la géométrie des faces concernées : plan pour les faces

Est/Ouest, carreaux de sphères pour les faces haut/bas, et surface de cônes pour les faces Nord/Sud.

- la longueur du rai dans la cellule propagé en mode de compression (P) et en mode de cisaillement (S). Notons que dans la plupart des cellules le rai se propage dans un mode unique (P ou S), et que l’une des deux longueurs est nulle. Les deux longueurs sont non nulles pour les cellules où le basculement (de P vers S ou réciproquement) s’opère.

De plus, chaque cellule intègre les informations apportées par l’ensemble des rais qui la traversent (cf. 3.14) :

- le nombre de rais dans la cellule,
- le nombre d’impacts de rai sur chacune des six faces de la cellule,
- une information sur l’occupation volumique des rais à l’intérieur de la cellule. Pour cela la cellule est subdivisée régulièrement en blocs hexaédriques appelés *mini-blocs*, et la liste des mini-blocs traversés par un ou plusieurs rais est mémorisée.

```

struct cell_info_item_t {
    long int      rayid;      /* identifiant du rai          */
    double        P_length;  /* longueur du rai en propagé en P */
    double        S_length;  /* longueur du rai en propagé en S */
    struct coord_geo_t *in,   /* point d'entrée du rai dans la cellule */
                  *out;     /* point de sortie du rai dans la cellule */
};

```

FIG. 3.13 – Structure de données associée à l’information apportée par un rai dans une cellule.

```

struct cell_info_t {
    int           nitems;     /* nombre de rais dans la cellule */
    struct cell_info_item_t *item; /* information apportée par chacun des nitems rais */
    int           *faces_hit; /* nombre d'impacts sur les six faces de la cellule */
    int           nblocks;   /* nombre de mini-blocs impactés */
    blocknum_t    *block_hit; /* liste des identifiants de mini-blocs impactés */
};

```

FIG. 3.14 – Structure de données associée à l’information apportée par l’ensemble des rais dans une cellule.

3.2.4 Calcul des scores associés à chaque cellule

Les informations décrites ci-dessus sont converties en quatre *scores élémentaires* :

- d , le nombre de rais (en \log_{10}),
- l , la longueur moyenne des rais,
- b , le pourcentage de mini-blocs traversés par les rais,
- et h , le nombre de faces impactées par les rais.

Ces scores sont normalisés par rapport à leur valeur maximale, par exemple la longueur moyenne des rais dans une cellule est rapportée à la longueur d'une grande diagonale de cette même cellule.

Ces scores élémentaires seront combinés en un score unique appelé *score composite* (cf. chapitre 4) qui permettra d'évaluer le degré d'illumination des cellules, par les rais sismiques qui les traversent. Plus ce score sera élevé, meilleure sera l'illumination de la cellule par les rais. La détermination de ce score composite requiert l'expertise du sismologue. C'est une fonction $f(d, l, b, h)$ des scores élémentaires, composée d'opérateurs arithmétiques usuels, appliquée sur chaque cellule. En pratique cette fonction est une routine en C que l'utilisateur peut modifier à sa guise.

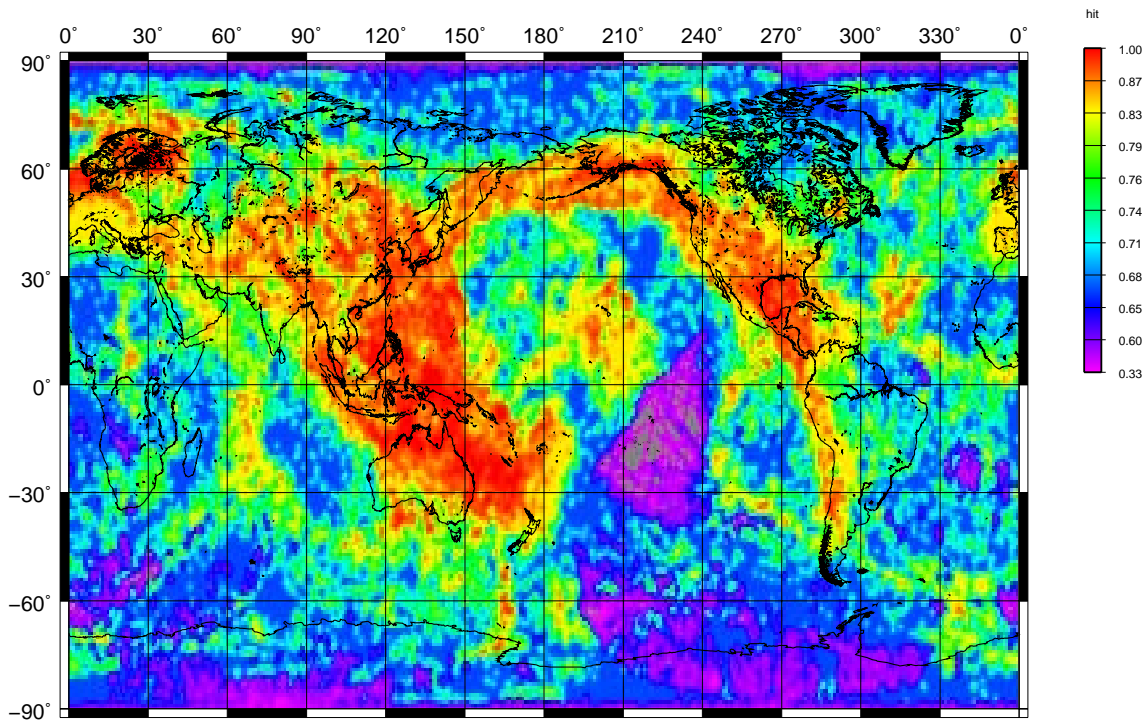
Notons que dans la plupart des travaux de tomographie sismique le score utilisé pour caractériser l'illumination est souvent rudimentaire. Par exemple le nombre de rais par cellules, encore appelé densité de rais lorsqu'il s'agit du nombre de rais en échelle logarithmique, est souvent utilisé ([TKALCIC *et al.* 02], [BIJWAARD *et al.* 98]). Dans ce travail, la combinaison des scores utilisée est simple et consiste en un produit des quatre scores élémentaires. Dans un futur proche il faudra étudier avec les sismologues les effets des différentes combinaisons de ces paramètres, de façon à caractériser au plus juste l'illumination d'une cellule.

La figure 3.15 présente les scores élémentaires des cellules de la couche 7 du maillage (décrit par la figure 3.20), qui s'étend de 760 km à 2740 km de profondeur. Ces scores correspondent à l'exécution réalisée au chapitre 3.4, où l'on a calculé le trajet de 348 401 rais dans un maillage global de la Terre comportant 712 800 cellules. En figure 3.16 est représenté le score composite qui résulte du produit des quatre scores élémentaires. L'échelle de couleur employée est linéaire et va du violet, pour les scores les plus faibles, au rouge pour les scores les plus élevés. Les cellules non impactées par des rais sont représentées en gris sur toutes ces figures. On remarque que les cellules les moins impactées sont situées dans les hautes latitudes (Nord et Sud) ainsi que dans l'océan Pacifique.

3.3 La parallélisation

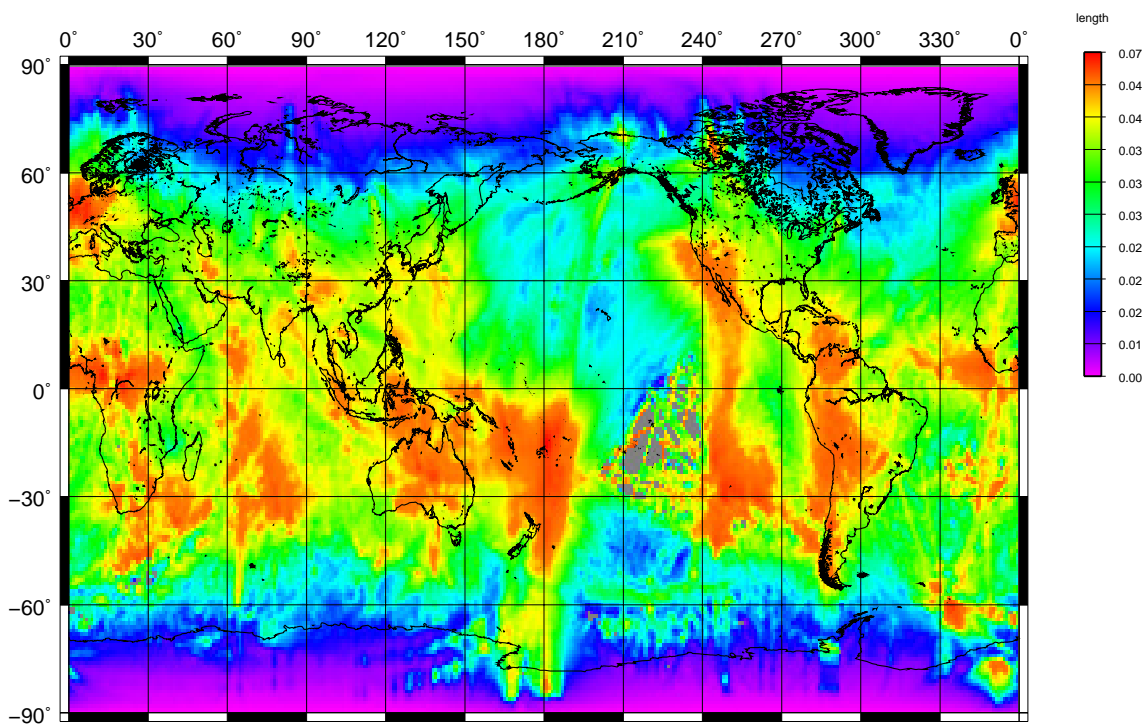
La difficulté pour paralléliser l'application de tracé de rais dans un maillage, réside dans le fait d'avoir à stocker simultanément en mémoire, pour chaque processeur, les rais et le maillage.

Une possibilité serait d'allouer à chaque processeur une partie distincte du maillage basée sur une décomposition géographique (par exemple attribuer les deux hémisphères à deux processeurs). Chaque rai serait alors tracé par le processeur responsable de la région géographique dans lequel il se propage. Une communication inter-processeurs serait néces-



hit score, de 760 à 2740 km

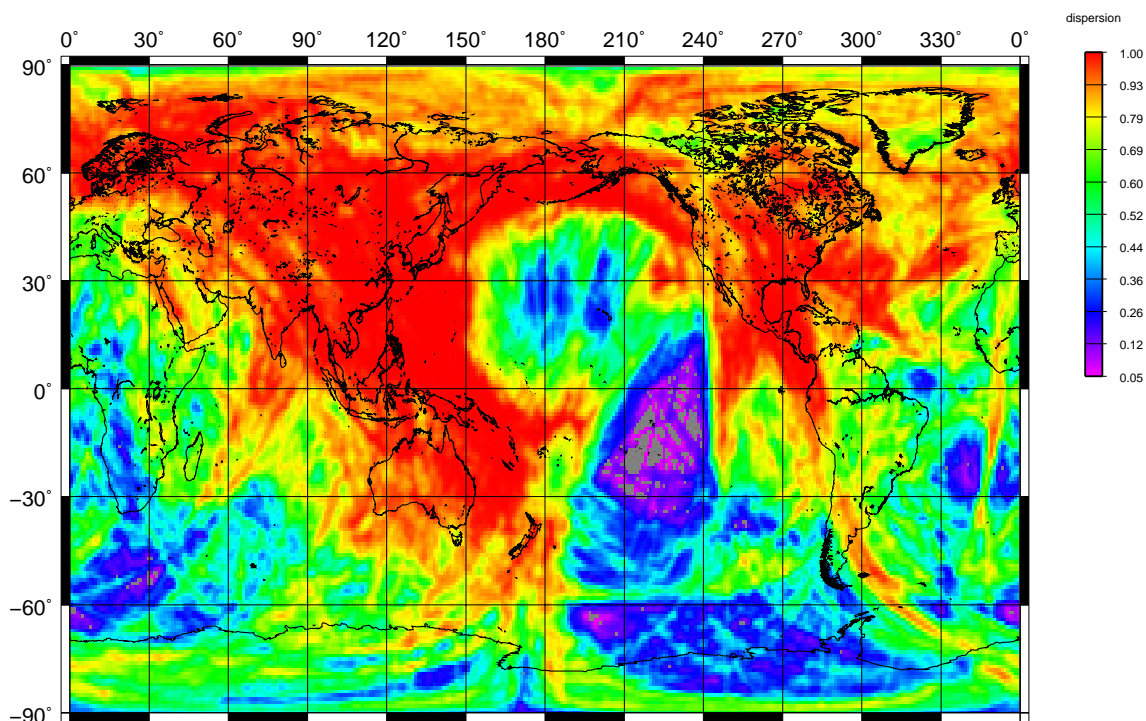
(a) Faces impactées par les rais.



length score, de 760 à 2740 km

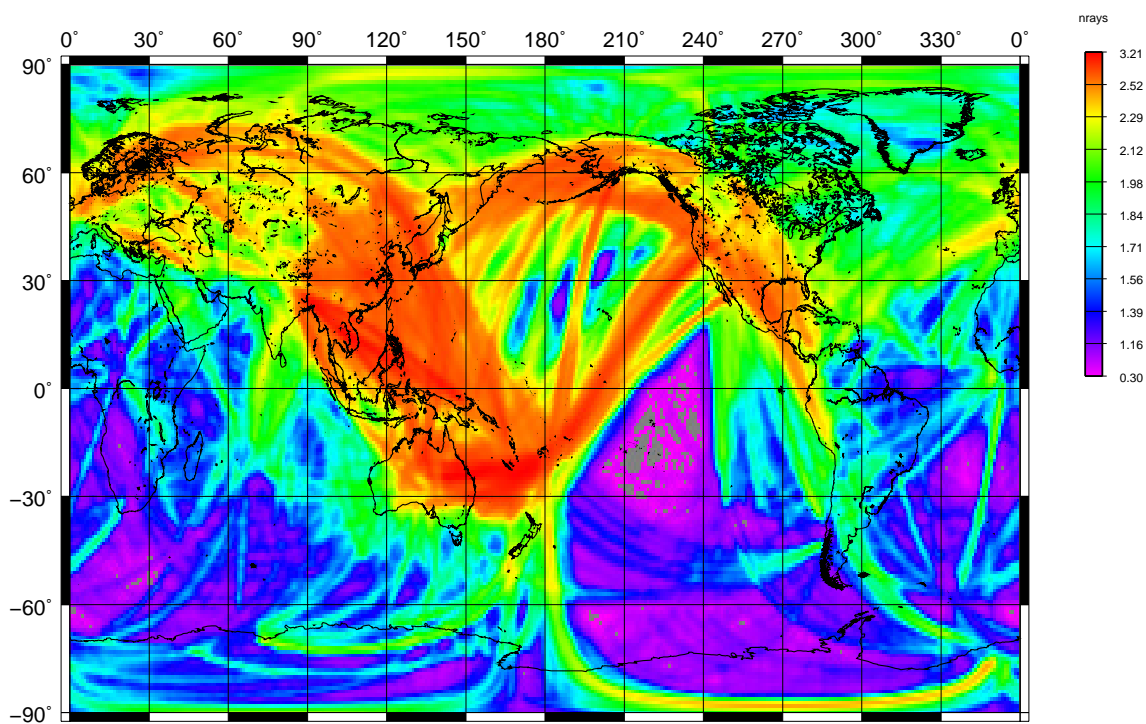
(b) Longueur moyenne des rais.

FIG. 3.15 – (Partie 1/2) – Scores élémentaires permettant de caractériser l'illumination des cellules du maillage par les rais sismiques entre 760 et 2740 km de profondeur.



dispersion score, de 760 à 2740 km

(c) Mini-blocs traversés par les rais.



nrays score, de 760 à 2740 km

(d) Nombre de rais (densité).

FIG. 3.15 – (Partie 2/2) – Scores élémentaires permettant de caractériser l'illumination des cellules du maillage par les rais sismiques entre 760 et 2740 km de profondeur.

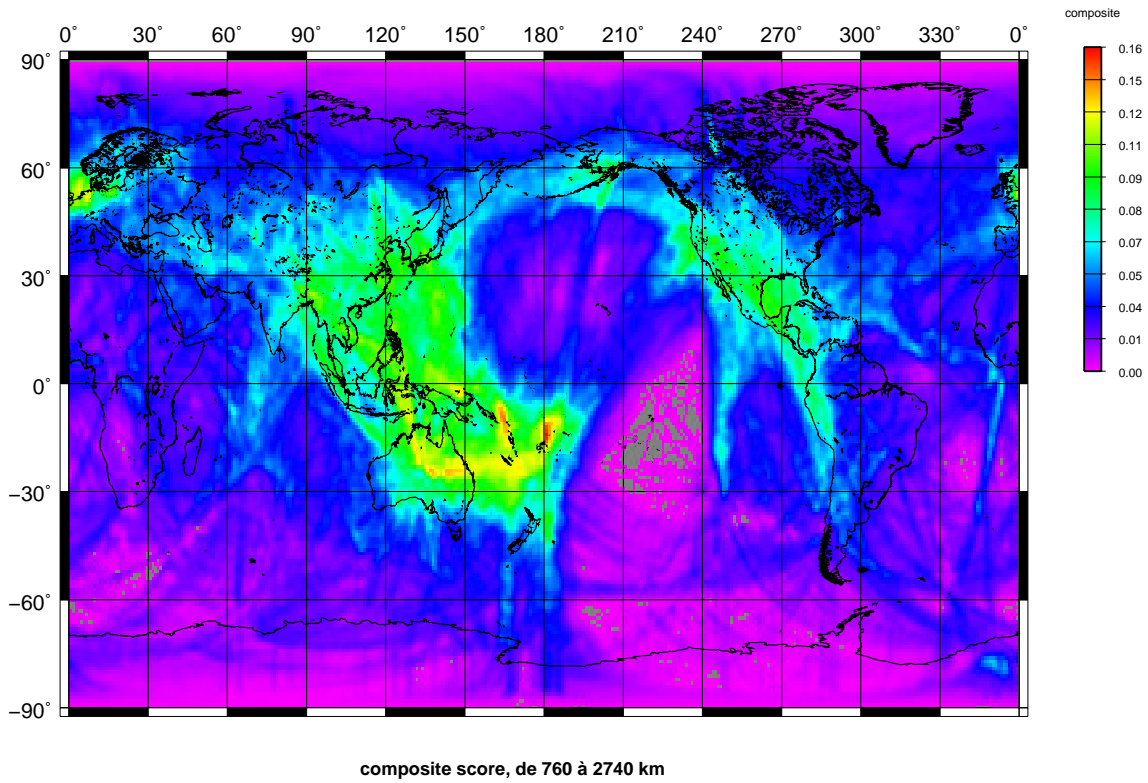


FIG. 3.16 – Score composite résultant du produit des quatre scores élémentaires entre 760 et 2740 km de profondeur.

saire dès que le rai change de région géographique. Cette approche pose deux problèmes. Premièrement, le calcul des rais sur les différents processeurs serait largement déséquilibré puisque les capteurs sont irrégulièrement placés à la surface du globe (il y a très peu de capteurs au niveau des océans par exemple) et par conséquent certains processeurs ne verraient que très peu de rais passer dans la région qui leur est allouée. Deuxièmement, l'augmentation du nombre de processeurs ferait croître le nombre de communications, et conduirait sans aucun doute à de piètres performances.

C'est pourquoi la stratégie de parallélisation utilisée dans ce travail est basée sur la réplication du maillage sur chaque processeur (plus exactement sur un maillage *allégé* pour s'affranchir d'un éventuel problème de mémoire) et sur une distribution des rais à tracer sur les différents processeurs. Chaque processeur est responsable du tracé de ses rais dans la copie du maillage global qui lui est propre. Lorsque tous les rais sont tracés, une phase de fusion de l'ensemble des informations est réalisée.

Pour obtenir une parallélisation efficace deux points doivent être pris en compte :

- l'utilisation de la mémoire sur chaque processeur,
- la distribution des rais pour garantir un bon équilibrage de charge.

3.3.1 Utilisation de la mémoire : maillage allégé

La parallélisation retenue, consistant à répliquer le maillage sur chaque processeur, pose le problème de la taille mémoire nécessaire. Par exemple, le maillage avec lequel nous avons mené une série d'expérimentations, dans la section 3.4, possède 11 couches, et comporte 712 800 cellules de taille $1^\circ \times 1^\circ$. Il occupe, vide de toutes informations sur les rais, environ 180 Mo de mémoire en stockant les cellules, les sommets ainsi que la structure de données permettant de gérer les liens entre les cellules.

Pour contourner ce problème d'occupation mémoire par le maillage, un maillage "allégé" est utilisé. En effet, certaines informations contenues dans le maillage ne sont pas indispensables pour le tracé du rai, comme les liens entre cellules et les coordonnées de leurs sommets. Ainsi, bien que la localisation d'une cellule soit nécessaire lorsqu'il faut lui ajouter les informations associées à un rai qui la traverse, celle-ci peut être aisément réalisée à partir des coordonnées des points de discrétisation du rai, en déterminant la couche contenant la cellule, à partir de la profondeur du point de discrétisation et en utilisant les valeurs de θ et de ϕ de la couche considérée. Enfin, seules les cellules traversées par au moins un rai sont effectivement allouées en mémoire. La structure de données ainsi décrite constitue ce que nous appelons le *maillage allégé*.

Ce maillage allégé permet d'exécuter notre application sur différentes plates-formes disponibles dans les laboratoires ou les centres de calculs universitaires. Ainsi lorsque l'on trace dans le maillage allégé l'ensemble des 348 401 rais de l'année 1999 utilisé dans les

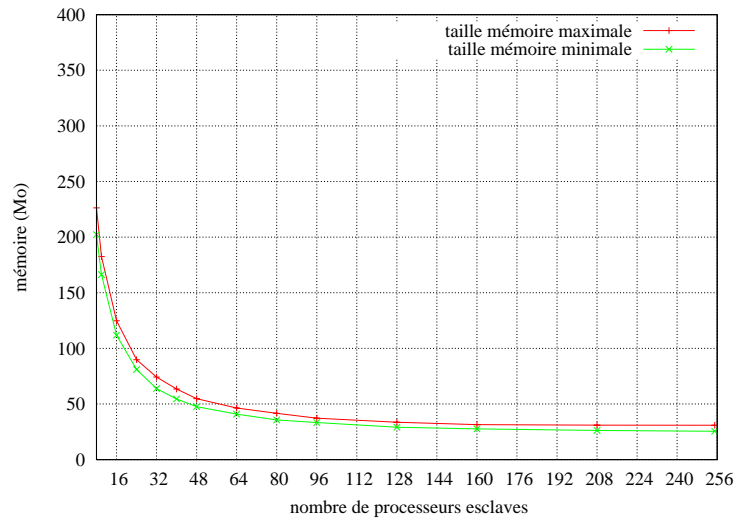
expérimentations, la taille mémoire occupée (code et totalité des données) par l'application est par exemple sur 8 processeurs en moyenne de 320 Mo par processeur, et de 190 Mo par processeur si on les trace sur 16 processeurs. Sans le maillage allégé, cette occupation mémoire serait donc environ de 500 Mo et 350 Mo respectivement, soit un gain de 36% avec 8 processeurs, et 48% avec 16 processeurs. Ainsi l'exécution de ce programme est rendue possible sur des configurations relativement modestes. Par exemple, nous avons réussi à faire le calcul sur un cluster de 4 nœuds bi-processeurs disposant de 1 Go de RAM chacun.

La figure 3.17 donne une idée de l'occupation mémoire, avant et après la phase d'échange, quand le nombre de processeurs augmente (les chiffres sont tirés d'une exécution *maître/esclave* sur une SGI Origin 3800). La différence maximale d'occupation mémoire constatée entre les différents processeurs s'explique principalement par le fait que la distribution des rais à l'échelle globale est intrinsèquement biaisée (*ie.* répartition des récepteurs et séismes), certaines régions étant plus impactées par les rais que d'autres. Il en résulte que même si chaque processeur trace un nombre à peu près équivalent de rais dans sa copie du maillage global, la répartition géographique de ceux-ci a un impact sur l'occupation mémoire. En effet, comme on peut le voir sur la figure 3.9 page 46, la longueur d'un rai peut être extrêmement variable (en fonction de la distance source/récepteur et des différentes réflexion/transmission). Ceci implique que le nombre de cellules traversées par un rai est aussi extrêmement variable. La taille mémoire consommée par un rai dans le maillage dépend bien sûr du nombre de cellules traversées, mais aussi du nombre de cellules traversées vierges de tout rai (puisqu'il y a instanciation des structures de données associées à une cellule au premier impact d'un rai). Ainsi un rai ne traversant que des cellules vierges induirait une consommation mémoire supérieure à celle du même rai traversant des cellules déjà impactées. L'impact d'un rai sur l'occupation mémoire dépend donc non seulement de ses caractéristiques (source/récepteur/signature) mais aussi de la répartition des rais déjà tracés par le processeur concerné dans sa copie du maillage. Notons toutefois qu'à la période où ce test a été effectué, le jeu de rais utilisé était structuré par séismes, ce qui a très certainement induit un déséquilibre supplémentaire en mémoire. Actuellement un pré-traitement par ray-bundles (décrit à la page 116) est effectué sur le jeu de données initial en fusionnant des rais similaires en un rai unique. Ce pré-traitement "casse" complètement la structure du jeu de données initial et induit une forme de distribution aléatoire du jeu de données finalement utilisé dans la chaîne sismologique.

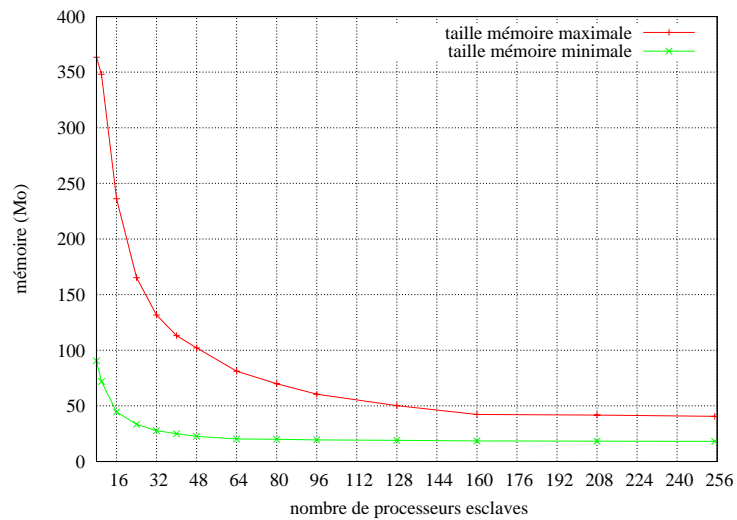
3.3.2 Distribution des rais dans le maillage

Un nombre quelconque de processeurs peut être utilisé lors d'une exécution parallèle, et nous plaçons habituellement un processus par processeur.

Chaque processeur reçoit la description du maillage sous forme XML correspondant à la zone d'étude, et les rais qui lui sont confiés (coordonnées de la source et du récepteur,



(a) Occupation mémoire avant la phase d'échange.



(b) Occupation mémoire après la phase d'échange.

FIG. 3.17 – Occupation mémoire minimale et maximale de l'application avant et après la phase d'échange, dans le cas d'une distribution des rais en mode *maître/esclave*.

signature du rai et angle d'incidence initial). Il commence alors à calculer ses rai segment par segment. Pour chaque segment, le processeur teste si le rai vient d'entrer dans une nouvelle cellule auquel cas une zone mémoire est allouée et les propriétés géométriques et géophysiques du rai y sont stockées. Dans le cas contraire les informations de la cellule traversée sont mises à jour.

Deux modes de distribution des rai ont été implémentés. En effet, la plate-forme visée initialement était une machine parallèle pour laquelle on estime que distribuer en une opération un même nombre de rai est une bonne stratégie. Par la suite, le portage de l'application vers des plates-formes hétérogènes nous a amené à introduire une distribution dynamique pour équilibrer la charge de calcul. Nous distinguons :

- le mode *scatter* : l'ensemble des rai est découpé en p blocs de même taille , où p est le nombre de processeurs, et chaque bloc est affecté à un processeur ;
- le mode *maître/esclave* : l'ensemble des rai est découpé en blocs de taille identique, en fonction du nombre de rai et du nombre de processeurs. Dans ce mode le processeur maître distribue les blocs de rai à la demande aux esclaves. Dès qu'un processeur esclave a terminé le calcul des rai de son bloc courant, il en demande un nouveau au processeur maître.

Ces deux modes de distribution ont été testés et les résultats sont présentés à la section 3.4. Selon le type de plate-forme utilisée, l'équilibrage de charge de ces modes de distribution est analysé.

3.3.3 Description de l’algorithme parallèle

Après la réplication du maillage allégé sur les processeurs, et le calcul des rais dans le maillage, l’information sismique associée à une cellule donnée est potentiellement répartie sur plusieurs processeurs. Nous devons alors fusionner, pour chaque cellule, l’ensemble de ces données distribuées.

Cette étape de fusion est réalisée en affectant à chaque processeur une partie distincte du maillage, appelée *sous-domaine*, issue d’un découpage régulier du maillage. Chaque processeur doit collecter auprès des autres processeurs les informations correspondant aux cellules de son sous-domaine. Une phase de communication *tous à tous* doit être réalisée où chaque processeur envoie les données qu’il a calculées aux différents processeurs concernés², et reçoit les données des autres processeurs concernant son *sous-domaine*. Notons que cette phase est extrêmement coûteuse en communication : pour n processeurs il y a n^2 échanges de sous-domaines nécessaires, et pour des exécutions comme celle présentée section 3.4, environ 1,5 Go de données au total sont échangés.

Chaque processeur ayant reçu toutes les données le concernant, fusionne les informations des sous-domaines et calcule le *score* de chaque cellule, correspondant à son illumination, puis retourne les résultats calculés au processeur maître.

Le schéma général de l’application, pour une implémentation maître/esclave est décrit ci-dessous, et met en évidence les différentes phases de communication entre les processeurs ainsi que les étapes de calcul. Il est présenté, pour une implémentation maître/esclave, dans la figure 3.3.3.

- | | |
|--|---|
| 1. description du maillage | Le processeur maître distribue la description du maillage aux autres processeurs (<i>communication un à tous</i>). |
| 2. distribution des blocs de rais | Chaque processeur demande au processeur maître un bloc de rais à traiter (<i>communication un à un</i>). |
| 3. le tracé de rais | Chaque processeur calcule les trajets des rais et met à jour les cellules traversées. Retour à l’étape (2) jusqu’à épuisement des blocs de rais (<i>calcul</i>). |
| 4. échange de données | Chaque processeur envoie aux autres les données des cellules dont ils ont la charge, et récupère les données des cellules dont il a la charge (<i>communication tous à tous</i>). |

²En raison du découpage régulier du maillage en sous-domaines, chaque processeur peut déterminer la référence du processeur à qui est attribué un sous-domaine donné.

- | | |
|-----------------------------------|--|
| 5. fusion des informations | Les informations concernant une cellule sont fusionnées (<i>calcul</i>). |
| 6. calcul de score | Pour chaque cellule, on calcule un score à partir des informations des rais (<i>calcul</i>). |
| 7. concaténation | Le score de chaque cellule est envoyé au processeur maître (<i>communication tous à un</i>). |

3.4 Résultats expérimentaux sur différentes plates-formes

L'ensemble des expérimentations présentées dans cette partie s'appuie sur un jeu de données, extrait des bases de données de l'ISC³, qui correspond à l'ensemble des événements sismiques de l'année 1999. L'expérience consiste à calculer, dans un maillage global de la Terre de 712 800 cellules réparties sur 11 couches (cf. figure 3.20), les scores des cellules impactées par les trajets de 348 401 rais sismiques. La liste des signatures, ainsi que le nombre rais par signature, sont représentés sur la figure 3.19.

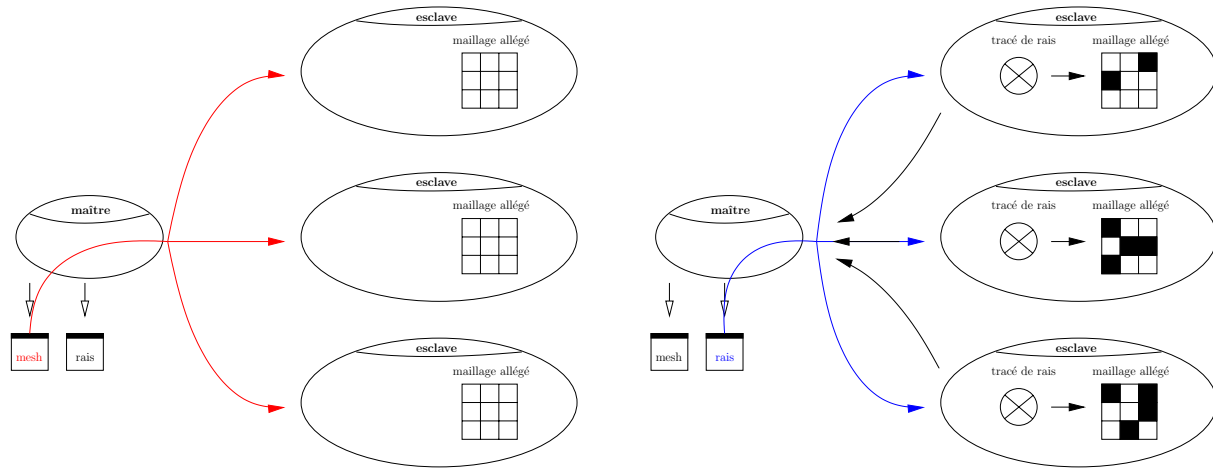
Le tracé de ces rais dans le maillage initial a été réalisé sur plusieurs types de plates-formes disponibles, soit dans nos laboratoires (cluster de stations), soit dans un centre informatique national (machine parallèle). Nous avons également utilisé une grille de calcul constituée à la fois de machines standards des laboratoires et la machine parallèle du centre informatique national reliées par le réseau Renater.

Les différentes caractéristiques techniques de ces plates-formes ont une grande influence sur les performances de l'application. Après une description des choix d'implémentation (section 3.4.1) et une présentation détaillée des différentes plates-formes utilisées (section 3.4.2), nous présentons trois séries de tests qui consistent :

1. à mesurer la scalabilité, ou le passage à l'échelle, de l'application sur une machine parallèle et un cluster ;
2. à comparer les performances de cette application en utilisant 10 processeurs, sur une machine parallèle, un cluster et une grille de calcul métropolitaine (*ie.* à l'échelle d'une ville) ;
3. à comparer les performances de l'application en utilisant 16 processeurs, sur une machine parallèle et sur une grille de calcul nationale.

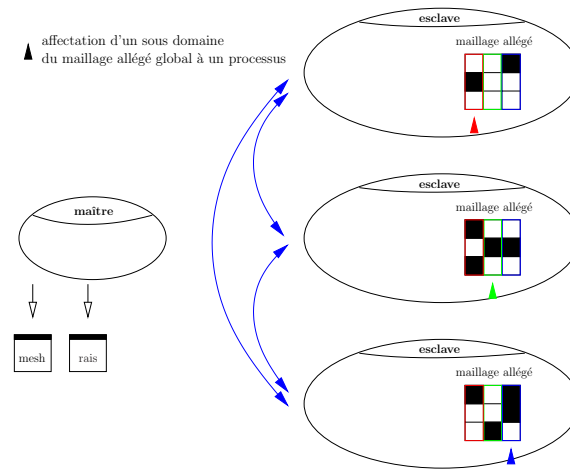
Dans la deuxième et la troisième séries de tests réalisés, deux configurations de grille sont comparées aux mêmes exécutions sur des plates-formes homogènes (machine parallèle, cluster) dont la puissance de calcul est du même ordre de grandeur. De plus, la troisième

³ISC : International Seismic Center.



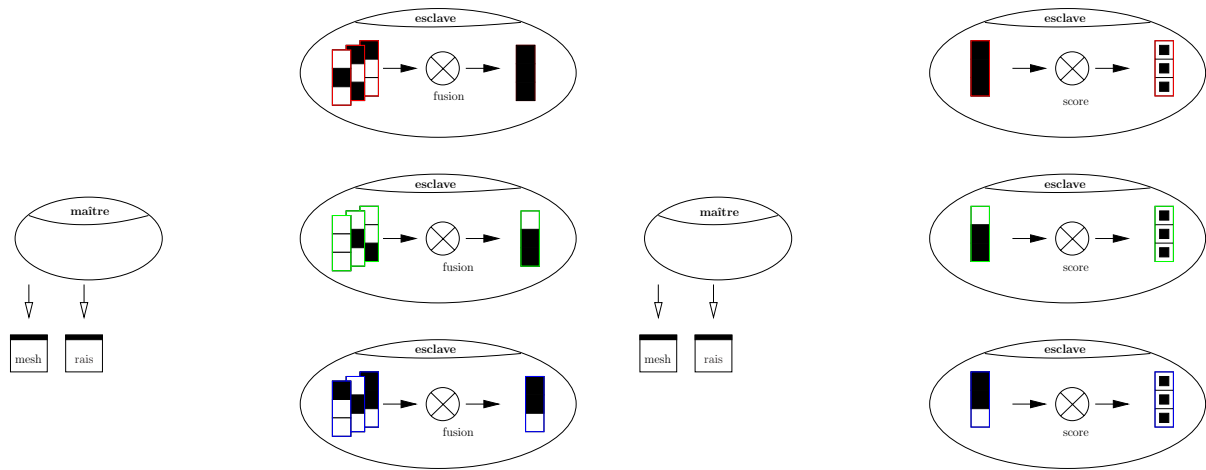
(a) Le processeur maître envoie la description du maillage à tous les esclaves. Création du maillage *allégé* par chaque esclave.

(b) Distribution des blocs de rais à la demande et tracé de rais.



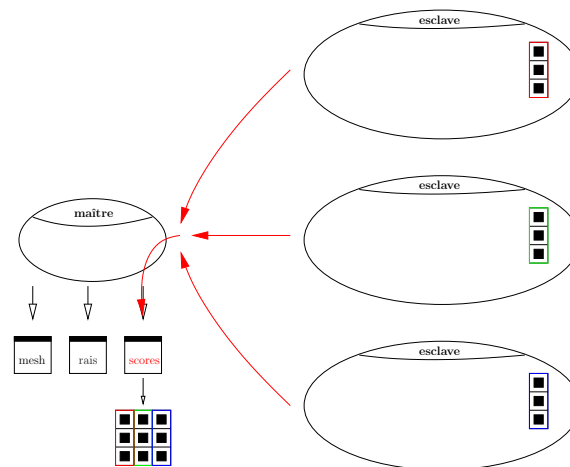
(c) Affectation d'un sous-domaine à chaque processeur et échange de données.

FIG. 3.18 – (Partie 1/2) – Schéma de la parallélisation de l'application pour une distribution de données en mode *maître/esclave*.



(d) Fusion des informations récupérées par chaque esclave.

(e) Calcul de l'illumination des cellules par chaque esclave.



(f) Le processeur maître récupère l'ensemble des résultats.

FIG. 3.18 – (Partie 2/2) – Schéma de la parallélisation de l'application pour une distribution de données en mode *maître/esclave*.

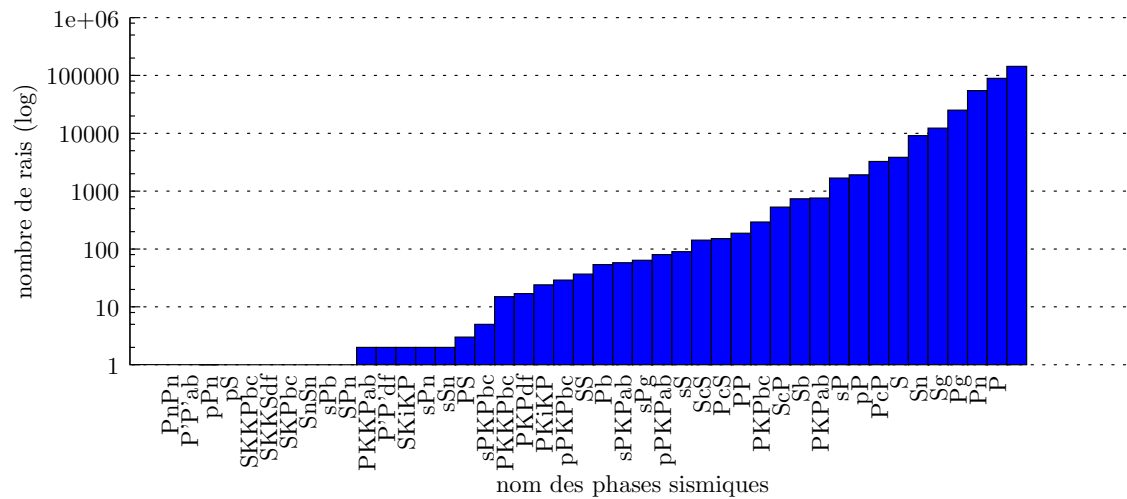


FIG. 3.19 – Distribution par signatures du nombre de rais calculés à partir de 348 401 rais extraits des bases de données de l’ISC.

```
<?xml version="1.0"?>
<mesh
lat-unit-size="1" lon-unit-size="1"
lat-min="-90" lat-max="90"
lon-min="0" lon-max="360">

  <model>
    <layer name="layer0" zstart="0" zend="20" lat-unit="1" lon-unit="1"/>
    <layer name="layer1" zstart="20" zend="35" lat-unit="1" lon-unit="1"/>
    <layer name="layer2" zstart="35" zend="120" lat-unit="1" lon-unit="1"/>
    <layer name="layer3" zstart="120" zend="210" lat-unit="1" lon-unit="1"/>
    <layer name="layer4" zstart="210" zend="410" lat-unit="1" lon-unit="1"/>
    <layer name="layer5" zstart="410" zend="660" lat-unit="1" lon-unit="1"/>
    <layer name="layer6" zstart="660" zend="760" lat-unit="1" lon-unit="1"/>
    <layer name="layer7" zstart="760" zend="2740" lat-unit="1" lon-unit="1"/>
    <layer name="layer8" zstart="2740" zend="2889" lat-unit="1" lon-unit="1"/>
    <layer name="layer9" zstart="2889" zend="5153" lat-unit="1" lon-unit="1"/>
    <layer name="layer10" zstart="5153" zend="6371" lat-unit="1" lon-unit="1"/>
  </model>
</mesh>
```

FIG. 3.20 – Description XML du maillage global utilisé lors des expérimentations (712 800 cellules réparties sur 11 couches).

série de tests, utilisant une même version de l'application, a été réalisée à deux périodes différentes (août 2002 et mai 2003), entre lesquelles, l'infrastructure du réseau Renater a évolué de Renater 2bis à Renater 3, nous permettant ainsi de mesurer l'influence de la qualité du réseau sur ce type d'application.

Notons toutefois que la version de l'application utilisée pour les tests de scalabilité est plus récente (juin 2005) que celle utilisée pour les tests avec 10 et 16 processeurs (août 2002). En effet, un certain nombre de changements significatifs ont été apportés, comme par exemple des optimisations sur le calcul des rais et sur la phase d'échange, ainsi que l'ajout d'un calcul d'intersection "exacte" d'un rai sur une face d'une cellule (en tenant compte de la géométrie de la face, cf. figure 3.2(b)). Les évolutions techniques au niveau de la grille de calcul (machines disponibles, bande passante du réseau) entre l'année 2002 et 2005, ne nous ont pas permis de relancer les tests avec la version optimisée.

Les comportements des exécutions sont examinés au regard des temps pris par l'application pour réaliser certaines étapes, décrites à la section 3.3.3, à savoir :

- le calcul des rais ainsi que la mise à jour du maillage (étapes 2 et 3),
- l'échange de données avec les autres processeurs et la fusion des informations (étapes 4 et 5),
- la totalité des traitements (étapes 1 à 7),
- le nombre total de blocs de rais reçu par processeur, en mode maître/esclave.

A titre indicatif, les tailles des blocs dans les expériences décrites par la suite, sont respectivement de 3484 et 2177 rais pour 10 et 16 processeurs, ce qui correspond à une distribution d'une dizaine de blocs par processeur dans le cas d'une plate-forme homogène comme un cluster.

3.4.1 Implémentation

D'un point de vue logiciel, l'implémentation des programmes développés à l'occasion de cette thèse est faite de façon modulaire et portable. Chaque fonctionnalité (maillage, tracé de rais, *etc.*) est écrite sous forme d'une bibliothèque, ou brique logicielle, dont l'organisation est présentée dans la figure 3.21. L'application de tracé de rais dans un maillage, que nous appelons RAY2MESH, nécessite la bibliothèque de construction du maillage, appelée MESH, ainsi que celle du tracé de rais, RAY-DESCARTES. Cette dernière requiert la bibliothèque d'analyse de signature de rai, RAYCODE, ainsi qu'un accès à un modèle de vitesse, VELOCITY MODEL.

Concernant la portabilité, les programmes ont été implémentés en langage C, et la bibliothèque de communication utilisée pour la parallélisation est MPI [MPI 95]. L'ensemble des programmes tourne sous différents systèmes (Linux, IRIX, SunOS, Mac OS X), et la visualisation en 3D utilise VTK [SCHROEDER *et al.* 97].

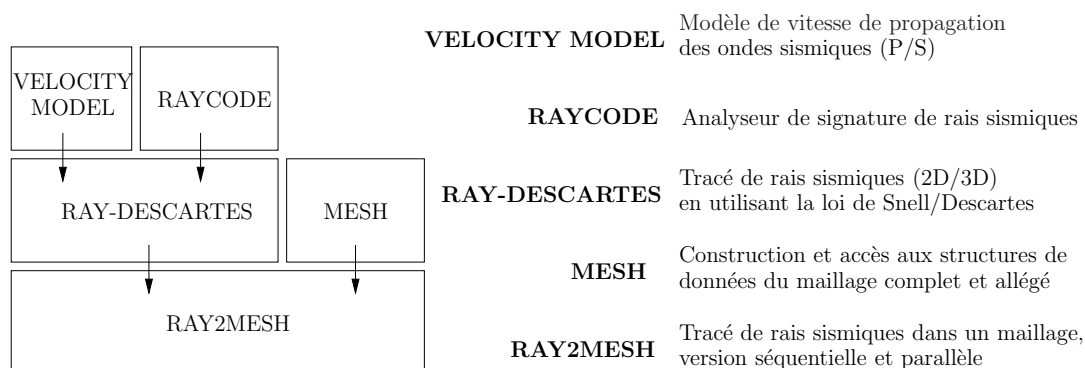


FIG. 3.21 – Organisation des briques logiciels.

3.4.2 Les plates-formes utilisées

Les ressources de calcul utilisées lors de nos expériences, sont prises parmi les processeurs listés dans le tableau 3.1, dont les principales caractéristiques sont énumérées ci-dessous. Les processeurs utilisés étant très hétérogènes, les *puissances relatives* des processeurs indiquées par la colonne ratio, sont alors établies à partir d'une série de benchmarks en utilisant la version séquentielle de l'application. La puissance de ces processeurs est normalisée par rapport à la puissance de la machine la plus lente en l'occurrence celle appelée pellinore.

Type de machine	Nom	Nb CPUs	Type processeur	Ratio
PC de bureau Campus Illkirch	pellinore	2	PIII/800	1
	dinadan	1	PIII/933	1,14
	sekhmet	1	XP1800	1,61
	caseb	1	XP1800	1,61
	lattice	1	XP1800	1,61
	merlin	2	XP2000	2,26
machine parallèle CINES Montpellier	leda	512	R14K/500	1,05
PC de bureau Campus Esplanade	nestea	1	P4/1700	1,4
	darjeeling	1	P4/1700	1,4
cluster EOST	calserv	12	P4/1700 Xeon	1,4
cluster HPC	hpc	60	Itanium 2/1300	1,8

TAB. 3.1 – Caractéristiques des ressources de calcul utilisées dans les expériences.

Les plates-formes utilisées ont les caractéristiques suivantes :

– **Machine parallèle :**

la machine utilisée, leda dans le tableau 3.1, est une SGI Origin 3800 située au CINES⁴ à Montpellier. La puissance de calcul de chaque processeur n'est pas très élevée la puissance relative étant de 1,05, mais le réseau d'interconnexion entre les processeurs, ainsi que les opérations entrées/sorties, sont extrêmement performants. La bibliothèque de communication MPI utilisée est celle fournie par le constructeur. Le système d'exploitation est IRIX 6.5.

– **Cluster EOST :**

ce cluster de PC, calserv dans le tableau 3.1, est composé de 6 nœuds interconnectés par un switch Gigabit Ethernet. Chaque nœud est un bi-Pentium Xeon 1,7 Ghz avec 1 Go de RAM, d'une puissance relative par processeur de 1,4. La bibliothèque de communication utilisée est LAM/MPI [SQUYRES *et al.* 00]. Le système d'exploitation est Linux.

– **Cluster HPC :**

ce cluster, hpc dans le tableau 3.1, est formé de 30 bi-processeurs 64 bits Itanium 2 à 1.3 GHz ayant 8 Go de RAM, la puissance relative par processeur est de 1,8. Les machines sont interconnectées par un réseau Myrinet 2000 (2Gbits/s) [BODEN *et al.* 95]. La bibliothèque de communication utilisée est MPICH [GROPP *et al.* 96] fonctionnant au dessus de Myrinet. Le système d'exploitation est Linux.

– **Grille :**

la grille de calcul utilisée est celle disponible dans le cadre du projet TAG⁵, elle est constituée de machines très diverses, distribuées sur plusieurs sites (Strasbourg, Montpellier, Clermont-Ferrand, Bordeaux⁶) dont les réseaux locaux sont fédérés par le réseau Renater. D'un point de vue logiciel, toutes les machines sont installées avec Globus [FOSTER & KESSELMAN 97] (de la versions 1.1.4 à 2.4) et nous utilisons MPICH-G2 [FOSTER & KARONIS 98] comme bibliothèque de communications. Deux sous-ensembles de cette grille, qui reflètent des exploitations possibles de grilles de calcul ont été sélectionnés :

1. une **grille métropolitaine** utilisant des ressources de calcul au sein d'une même université : en l'occurrence dix PC distribués sur deux campus universitaires géographiquement distants de 6 km (Illkirch, Strasbourg-Esplanade). Le réseau d'interconnexion reliant les différentes machines sur les deux campus supporte un débit de 622 Mb/s, mais les communications sont limitées par les routeurs et cartes réseaux de type fast-ethernet 100 Mb/s. Typiquement, le débit va de 3 Mo/s entre les deux campus jusqu'à 8 Mo/s pour des machines connectées au même routeur ;

⁴CINES : Centre Informatique National de l'Enseignement Supérieur.

⁵TAG : Projet *Transformations et Adaptations pour la Grille*.

⁶Cette grille de test a évolué dans le temps. Les sites de Bordeaux et Clermont-Ferrand n'étaient pas encore accessibles à l'époque où les expérimentations de ce travail ont été effectuées.

2. une **grille nationale** permettant la distribution des calculs à l'échelle nationale en utilisant un réseau à haut débit : huit processeurs du CINES sont "appuyés" par huit autres processeurs sur des campus strasbourgeois. La particularité de cette configuration est qu'elle a bénéficié, à la fin de l'année 2002, d'une amélioration du réseau d'interconnexion entre les sites de Strasbourg et Montpellier, dans le cadre du passage de Renater 2bis à Renater 3.

3.4.3 Test de scalabilité

La scalabilité, ou étude du comportement de l'application lorsque le nombre de processeurs augmente, est analysée dans cette partie au travers de plusieurs tests réalisés avec les versions *scatter* et maître/esclave du tracé de rais. Mais auparavant, une première évaluation des performances est réalisée sur le tracé de rais seul, permettant de déterminer l'efficacité parallèle de la phase de calcul des rais.

Tracé de rais seul

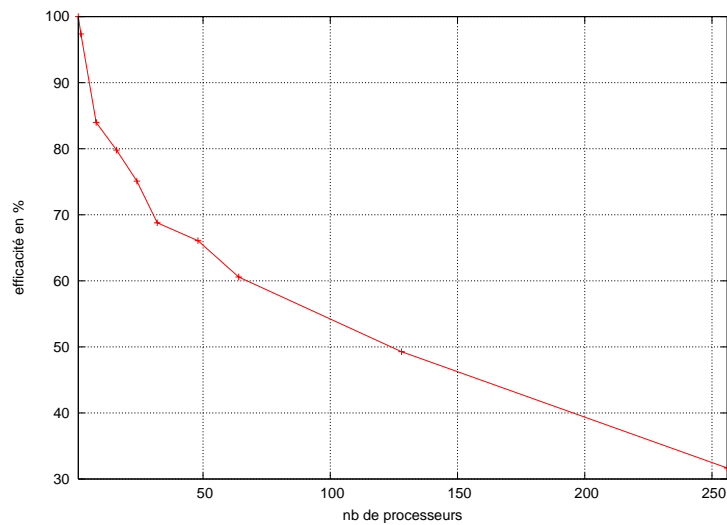
Il s'agit d'évaluer ici la scalabilité du tracé de rais sismiques seul, avec une distribution des données en mode *scatter*, sans tenir compte du temps mis par les processeurs pour mettre à jour les cellules du maillage traversées par les rais. Le tableau 3.2 présente les performances du tracé de rais obtenues en utilisant de 1 jusqu'à 256 processeurs de la machine parallèle du CINES.

La colonne np indique le nombre de processeurs utilisés pour le calcul des rais. Tous les temps sont en secondes, et t_{min} et t_{max} représentent respectivement les temps de calcul minimum et maximum observés sur les np processeurs. Ces temps mesurent à la fois le temps de transfert des informations sur les rais à calculer (en mode *scatter*) ainsi que le calcul des rais. Les opérations d'entrée/sortie ainsi que la mise à jour du maillage ne sont donc pas prises en compte dans ces mesures. La colonne $speed_a = nrais/t_{max}$ représente la vitesse absolue de calcul des $nrais$ rais (348 401) effectué par les np processeurs. La colonne $speed_r = speed_a/np$ représente la vitesse relative de calcul des rais par processeur. Ces deux vitesses sont exprimées en nombre de rais par seconde. La colonne e désigne l'efficacité parallèle par rapport au cas mono-processeur.

On observe que l'efficacité baisse raisonnablement quand le nombre de processeurs augmente (figure 3.22). En effet, avec un grand nombre de processeurs la taille du jeu de données de rais à tracer par processeur devient trop faible, et le temps de transfert de ce jeu de données devient non négligeable devant le temps de calcul des rais. Notons toutefois que l'objectif est de lancer cette application sur un nombre de rais beaucoup plus important que les 350 000 de cette expérimentation, à savoir plusieurs millions de rais. Dans ce cas,

np	t_{min}	t_{max}	$speed_a$	$speed_r$	e
1	2350.67	2350.67	148.21	148.21	100.00%
2	1156.06	1207.02	288.65	144.32	97.38%
8	255.98	349.88	995.77	124.47	83.98%
16	123.16	184.12	1892.20	118.26	79.79%
24	79.48	130.42	2671.44	111.31	75.10%
32	60.61	106.79	3262.58	101.96	68.79%
48	38.56	74.10	4701.71	97.95	66.09%
64	31.87	60.63	5746.54	89.79	60.58%
128	22.82	37.27	9348.53	73.04	49.28%
256	20.29	29.00	12014.37	46.93	31.67%

TAB. 3.2 – Performance du tracé de rais sismiques sur la machine parallèle du CINES.

FIG. 3.22 – Représentation graphique de l'efficacité du tracé de rais sismiques e en fonction du nombre de processeurs.

le ratio calculs sur communications permettra d'avoir une efficacité plus grande que celle constatée ici et d'exploiter avantageusement encore plus de processeurs.

Tracé de rais dans un maillage

Dans le mode le plus fréquent d'exploitation, le tracé de rais renseigne au fur et à mesure les cellules traversées, et le stockage de certaines informations (cf. section 3.2.3) induisent un délai lié à l'accès mémoire. Avec le même jeu de données que précédemment, on cherche à connaître la scalabilité du tracé de rais en tenant compte de ces accès mémoire. Nous voulons également savoir quelle est l'influence du mode de distribution sur les performances de l'application.

Les figures 3.23(a) et 3.23(b) montrent que sur une machine parallèle comme celle du CINES, l'accélération des performances existe jusqu'à 64 processeurs avec un fort ralentissement à partir de 40 processeurs. Le mode de distribution n'influence pas le comportement général de l'application sur cette machine. A partir de 64 processeurs le temps total augmente en raison de la trop faible part de calcul par rapport aux communications. L'explication la plus plausible est le faible nombre de rais à calculer par processeur quand on emploie beaucoup de processeurs, et donc le rapport calcul/communication devient plus faible. Pour vérifier cette hypothèse nous avons doublé le jeu de données (figure 3.23(d)) et l'exécution sur la SGI Origin 3800 montre que la chute d'efficacité intervient plutôt vers 60 processeurs au lieu de 40 avec le jeu de données initial.

Les mêmes tests sur le cluster Itanium 2 (figure 3.24) ne permettent pas d'atteindre le point d'inflexion de la courbe d'accélération, en raison du nombre de processeurs limité à 50. Excepté les temps de calcul plus faibles, la scalabilité est similaire sur cette plate-forme.

Enfin, en vue de l'exécution sur des plates-formes de type grille de calcul, nous avons développé une version de l'application permettant de compresser les messages avant envoi (compression avec la zlib [ZIV & LEMPEL 77]). Nous la testons sur la machine parallèle (figure 3.23(c)) pour en connaître les effets sur ce type de plate-forme qui formera une partie des ressources de la grille. La forme générale de la courbe obtenue avec la zlib (figure 3.23(c)) est comparable à celle obtenue sans son utilisation (figure 3.23(b)). L'impact mineur de la zlib est principalement dû à la "faible" quantité de rais tracés (environ 350 000 rais). De ce fait la quantité de données à compresser lorsque le nombre de processeurs augmente, diminue et le gain obtenu dans la taille des messages devient faible au regard de la latence. Nous avons constaté le même impact mineur sur la grille, en utilisant le même jeu de données.

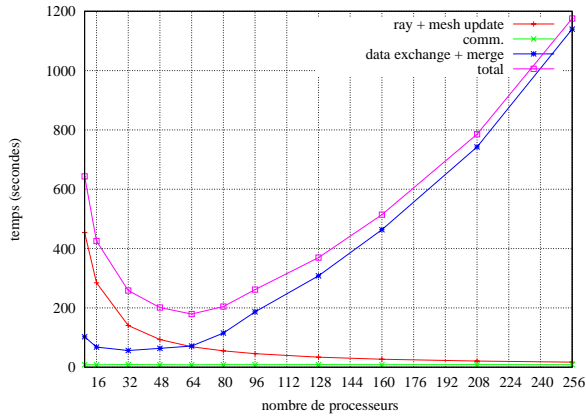
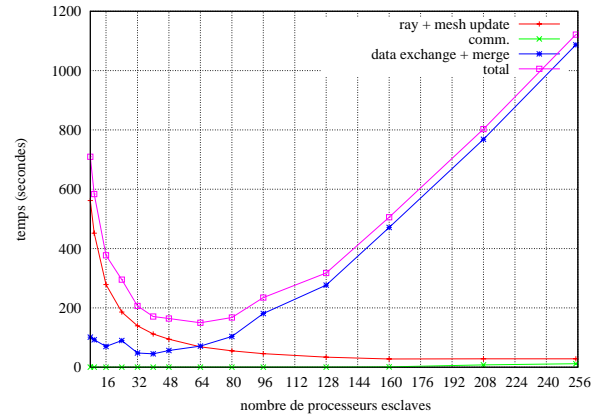
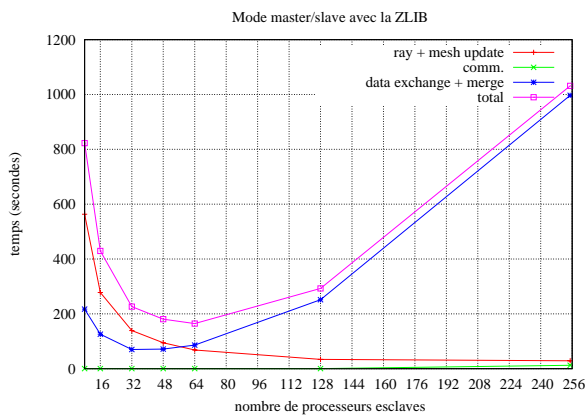
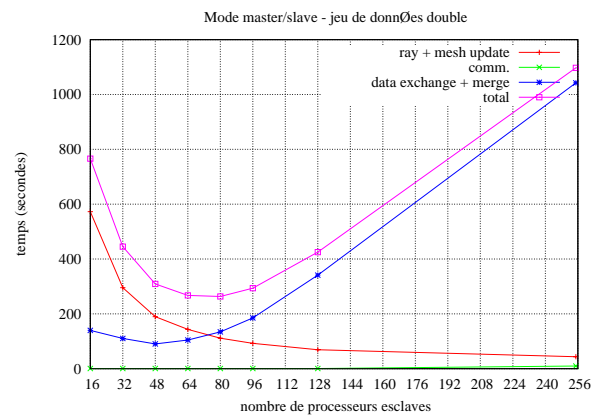
(a) Distribution des données en mode *scatter*.(b) Distribution des données en mode *maître/esclave*.(c) Distribution des données en mode *maître/esclave* avec compression (*zlib*) des messages.(d) Distribution d'un *double* jeu de données en mode *maître/esclave*.

FIG. 3.23 – Performances du tracé de rais sismiques dans un maillage global de la Terre, obtenues sur la machine parallèle SGI Origin 3800 du CINES.

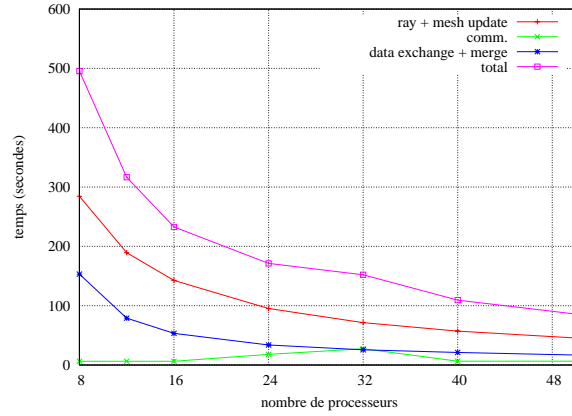
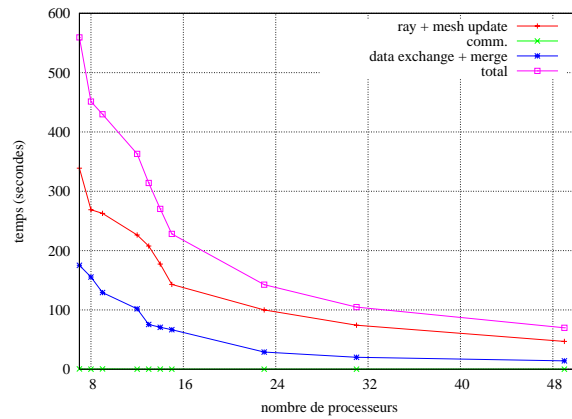
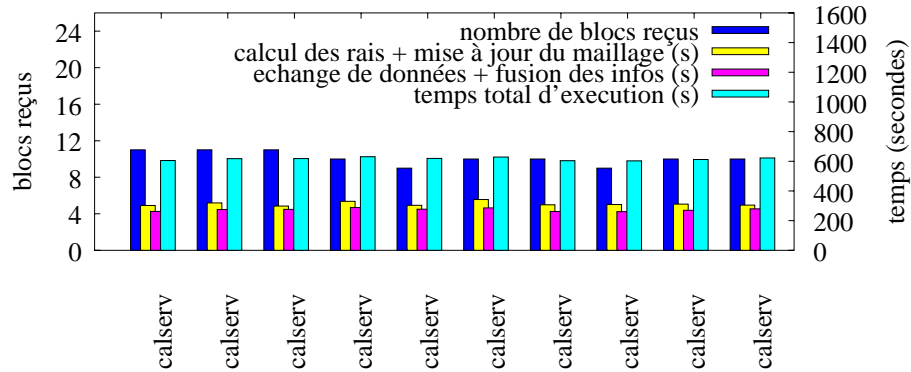
(a) Distribution des données en mode *scatter*.(b) Distribution des données en mode *maître/esclave*.

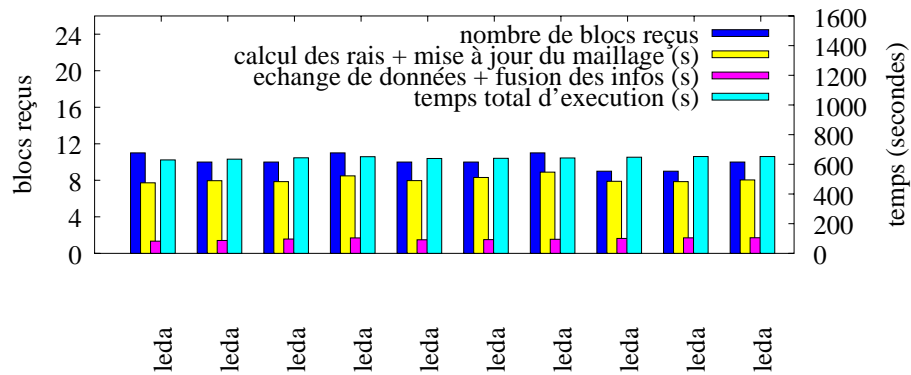
FIG. 3.24 – Performances du lancé de rais sismiques dans un maillage global de la Terre, obtenues sur le cluster HPC.

3.4.4 Série de tests avec 10 processeurs

La figure 3.25 présente, pour 10 processeurs, les temps mesurés sur le cluster (3.25(a)) et la machine parallèle (3.25(b)). Ce sont là des mesures de référence qui permettent d'évaluer l'exécution sur la configuration de type grille métropolitaine présentée sur la figure 3.26.



(a) Cluster de PC



(b) Origin 3800 - CINES, Montpellier

FIG. 3.25 – Performances sur un cluster de PC et une SGI Origin 3800 (10 processeurs).

On retrouve dans les résultats du cluster et de la machine parallèle, les spécificités de ces plates-formes. Les processeurs étant homogènes dans les deux cas, le nombre de blocs de rais distribués est équitablement réparti sur chaque processeur. Les puissances de calcul agrégées (sommées des ratios dans le tableau 3.1) utilisées sur le cluster et la machine parallèle sont respectivement 14 et 10,5. Les performances relevées pour le calcul des rais corroborent ces mesures : à nombre égal de processeurs le cluster a une puissance de calcul environ 50% supérieure à celle de la machine parallèle (en comparant les temps de calcul des rais et la mise à jour du maillage pour ces deux plates-formes, en jaune). En revanche,

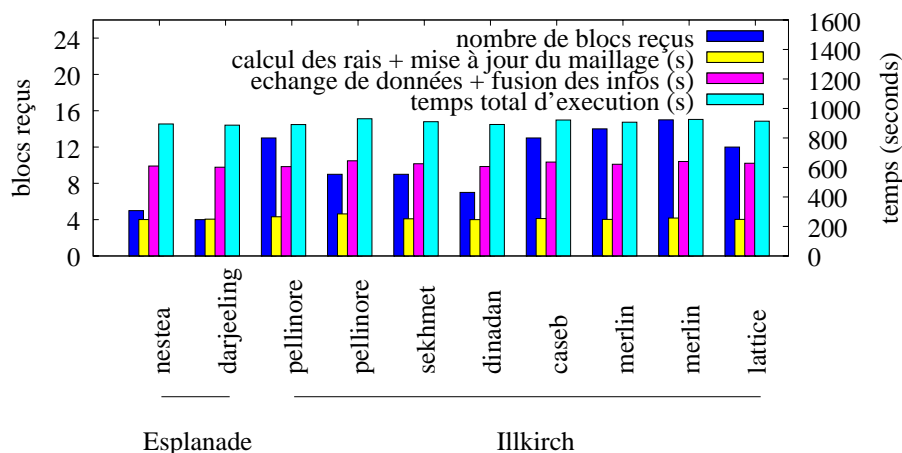


FIG. 3.26 – Performances de la configuration type grille métropolitaine de Strasbourg (10 processeurs).

le réseau d'interconnexion de la machine parallèle est extrêmement efficace par rapport à celui du cluster, et la phase d'échange des données entre processeurs est environ trois fois plus rapide sur la machine parallèle. Globalement, les temps totaux d'exécution sur le cluster et la machine parallèle sont équivalents, de l'ordre de 600 secondes.

L'exécution sur la grille métropolitaine (figure 3.26) ne met pas en défaut l'équilibrage de charge de l'application malgré l'hétérogénéité des processeurs et du réseau. Les données se trouvent initialement sur la machine pellinore du campus d'Illkirch, à partir de laquelle les blocs de rais sont distribués aux différents processeurs. Le nombre de blocs distribués à chacun des processeurs varie beaucoup (15 blocs pour le processeur le plus rapide, 4 pour le processeur distant le plus lent) mais les temps de fin sont quasi égaux, après 900 secondes. Le temps de calcul des rais, sur la grille, est légèrement meilleur que celui du cluster, ce qui s'explique par la puissance agrégée de 15,3. Par contre, de manière prévisible, la phase d'échange de données, pénalisée par le réseau, est deux fois plus lente que celle du cluster. Il en résulte que le temps global de l'application est 50% plus lent sur la grille que sur le cluster ou la machine parallèle.

3.4.5 Série de tests avec 16 processeurs

L'expérience suivante a été menée sur une même configuration de grille reliant Strasbourg et Montpellier à deux reprises. La première expérience a eu lieu en août 2002, et elle a été effectuée une deuxième fois en mai 2003. Entre-temps, l'infrastructure reliant les deux sites a progressé, comme le montre la figure 3.27. Des tests de type ping-pong nous ont permis d'évaluer qu'une application MPI pouvait disposer d'un débit moyen de 0,2 Mo/s et 0,5 Mo/s des campus strasbourgeois vers le CINES avec Renater 2. Après la connexion à

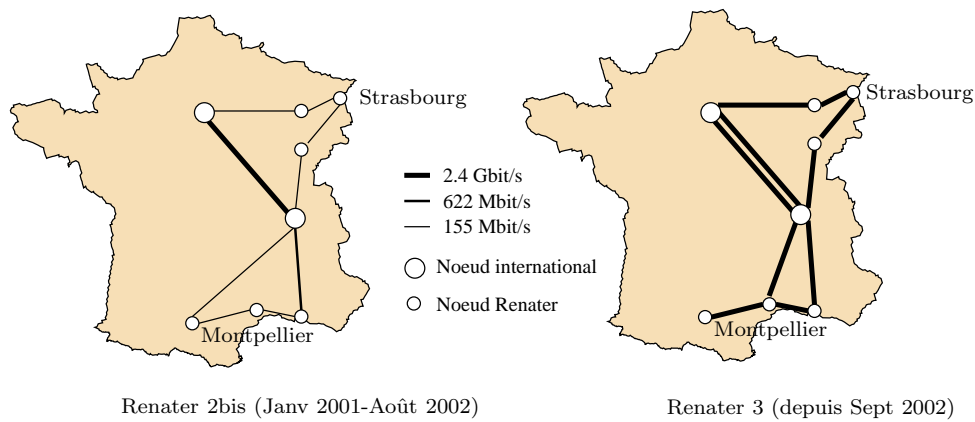
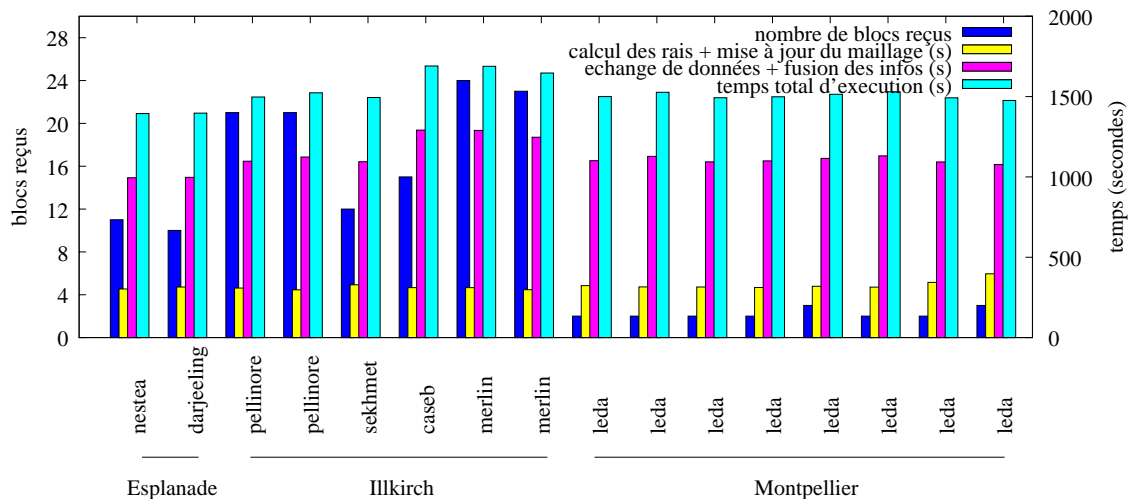


FIG. 3.27 – Partie de la dorsale nationale Renater empruntée et évolution des débits entre Renater 2bis et Renater 3.

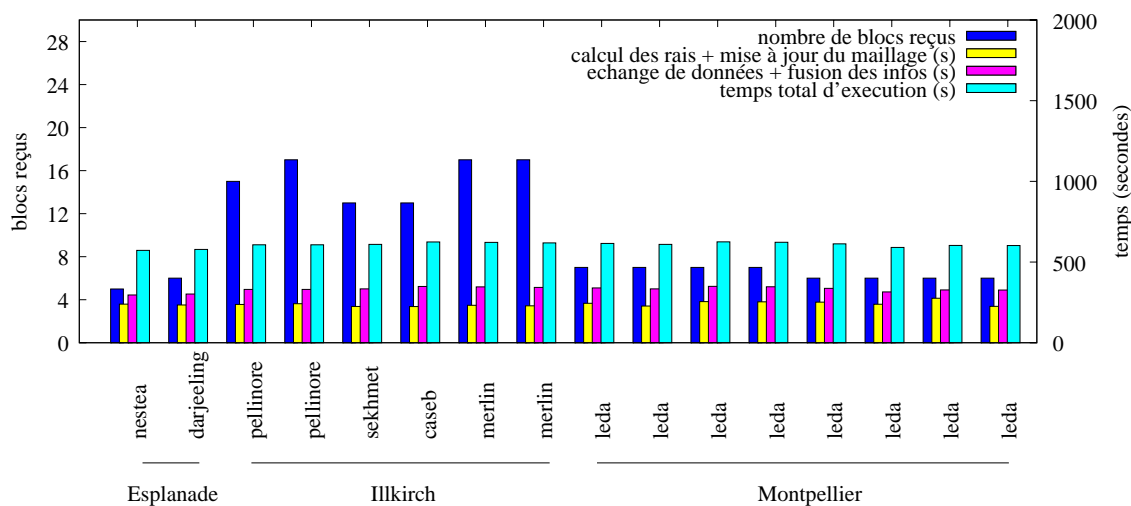
Renater 3, le débit moyen des deux campus vers le CINES approche 1 Mo/s. Les résultats sont présentés en figure 3.28(a) pour Renater 2, et figure 3.28(b) pour Renater 3.

En comparant les temps d'exécution de l'application globale, on constate que les performances ont quasiment triplé dans la nouvelle version de Renater. Les processeurs situés au CINES ont reçu au maximum, trois blocs de rais par processeur avec Renater 2, et sept avec Renater 3, les données initiales étant à Illkirch sur pellinore. On retrouve pour l'exécution avec Renater 3 comme pour la configuration métropolitaine, pour les processeurs d'un site donné, une corrélation entre la puissance relative des processeurs et le nombre de blocs reçus par processeur, à l'exclusion de pellinore qui est évidemment avantagée par la localisation des données sur son disque local. En revanche, la faible qualité de l'interconnexion entre Strasbourg et Montpellier, sous Renater 2, avait complètement déséquilibré la répartition de charge. La faible participation en calcul des processeurs du CINES ne s'explique que par la forte latence pénalisant les communications nationales (demande de travail des processeurs, envoi des blocs à traiter). Cette observation est encore plus flagrante sur la durée de l'échange des données (étapes 5 et 6) qui représentait alors 75% du temps total de l'application (contre 50% avec Renater 3).

Par ailleurs, les performances obtenues sur cette plate-forme avec une puissance de calcul agrégée proche de 21 sont comparées à une configuration de 16 processeurs sur la machine parallèle du CINES présentée en figure 3.29 (puissance cumulée de 16,8). Les temps d'exécution totaux sur ces deux plates-formes sont comparables, de l'ordre de 550 secondes pour la machine du CINES, contre 600 pour la grille nationale.



(a) Grille nationale Renater 2



(b) Grille nationale Renater 3

FIG. 3.28 – Performances avec 16 processeurs sur la grille nationale avec Renater 2 puis Renater 3.

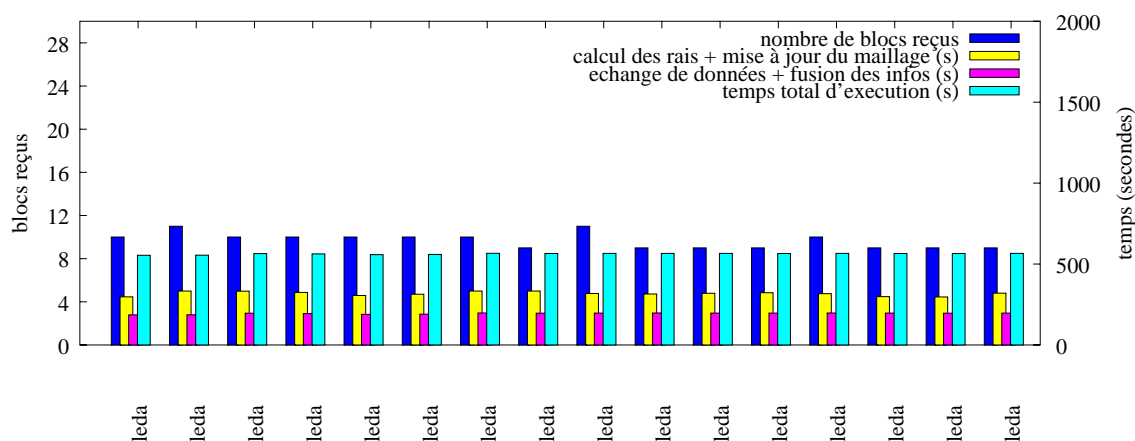


FIG. 3.29 – Performances sur une Origin 3800 (16 processeurs).

3.5 Conclusion

Dans ce chapitre, une méthode "parallèle" de tracé de rais sismiques dans un maillage global a été présentée. Elle constitue une étape fondamentale dans la construction d'un modèle de Terre par tomographie sismique. Les longueurs des rais dans chaque cellule ont été calculées, ainsi que des critères géométriques permettant de caractériser le degré d'illumination d'une cellule par les rais qui la traversent.

Ces informations sont nécessaires, pour la construction d'un maillage constitué de cellules de tailles irrégulières, appelé maillage adaptatif et présenté dans le chapitre suivant.

Des tests de performance ont été menés sur plusieurs plates-formes : machine parallèle, cluster, grille de calcul. Ils montrent notamment, au niveau de la scalabilité, la corrélation entre le nombre de rais à tracer et le nombre de processeurs optimal. Ainsi sur la machine parallèle du CINES pour le jeu de 348 401 rais l'optimal est à 64 processeurs. Au delà, la phase d'échange de données entre les processeurs (communication *tous à tous*) devient pénalisante pour les performances globales de l'application. Par ailleurs le mode maître/esclave où l'on distribue les blocs de rais "à la demande" en fonction de la charge ou de la performance des processeurs, s'avère satisfaisant du point de vue de l'équilibrage de charge pour les plates-formes hétérogènes testées.

L'approche retenue, qui distribue les rais à tracer sur les différents processeurs disponibles, permet d'envisager le traitement d'un jeu de données de très grande taille contenant éventuellement plusieurs millions de rais, répondant de ce fait aux attentes des sismologues qui s'intéressent à la réalisation de tomographie globale.

Chapitre 4

Le maillage adaptatif

Dans le chapitre précédent, différents critères géométriques ont été calculés dans un maillage régulier par couches de la Terre, afin d'évaluer l'occupation des rais sismiques dans le volume de chacune de ses cellules. C'est à partir de ces critères que l'on caractérise l'illumination d'une cellule par les rais qui la traversent.

En pratique certaines cellules sont très bien illuminées, grâce à de nombreux rais sismiques les traversant avec une bonne couverture angulaire (c'est-à-dire avec une distribution des angles bien répartie dans toutes les directions). D'autres cellules par contre, le sont beaucoup moins (par exemple lorsque les rais ont des angles d'incidence similaires conduisant à des rais quasi parallèles), voire pas du tout (lorsqu'aucun rai ne les traverse). Cette hétérogénéité de l'illumination des cellules, sur l'ensemble du maillage, est due principalement à une distribution non uniforme des séismes et des stations à la surface de notre planète (cf. figure 4.1). En effet, certaines régions du globe, en fonction de leur contexte sismotectonique, sont plus sismiques que d'autres, comme par exemple au niveau des limites de plaques tectoniques. Cette hétérogénéité est aussi due à une répartition non uniforme des capteurs sismologiques, avec notamment très peu de capteurs dans les océans (cf. figure 4.2), déséquilibrant de ce fait l'hémisphère Sud (12% des capteurs) par rapport à l'hémisphère Nord (88% des capteurs).

Pour pouvoir exploiter au mieux les données disponibles, à savoir les illuminations des cellules dans le maillage régulier initial, nous proposons dans ce chapitre une méthode originale de construction de maillage adaptatif, où le nouveau maillage créé est formé de *meta-cellules* hexaédriques par agrégation de cellules du maillage initial. La taille de ces meta-cellules est adaptée de façon à optimiser la distribution de l'illumination sur l'ensemble des meta-cellules, et donne lieu à un maillage irrégulier. Concrètement, si l'illumination d'une cellule n'est pas suffisante par rapport à un certain seuil fixé, l'algorithme de construction du maillage irrégulier tendra à l'agréger avec ses cellules voisines si l'illumi-

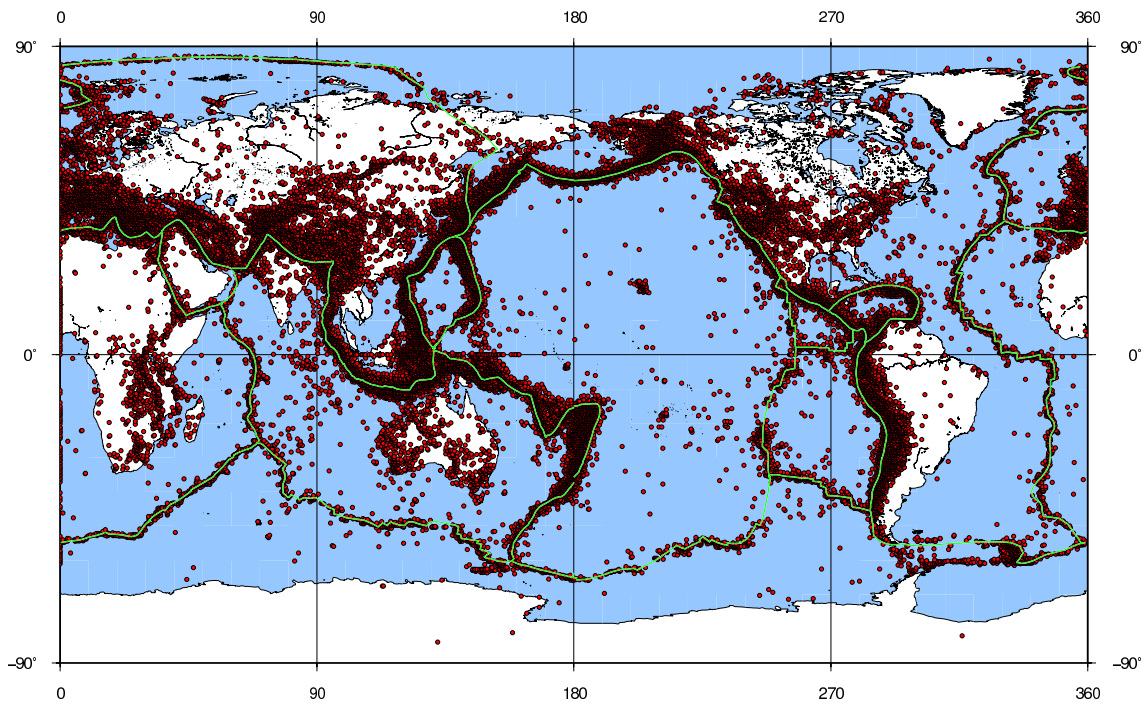


FIG. 4.1 – La sismicité mondiale de 1928 à 2004. En vert les contours des plaques tectoniques.

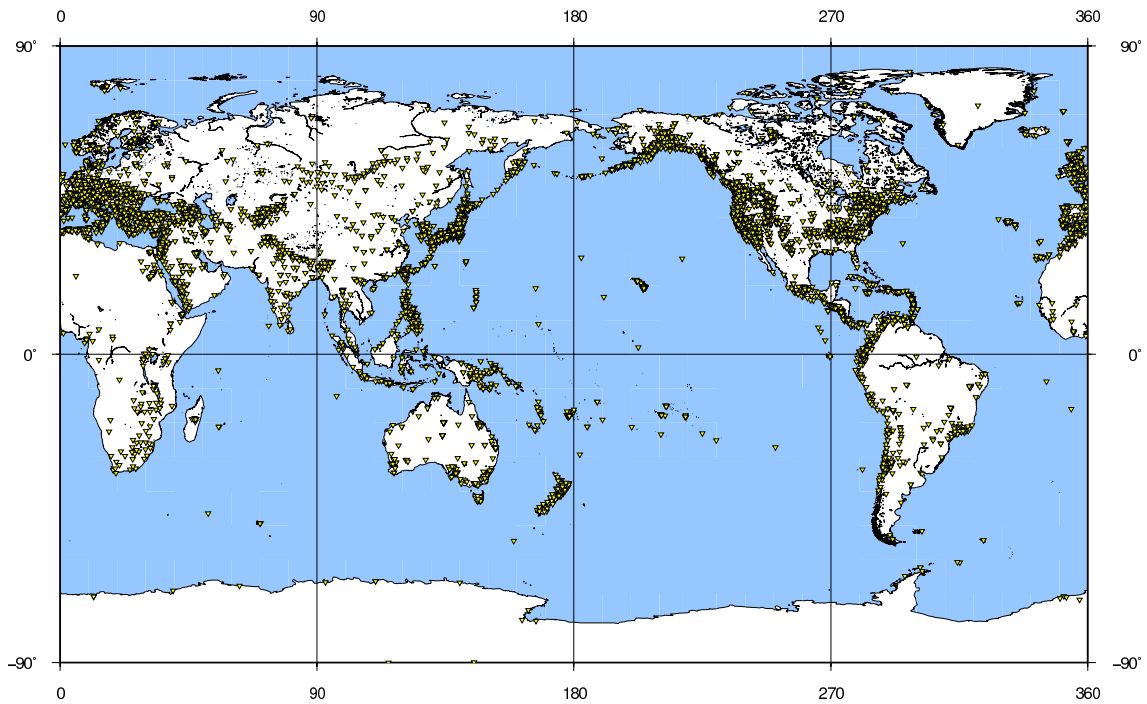


FIG. 4.2 – Répartition des stations sismologiques en activité à la fin de l'année 2004.

nation de la meta-cellule résultante est meilleure que celle de la plupart des cellules prises individuellement.

En se plaçant dans le contexte de la tomographie sismique, les avantages de ce nouveau maillage irrégulier ont un impact important sur la phase d'inversion tomographique et sur la solution obtenue. Tout d'abord, il conduit à une diminution significative du nombre d'inconnues dans le système d'équations linéaires (cf. chapitre 2.1), dont la solution permet d'affiner notre connaissance sur les vitesses de propagation des ondes sismiques, puisque chaque cellule ou meta-cellule est une inconnue de ce système. De plus, le système est plus fortement contraint, conduisant de ce fait à un meilleur conditionnement avec une meilleure stabilité des solutions obtenues ([ROWBOTHAM PETER S. & PRATT R. GERHARD 97]).

L'algorithme de construction de ce maillage irrégulier à partir du maillage initial de la Terre, est présenté dans sa version séquentielle dans la première partie de ce chapitre, et sa parallélisation est présentée dans la deuxième partie. Enfin une analyse des performances est réalisée sur deux plates-formes parallèles de type clusters dans la dernière partie.

L'implémentation séquentielle et parallèle de cet algorithme est faite dans la brique logicielle appelée MAILLEUR. Cette brique s'intègre avec celles développées dans le chapitre précédent, et leur organisation pour la construction du maillage irrégulier est présentée dans la figure 4.3.

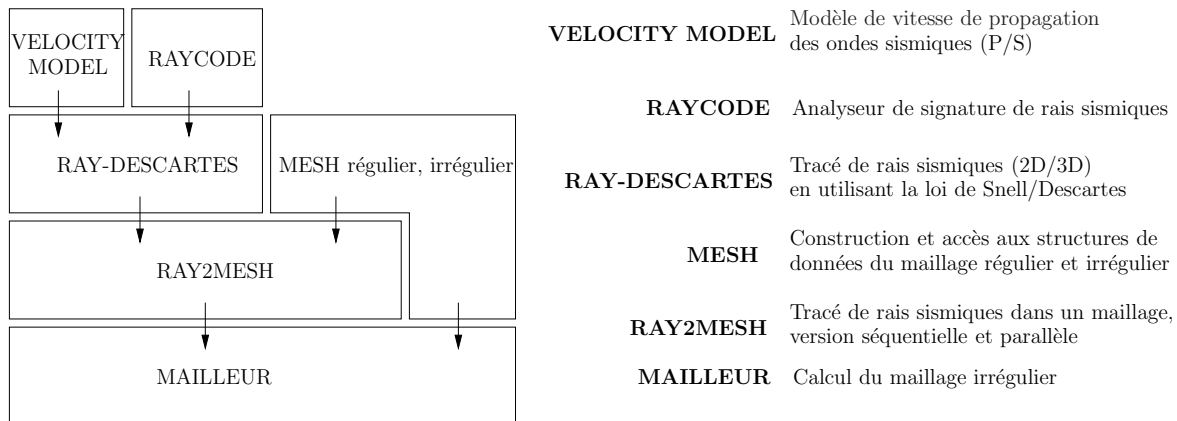


FIG. 4.3 – Organisation des briques logicielles pour la construction du maillage irrégulier.

4.1 Construction du maillage irrégulier

L'une des particularités du maillage initial est qu'il est constitué de couches concentriques, et l'adaptation de celui-ci, en un maillage irrégulier peut donc se faire indépendamment sur chacune de ses couches.

Soit Ω le maillage initial d'une couche, constitué de cellules $C_{x,y}$, où x et y sont les coordonnées en latitude et en longitude de la cellule dans le maillage Ω , la profondeur étant celle de la couche considérée¹.

Le maillage irrégulier Ω' est construit par agrégation de cellules du maillage Ω pour former des meta-cellules. L'algorithme, pour déterminer quelles cellules doivent être agrégées, doit explorer l'ensemble de toutes les solutions possibles, c'est-à-dire calculer l'illumination obtenue dans chacune d'elles. Par la suite, on appelle *configuration* une solution possible d'agrégation associée à son illumination.

Quand l'algorithme a déterminé localement quelle configuration était la meilleure, il transforme la configuration en une *meta-cellule*. Cela signifie que cette agrégation de cellules devient définitive pour le maillage irrégulier Ω' .

La meta-cellule est un objet essentiel pour lequel nous introduisons le vocabulaire et les notations suivantes.

Définition 1. Une **meta-cellule** $M_{x,y}^{p,q}$ dans le maillage irrégulier Ω' , est l'ensemble rectangulaire formé de $p \times q$ cellules adjacentes de Ω tel que la cellule $C_{x,y}$ soit située dans le coin Nord-Ouest (cf. figure 4.4). On appelle $C_{x,y}$ la *cellule de tête* de la meta-cellule $M_{x,y}^{p,q}$.

En pratique la taille maximale d'une meta-cellule est limitée par les paramètres p_{max} et q_{max} ($0 \leq p < p_{max}$ et $0 \leq q < q_{max}$), fixés par le sismologue, pour préserver une bonne résolution lors de l'inversion tomographique. Notons que pour une couche donnée ces paramètres peuvent être choisis indépendamment de ceux des autres couches. Par exemple, pour une couche située à 200 km de profondeur [SPAKMAN & BIJWAARD 01] construisent un maillage initial constitué de cellules de taille $0.6^\circ \times 0.6^\circ$, à partir desquelles ils génèrent des meta-cellules dont la taille maximale est de $6^\circ \times 6^\circ$ (ce qui correspondrait à fixer les paramètres p_{max} et q_{max} à 10).

L'algorithme de construction du maillage consiste en deux phases principales. La première est une phase de calcul, où la totalité de l'espace des configurations est explorée afin d'en déterminer les différents scores. La deuxième phase permet de sélectionner parmi ces configurations, l'ensemble des meta-cellules disjointes qui constitueront le maillage irrégulier Ω' . Notons que les contraintes géophysiques ne nécessitent ni de construire un maillage conforme, ni de paver l'intégralité de la couche considérée. Les zones sans information sismique, pouvant être ignorées, ne sont pas recouvertes par les meta-cellules. Ce point est repris en détail à la section 4.1.4.

¹Pour des raisons de lisibilité, l'indice correspondant à la profondeur d'une cellule dans la couche considérée est omis des notations.

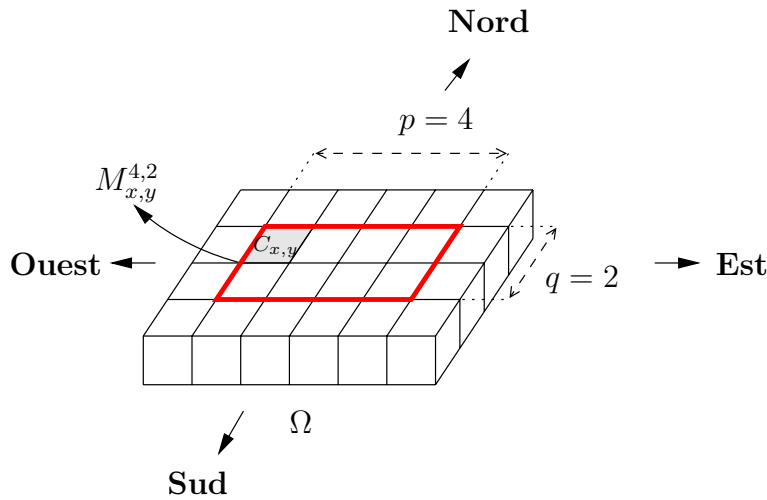


FIG. 4.4 – Représentation de la meta-cellule $M_{x,y}^{4,2}$ de taille $p \times q = 4 \times 2$ ayant pour cellule de tête $C_{x,y}$ dans le maillage Ω .

4.1.1 Évaluation de l'illumination

Les informations qui permettent d'évaluer l'illumination d'une cellule par le calcul d'un score, présenté à la section 3.2.3, sont basées sur la géométrie des rais sismiques dans le volume de la cellule. Elles sont classées en deux catégories.

1. Les informations propres à la cellule :
 - le nombre de rais dans la cellule,
 - le nombre d'impacts de rais sur les six faces de la cellule,
 - le nombre et la liste des mini-blocs traversés par les rais.

2. Les informations concernant chaque rai dans la cellule :
 - un identifiant du rai,
 - les coordonnées des points d'entrée et de sortie du rai,
 - la longueur de chaque segment de rai dans la cellule se propageant en mode de compression (P) et en mode de cisaillement (S).

Par analogie avec une cellule, ces informations peuvent être transposées à une meta-cellule $M_{x,y}^{p,q}$ dans le maillage Ω' . Celles-ci sont déterminées à partir des informations individuelles des cellules constituantes $C_{x+i,y+j}$, avec $i \in [0, p[$ et $j \in [0, q[$, dans le maillage Ω . Par exemple, le nombre d'impacts de rais sur la face Nord d'une meta-cellule $M_{x,y}^{p,q}$ se détermine en sélectionnant, parmi ses propres cellules constituantes, celles qui sont adjacentes à la face Nord de $M_{x,y}^{p,q}$. Ensuite, la somme des impacts sur la face Nord de chacune des cellules sélectionnées est réalisée.

Une fois l'ensemble de ces informations calculé pour une meta-cellule $M_{x,y}^{p,q}$, on évalue son illumination en utilisant la même fonction de score f que celle définie au chapitre 3.2.4. Le score résultant, $f(M_{x,y}^{p,q})$, est la valeur scalaire notée $s_{x,y}^{p,q}$.

4.1.2 Comparaison des scores

Pour la réalisation du maillage irrégulier, il est nécessaire d'avoir un ordre sur les scores qui permette de classer les configurations de la plus "satisfaisante" à la moins "satisfaisante" vis-à-vis d'un objectif fixé. On définit pour cela une fonction de distance, $F : \mathbb{R} \rightarrow \mathbb{R}$, telle que $F(s)$ donne la distance du score s par rapport à un score optimal s_{optim} , qui correspond à l'illumination idéale d'une meta-cellule. Dans la suite, nous appelons *meta-score* une telle fonction.

Bien sûr, en fonction de la répartition des données dans le maillage, ce score optimal ne pourra pas systématiquement être atteint dans toutes les meta-cellules. On dira qu'une meta-cellule est d'autant mieux illuminée que son score est proche du score optimal. La forme de la fonction F et le choix du score optimal nécessitent l'expertise du sismologue sur la base des informations sismiques préalablement calculées à la section 3.2.4.

Dans les tests réalisés en section 4.3, l'illumination d'une cellule est caractérisée par la densité de rais après analyse des données géophysiques telle que celle produite à la figure 3.15(d). La valeur optimale de celle-ci a été fixée à 2.5 (*ie.* $10^{2.5}$ rais par meta-cellule) et est similaire à celle utilisée par [SPAKMAN & BIJWAARD 01] (de l'ordre de 2.7).

Le meta-score est défini par $F(s) = s_{optim} - |s_{optim} - s|$. Ce choix garantit qu'une meta-cellule sera considérée comme bien illuminée si son score est proche du score optimal, puisque la fonction a son maximum pour $s = s_{optim}$. La fonction de calcul de meta-score utilisée, $F(s)$, est représentée dans la figure 4.5.

4.1.3 Calcul des configurations des meta-cellules

La première phase de construction du maillage irrégulier consiste à déterminer toutes les configurations de meta-cellules possibles. Celles-ci sont calculées pour chaque cellule du maillage initial et sont triées afin de pouvoir sélectionner aisément les meilleures meta-cellules possibles. Elles sont définies comme suit.

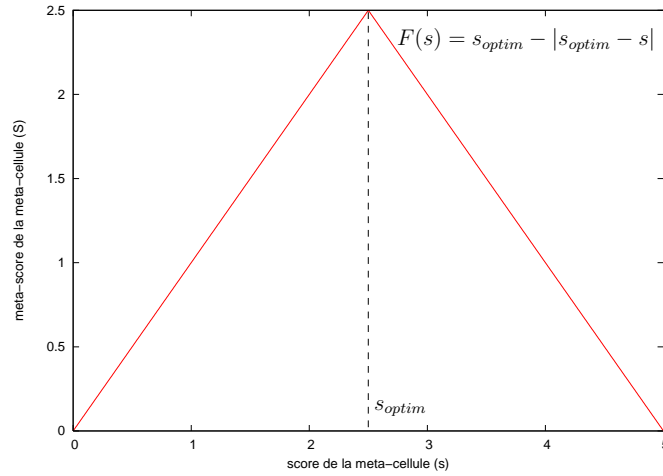


FIG. 4.5 – Représentation de la fonction de calcul du meta-score $F(s)$ en fonction du score s d’une meta-cellule, et du score optimal s_{optim} . Le maximum de $F(s)$ (ie. le meilleur meta-score) est obtenu pour $s = s_{optim}$.

Définition 2. Étant donnée une cellule $C_{x,y}$, nous appelons $\mathcal{M}_{x,y}$ la liste formée par l’ensemble des configurations possibles de meta-cellules $\{M_{x,y}^{p,q}, p \in [0, p_{max}[$ et $q \in [0, q_{max}[$ pour lesquelles les meta-scores $F(s_{x,y}^{p,q})$ ont été calculés. Cette liste contient au plus $Card(\mathcal{M}_{x,y}) = p_{max} \times q_{max}$ configurations (les meta-cellules traversées par aucun rai sont ignorées). Elle est triée selon les meta-scores décroissants.

Le premier élément de $\mathcal{M}_{x,y}$ correspond donc à la meta-cellule ayant la meilleure configuration possible en ce qui concerne l’illumination.

La phase de calcul des configurations consiste, pour chaque cellule $C_{x,y}$ du maillage Ω , à construire sa liste $\mathcal{M}_{x,y}$ ainsi qu’à calculer les meta-scores associés. C’est une étape d’autant plus lourde, en temps de calcul et en consommation mémoire, que la taille maximale $p_{max} \times q_{max}$ d’une meta-cellule est grande. Ainsi si les tailles p_{max} et q_{max} sont doublées, le nombre de configurations à calculer et à stocker est multiplié par un facteur 4. Par ailleurs, le temps de calcul $T_{x,y}$ de tous les scores associés aux différentes configurations relevant d’une cellule

de tête $C_{x,y}$ donnée dépend de p_{max} et q_{max} et s'exprime comme suit dans le cas général² :

$$\begin{aligned}
 T_{x,y} &= \sum_{p=1}^{p_{max}} \sum_{q=1}^{q_{max}} \alpha p q \\
 &= \alpha \sum_{p=1}^{p_{max}} p \sum_{q=1}^{q_{max}} q \\
 &= \alpha \frac{p_{max}(p_{max} + 1)}{2} \frac{q_{max}(q_{max} + 1)}{2} \\
 &\approx \alpha \frac{p_{max}^2}{2} \frac{q_{max}^2}{2}
 \end{aligned}$$

où α désigne le temps nécessaire à la consultation en mémoire des informations spécifiques à une cellule donnée du maillage initial.

Dans le cas où l'on utilise des scores construits à partir des informations propres à la cellule (par opposition aux informations concernant les rais dans la cellule) on considère que α est une constante. Sinon, α peut ne pas être constant dans la mesure où il pourrait dépendre du nombre de rais traversant les cellules concernées.

Si les tailles p_{max} et q_{max} sont doublées, le temps de calcul $T_{x,y}$ des scores est multiplié par un facteur 16. Rappelons qu'en pratique, les contraintes géophysiques imposent une limite aux valeurs pertinentes de p_{max} et q_{max} pour préserver la qualité de la résolution, ce qui évite une explosion des coûts en mémoire et en temps de calcul. Par ailleurs les meta-cellules de forme trop allongée (*ie.* avec un rapport longueur sur largeur très supérieur à 1) ne sont pas souhaitables d'un point de vue géophysique ([TKALCIC *et al.* 02]). Dans ce travail, le facteur de forme utilisé est de 2.5, ce qui exclut les configurations trop allongées et réduit encore le nombre de configurations à gérer. Enfin, les meta-cellules n'ayant aucun rai qui les traverse ne sont pas prises en compte dans la construction du maillage irrégulier. Ceci s'explique par le fait qu'aucune information ne permet de contraindre la valeur de leur lenteur dans le système d'équations.

Notons également que dans le cas particulier où les scores peuvent être combinés par des opérateurs associatifs, $T_{x,y}$ est inférieur à la formule présentée pour le cas général, puisqu'il suffit d'accéder une seule fois à chaque cellule pour l'ensemble des calculs concernant une cellule de tête $C_{x,y}$, et $T_{x,y} = \alpha \cdot p_{max} \cdot q_{max}$.

²La fonction de score utilisée est quelconque.

Définition 3. Nous appelons \mathcal{L} la liste des meilleures configurations associées à chaque cellule de tête $C_{x,y}$. Elle est triée selon les *meta-scores* décroissants (par la fonction `sort`), tel que :

$$\mathcal{L} = \text{sort}(\{\cup \text{Head}(\mathcal{M}_{x,y}), \forall x, y \mid C_{x,y} \in \Omega\}),$$

où $\text{Head}(m)$ est la fonction restituant l'élément en tête de la liste m .

La totalité de l'espace des configurations des meta-cellules ayant été explorée, la liste \mathcal{L} est initialisée. Par la suite, cette liste sera amenée à être modifiée; elle sera cependant gardée constamment triée tout au long de la construction du maillage irrégulier. Dans le cas où plusieurs meta-scores de valeur identique doivent être insérés dans \mathcal{L} , l'insertion pour préserver l'ordre total de la liste se fait en respectant l'ordre lexicographique sur les indices de la cellule de tête (si les meta-cellules $M_{2,7}^{4,4}$ et $M_{5,3}^{2,3}$ ont des meta-scores identiques l'ordre de ces éléments dans la liste \mathcal{L} sera : $M_{2,7}^{4,4}$ suivi de $M_{5,3}^{2,3}$).

L'exemple de la figure 4.6 montre les trois premiers éléments de la liste \mathcal{L} . Ce sont les meilleures configurations associées aux cellules de tête $C_{2,4}$, $C_{3,6}$ et $C_{6,3}$ correspondant aux meta-cellules $M_{2,4}^{3,4}$, $M_{3,6}^{2,3}$ et $M_{6,3}^{3,2}$, placées respectivement en tête des listes $\mathcal{M}_{2,4}$, $\mathcal{M}_{3,6}$ et $\mathcal{M}_{6,3}$.

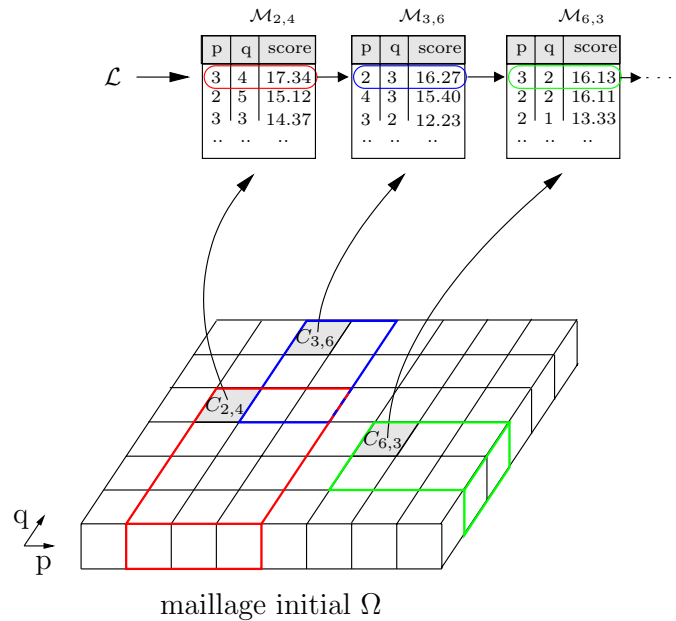


FIG. 4.6 – Exemple de configurations des meta-cellules constituant le début de la liste \mathcal{L} , juste après sa création.

4.1.4 Sélection des configurations des meta-cellules

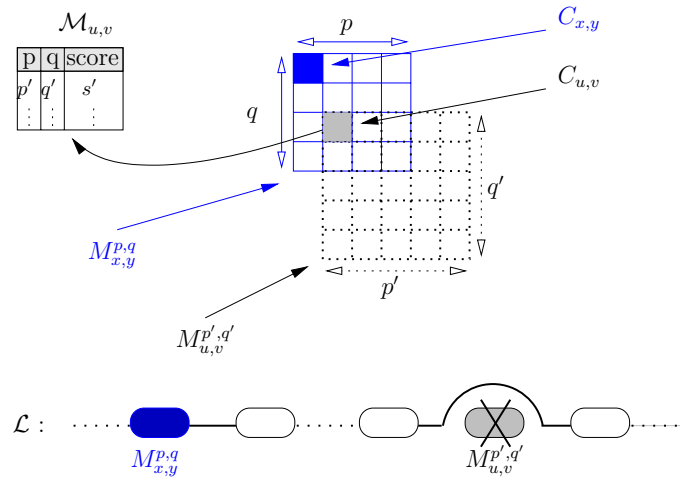
L'étape précédente a permis de trouver les meilleures configurations de meta-cellules du point de vue de l'illumination, parmi l'ensemble des configurations possibles. Elles sont triées par illumination décroissante dans la liste \mathcal{L} . L'ensemble des configurations de \mathcal{L} ne constitue évidemment pas le maillage irrégulier final puisque les meta-cellules associées à ces configurations peuvent se superposer partiellement comme cela apparaît sur la figure 4.6 pour les meta-cellules $M_{2,4}^{3,4}$ et $M_{3,6}^{2,3}$.

Aussi, la méthode employée pour construire le maillage irrégulier est itérative et opère sur la liste \mathcal{L} en utilisant deux traitements, la *sélection* et le *nettoyage*, dont les détails sont présentés par la suite. De façon succincte, la *sélection* est l'opération qui, à partir d'une configuration donnée, intègre la meta-cellule correspondante dans le maillage irrégulier, alors que le *nettoyage* est destiné à supprimer toutes les configurations dont les meta-cellules se superposent avec celle sélectionnée. À la première itération, la configuration de la meta-cellule ayant le meilleur meta-score est sélectionnée (il s'agit donc du premier élément de \mathcal{L}) pour faire partie du maillage irrégulier et devient l'élément courant. Le nettoyage des configurations devenues incompatibles est ensuite réalisé. Pour les itérations suivantes, le choix de la configuration à sélectionner est effectué en prenant l'élément immédiatement en aval de l'élément courant dans la liste \mathcal{L} et le nettoyage associé est réalisé. La construction du maillage se fait donc ainsi de suite jusqu'à ce que l'élément courant soit le dernier élément à traiter de la liste \mathcal{L} .

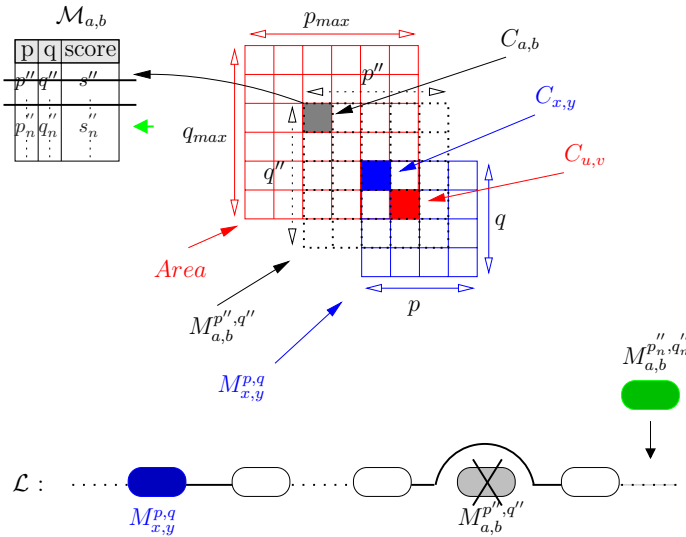
Après une présentation détaillée des opérations effectuées par la *sélection* et le *nettoyage*, nous présenterons l'algorithme de construction du maillage irrégulier.

Sélection. Étant donné l'élément courant de \mathcal{L} , la meta-cellule correspondante $M_{x,y}^{p,q}$ est sélectionnée et est immédiatement construite pour faire partie du maillage irrégulier final. Il en résulte que les cellules qui la composent ne sont plus disponibles pour faire partie d'une autre meta-cellule, sinon les meta-cellules se superposeraient. De telles meta-cellules, qui utiliseraient une cellule déjà affectée à une autre meta-cellule instanciée, ne sont donc plus valides et les configurations associées doivent être supprimées des différentes listes \mathcal{M} . Notons qu'étant donnée la méthode de construction de la liste \mathcal{L} (cf. section 4.1.3), la suppression de l'élément de tête d'une liste \mathcal{M} provoque aussi la suppression de cet élément de la liste \mathcal{L} ainsi que l'insertion de son successeur, le nouveau $\text{Head}(\mathcal{M})$, dans la liste \mathcal{L} .

Nettoyage. La suppression de telles configurations est décomposée en deux étapes : le *nettoyage en avant* et le *nettoyage en arrière*. Supposons que $M_{x,y}^{p,q}$ soit la configuration de la meta-cellule sélectionnée dans la liste \mathcal{L} pour faire partie du maillage irrégulier final.



(a) Nettoyage en avant : l'élément $M_{u,v}^{p',q'}$ est supprimé de la liste \mathcal{L} , puis la liste $\mathcal{M}_{u,v}$ est vidée.



(b) Nettoyage en arrière : l'élément $M_{a,b}^{p'',q''}$ est supprimé de la liste \mathcal{L} et de $\mathcal{M}_{a,b}$. Recherche de la nouvelle meilleure configuration de $\mathcal{M}_{a,b}$, $M_{a,b}^{p_n, q_n}$, et insertion dans \mathcal{L} .

FIG. 4.7 – La création de la meta-cellule $M_{x,y}^{p,q}$ ayant pour cellule de tête $C_{x,y}$ dans le maillage Ω nécessite de nettoyer la liste \mathcal{L} des meta-cellules devenues incompatibles avec cette création. Une première phase de nettoyage *en avant* (a), travaille sur la zone de recherche définie par $M_{x,y}^{p,q}$, tandis qu'une deuxième phase de nettoyage *en arrière* (b), travaille sur la zone de recherche *Area*.

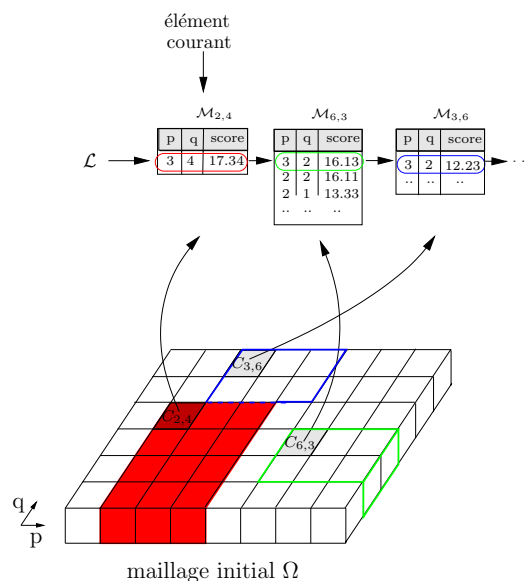
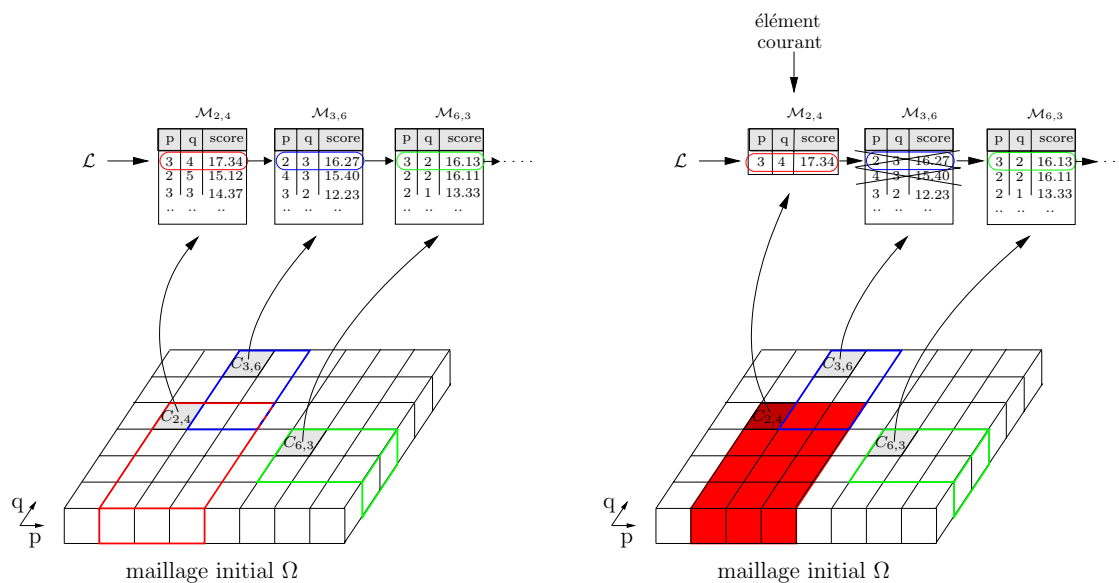
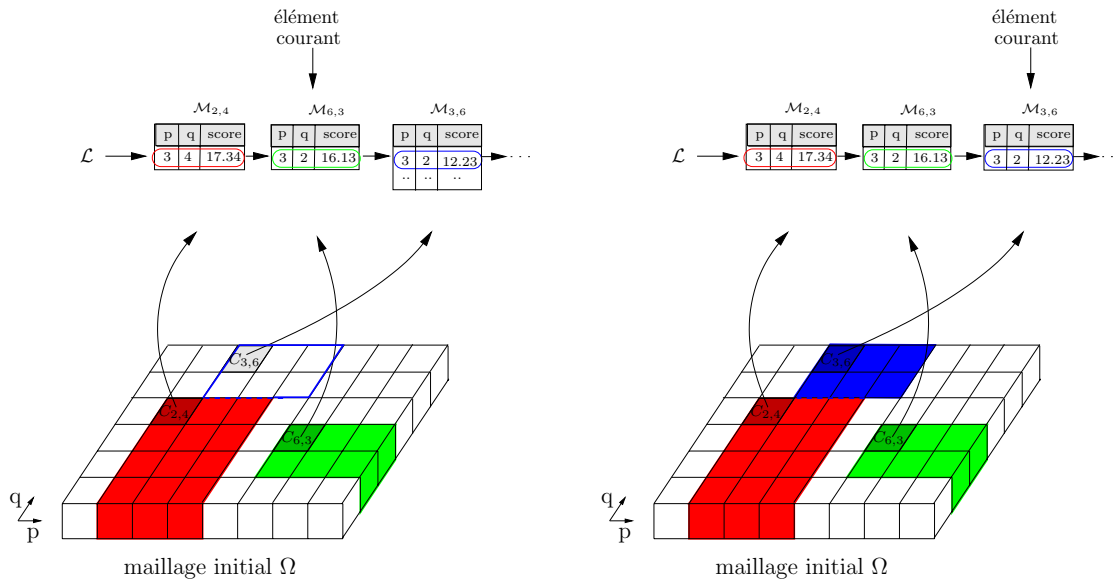
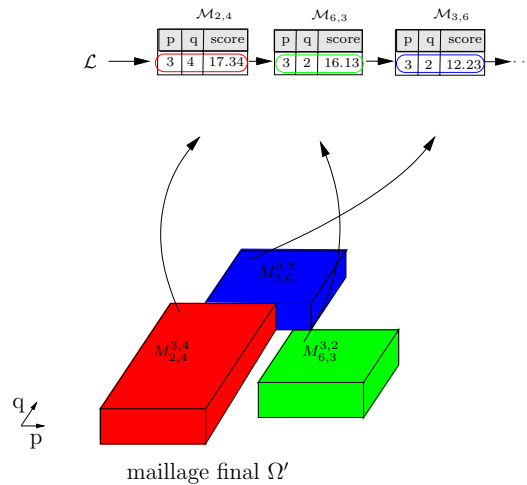


FIG. 4.8 – (Partie 1/2) – Exemple de la construction du maillage irrégulier par *sélection* et *nettoyage* sur la liste \mathcal{L} (le nettoyage arrière induit des modifications des éléments en aval de la liste non représentés sur le schéma).



(d) Sélection de $M_{6,3}^{3,2}$, suivi du nettoyage approprié.

(e) Sélection de $M_{3,6}^{3,2}$, suivi du nettoyage approprié.



(f) Vision partielle du maillage irrégulier final Ω'

FIG. 4.8 – (Partie 2/2) – Exemple de la construction du maillage irrégulier par *sélection* et *nettoyage* sur la liste \mathcal{L} (le nettoyage arrière induit des modifications sur des éléments en aval de la liste non représentés sur le schéma).

Le *nettoyage en avant* (cf. figure 4.7(a)), associé à la sélection de la meta-cellule $M_{x,y}^{p,q}$, consiste à supprimer toutes les configurations ayant pour cellule de tête l'une des cellules qui la compose. Ainsi, cet algorithme parcourt toutes les cellules $C_{u,v}$ contenues dans $M_{x,y}^{p,q}$ tel que $x \leq u < x + p$ et $y \leq v < y + q$, et pour chacune d'elles supprime la totalité des configurations de la liste $\mathcal{M}_{u,v}$. Bien évidemment l'élément qui était en tête de liste (la meilleure configuration), $\text{Head}(\mathcal{M}_{u,v})$, et qui est par définition dans la liste \mathcal{L} , en est aussitôt supprimé.

L'algorithme de *nettoyage en arrière* analyse successivement, pour chaque cellule $C_{u,v}$ de la meta-cellule sélectionnée $M_{x,y}^{p,q}$, la zone rectangulaire *Area* (en rouge sur la figure 4.7(b)) formée de $p_{max} \times q_{max}$ cellules où $C_{u,v}$ est située dans le coin Sud-est. En effet cette zone délimite l'ensemble des cellules de têtes du maillage Ω qui peuvent avoir des meta-cellules qui contiennent la cellule $C_{u,v}$. De telles meta-cellules, intersectant la meta-cellule $M_{x,y}^{p,q}$ créée, deviennent incompatibles et doivent être supprimées. Pour cela, il faut parcourir toutes les cellules de la zone de recouvrement possible *Area*, et pour chacune de ces cellules notée $C_{a,b}$ la liste associée $\mathcal{M}_{a,b}$ doit être nettoyée des configurations de meta-cellules utilisant $C_{u,v}$. Si la meilleure configuration de $\mathcal{M}_{a,b}$ a été supprimée, il faut non seulement répercuter cette suppression dans la liste \mathcal{L} , mais aussi lui rajouter par insertion triée la nouvelle meilleure configuration de $\mathcal{M}_{a,b}$ (*ie.* $\text{Head}(\mathcal{M}_{a,b})$).

Par exemple, dans la figure 4.8(b), la sélection de la meta-cellule de \mathcal{L} ayant le meilleur meta-score, $M_{2,4}^{3,4}$, a pour conséquence d'invalider plusieurs configurations de meta-cellules. En effet dans la liste $\mathcal{M}_{3,6}$, les configurations de meta-cellules $M_{3,6}^{2,3}$ (son premier élément) et $M_{3,6}^{4,3}$ (son deuxième élément) sont à supprimer puisque elles se superposent avec la meta-cellule sélectionnée $M_{2,4}^{3,4}$. Il en résulte que la nouvelle meilleure configuration de $\mathcal{M}_{3,6}$ est celle correspondant à $M_{3,6}^{3,2}$. Celle-ci est donc à insérer dans \mathcal{L} , en aval de la meta-cellule $M_{6,3}^{3,2}$, au vu de leurs meta-scores respectifs. Le processus de sélection dans la liste \mathcal{L} continue alors, en choisissant comme nouvelle meta-cellule à construire $M_{6,3}^{3,2}$, située immédiatement en aval de l'élément courant de \mathcal{L} (cf. figure 4.8(c)), suivi par la sélection de $M_{3,6}^{3,2}$ (cf. figure 4.8(d)). Une vision partielle du maillage irrégulier, produit par ces différentes opérations, associées à la sélection de trois meta-cellules ($M_{2,4}^{3,4}$, $M_{6,3}^{3,2}$, $M_{3,6}^{3,2}$) est présentée à la figure 4.8(f). Les cellules, constituant par exemple l'espace vide entre les meta-cellules $M_{2,4}^{3,4}$ (en rouge) et $M_{6,3}^{3,2}$ (en vert), seront traitées ultérieurement étant donné que les configurations correspondantes ont des scores plus petits que celui de $M_{3,6}^{3,2}$. Elles seront ignorées en cas de score nul.

Algorithme. La phase de sélection et de nettoyage avant et arrière est intégrée dans la procédure **SelectClean** ($M_{x,y}^{p,q}, \mathcal{L}$). Elle est décrite par l'algorithme 4.1, et utilise les fonctions classiques agissant sur des listes, telle que :

Head(l) restitue le premier élément de la liste l ,

Tail(l) restitue la liste l sans son premier élément,

Remove(i, \mathcal{L})

retourne la liste \mathcal{L} sans l'élément i . Elle a une complexité en $O(1)$ grâce à une implémentation utilisant un pointeur sur l'élément à effacer mémorisé sur chaque cellule du maillage initial, comme indiqué sur la figure 4.8,

InsertSorted(i, \mathcal{L})

insère l'élément i à la bonne place, en fonction de son score, dans la liste \mathcal{L} . La recherche de la position de l'élément à insérer est implémentée en utilisant une recherche dichotomique ayant une complexité en $O(\log_2 n)$, où n est la longueur de la liste \mathcal{L} .

Notons qu'à chaque sélection d'une meta-cellule de \mathcal{L} , l'algorithme de nettoyage avant et arrière supprime beaucoup de configurations de meta-cellules. Lors de la première sélection le nombre de configurations supprimées est en $O(p_{max}^2 q_{max}^2)$, mais après un certain nombre de sélections, la plupart des configurations à supprimer l'ont déjà été précédemment, et la complexité du nettoyage est en $O(1)$. Ce comportement est illustré par les figures 4.9(a) et 4.9(b), représentant le nombre cumulé de configurations supprimées, obtenues lors des tests en mode séquentiel présentés en section 4.3 sur le maillage d'une couche de 64 800 cellules, en utilisant respectivement $p_{max} = q_{max} = 10$, et $p_{max} = q_{max} = 20$. Ainsi, sur un tel maillage avec une liste initiale \mathcal{L} de $Card(\mathcal{L}) = 64\,800$ éléments, la procédure **SelectClean** a été appelée 17 827 (resp. 17 665) fois et environ 80% (resp. 95%) des configurations ont été supprimées avant la 2000^{ième} sélection.

La construction du maillage irrégulier est décrite par l'algorithme séquentiel 4.2, où la fonction **GetFirst(m)** restitue le premier élément de la liste m , et **GetNext(m)** restitue l'élément suivant à traiter dans la liste m , par rapport à l'élément courant. Cette construction consiste à sélectionner l'un après l'autre, par meta-score décroissant, les éléments de la liste \mathcal{L} , et à appeler la procédure **SelectClean** sur l'élément courant de \mathcal{L} . Après avoir sélectionné le dernier élément, la liste \mathcal{L} contient l'ensemble des configurations des meta-cellules qui constitue le nouveau maillage irrégulier Ω' , formé de meta-cellules qui ne s'intersectent pas.

Notons toutefois que ce maillage ne constitue pas un pavage de l'espace. En effet les zones dans lesquelles il n'y a aucun rai ne sont pas recouvertes puisque les meta-cellules correspondantes ne sont pas considérées. Du point de vue géophysique cette solution est tout à fait satisfaisante. En effet forcer le maillage irrégulier à paver tout l'espace y compris les zones sans rai, induirait une augmentation inutile du nombre d'inconnues du système d'inversion tomographique à résoudre.

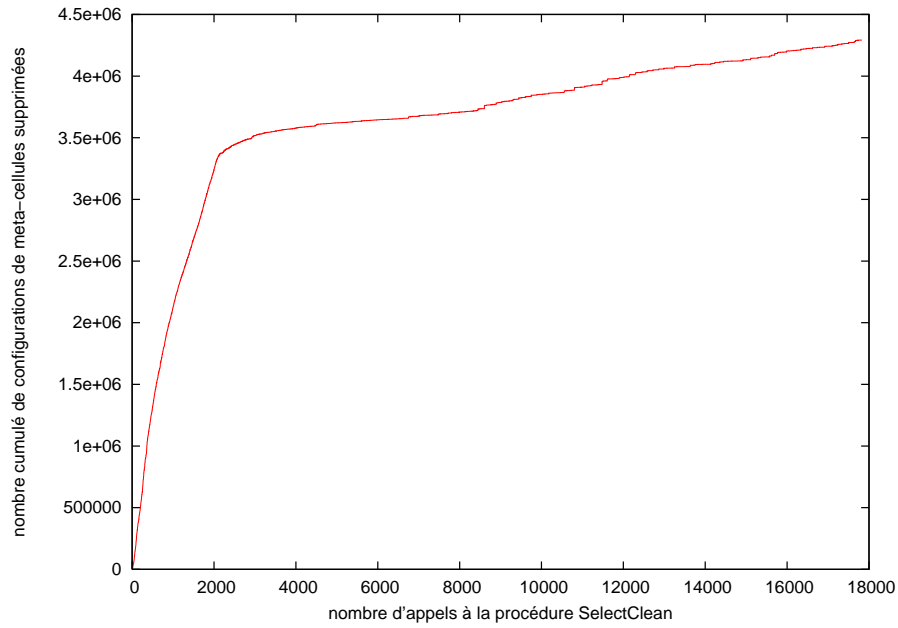
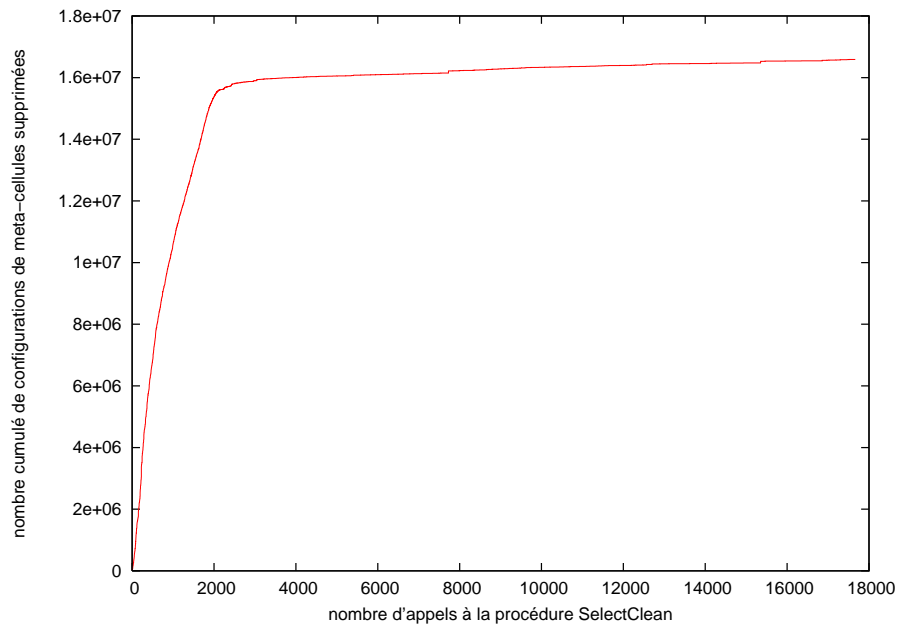
(a) $p_{max} \times q_{max} = 10 \times 10$ (b) $p_{max} \times q_{max} = 20 \times 20$

FIG. 4.9 – Nombre cumulé de configurations de meta-cellules supprimées en fonction de $p_{max} \times q_{max}$ et du nombre d'appels à la procédure `SelectClean`.

```

Input : la liste  $\mathcal{L}$ , la meta-cellule  $M_{x,y}^{p,q}$  sélectionnée
Output : la liste  $\mathcal{L}$  partiellement nettoyée
foreach  $\{(u,v) \mid C_{u,v} \in M_{x,y}^{p,q} \text{ and } C_{u,v} \neq C_{x,y}\}$  do
  // Nettoyage avant
   $M_{u,v}^{p',q'} \leftarrow \text{Head}(\mathcal{M}_{u,v})$ 
   $L \leftarrow \text{Remove}(M_{u,v}^{p',q'}, L)$ 
  Free  $(\mathcal{M}_{u,v})$ 

  // Nettoyage arrière
   $\text{Area} \leftarrow \{C_{u-i,v-j} \mid i \in [0, p_{max}[ \text{ and } j \in [0, q_{max}[ \}$ 
  foreach  $\{(a,b) \mid C_{a,b} \in \text{Area} \text{ and } C_{a,b} \neq C_{x,y}\}$  do
     $M_{a,b}^{p'',q''} \leftarrow \text{Head}(\mathcal{M}_{a,b})$ 
    if  $C_{u,v} \in M_{a,b}^{p'',q''}$  then
       $L \leftarrow \text{Remove}(M_{a,b}^{p'',q''}, L)$ 
       $\mathcal{M}_{a,b} \leftarrow \text{Tail}(\mathcal{M}_{a,b})$ 
      // Trouve une nouvelle meta-cellule valide  $M_{a,b}$  à insérer dans  $\mathcal{L}$ 
       $\text{found} = \text{False}$ 
      while not  $\text{found}$  and  $\mathcal{M}_{a,b} \neq \emptyset$  do
         $M_{a,b}^{p'',q''} \leftarrow \text{Head}(\mathcal{M}_{a,b})$ 
        if  $C_{u,v} \notin M_{a,b}^{p'',q''}$  then
           $\mathcal{L} \leftarrow \text{InsertSorted}(M_{a,b}^{p'',q''}, L)$ 
           $\text{found} = \text{True}$ 
        else
           $\mathcal{M}_{a,b} \leftarrow \text{Tail}(\mathcal{M}_{a,b})$ 
        endif
      endw
    endif
  endfch
endfch

```

Algorithme 4.1 : La procédure $\text{SelectClean}(M_{x,y}^{p,q}, \mathcal{L})$ valide l'élément $M_{x,y}^{p,q}$ de la liste \mathcal{L} , en procédant à un nettoyage *avant* et *arrière*. L'élément courant de la liste \mathcal{L} est positionné sur $M_{x,y}^{p,q}$.

```

Input : le maillage  $\Omega$  avec ses cellules  $C_{x,y}$ , les listes de meta-cellules  $\mathcal{M}_{x,y}$  et la liste  $\mathcal{L}$  initiale.
Output : la liste  $\mathcal{L}$  nettoyée, ne contenant que les meta-cellules qui constituent le nouveau maillage irrégulier  $\Omega'$ .
 $M_{x,y}^{p,q} \leftarrow \text{GetFirst}(\mathcal{L})$ 
while  $M_{x,y}^{p,q} \neq \emptyset$  do
   $\mathcal{L} \leftarrow \text{SelectClean}(M_{x,y}^{p,q}, \mathcal{L})$ 
   $M_{x,y}^{p,q} \leftarrow \text{GetNext}(\mathcal{L})$ 
endw

```

Algorithme 4.2 : Procédure principale du maillage séquentiel.

4.2 Algorithme parallèle de construction du maillage irrégulier

La construction d'un maillage irrégulier à partir d'un maillage régulier initial de la Terre, tel que décrit précédemment, nécessite une mémoire importante pour charger l'ensemble des informations associées aux rais qui traversent les cellules du maillage. De plus, l'exploration exhaustive des configurations des meta-cellules, ainsi que la détermination des scores associés, sont très consommatrices en ressources de calcul. C'est pourquoi la parallélisation de cet algorithme a été effectuée.

Cette parallélisation consiste à subdiviser une couche Ω du maillage régulier initial en un ensemble de sous-domaines $\Omega_{i,j}$, pris en charge par les processeurs $Proc_{i,j}$. Pour chaque sous-domaine $\Omega_{i,j}$, le processeur $Proc_{i,j}$ calcule d'abord la liste $\mathcal{L}_{i,j}$ des meta-cellules initiales. Ensuite, il opère les *sélections* et les *nettoyages* sur la liste $\mathcal{L}_{i,j}$ tels que décrits par la procédure `SelectClean` à la section 4.1.4, tour à tour, afin de construire le maillage irrégulier $\Omega'_{i,j}$ associé au sous-domaine $\Omega_{i,j}$. Le maillage irrégulier final est l'union des maillages partiels $\Omega'_{i,j}$.

4.2.1 Subdivision du maillage

La subdivision d'une couche Ω en sous-domaines dépend du nombre n de processeurs utilisés lors de l'exécution de l'application. De façon à répartir la charge en mémoire et en calcul sur les différents processeurs, les sous-domaines sont constitués d'un nombre équivalent de cellules.

La subdivision est donc réalisée en découpant régulièrement la couche Ω en l secteurs angulaires dans le sens des latitudes, et en L secteurs dans le sens des longitudes, tel que :

- $l \times L = n$,
- $\cup \Omega_{i,j} = \Omega$,
- et $\cap \Omega_{i,j} = \emptyset$.

Chaque sous-domaine a donc au plus huit voisins.

Par exemple, la décomposition d'une couche du maillage régulier en 16 sous-domaines est illustrée par la figure 4.10 pour $l = 4$ et $L = 4$.

4.2.2 Les zones de recouvrement

La construction par morceau du maillage irrégulier, c'est-à-dire sur chacun des sous-domaines $\Omega_{i,j}$, requiert d'avoir non seulement les cellules nécessaires à la création de la



FIG. 4.10 – Subdivision d’une couche Ω du maillage régulier, en 16 sous-domaines.

liste de meta-cellules $\mathcal{L}_{i,j}$, mais aussi les cellules nécessaires pour activer correctement la fonction `SelectClean` sur les meta-cellules sélectionnées. D’où la nécessité de définir une zone de recouvrement entre sous-domaines adjacents. De plus, il est important que tout processeur $Proc_{i,j}$ ait connaissance des meta-cellules situées dans les sous-domaines voisins dont la sélection pourrait interférer avec ses propres meta-cellules. C’est pourquoi les meta-cellules, dont la cellule de tête n’appartient pas au sous-domaine $\Omega_{i,j}$, mais dont la création pourrait interférer avec une meta-cellule de $\Omega_{i,j}$, sont également intégrées à la liste $\mathcal{L}_{i,j}$.

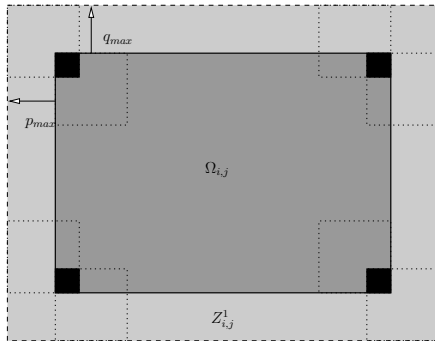
Il en résulte que la taille de la zone de recouvrement, qui doit être mise en place entre les sous-domaines se juxtaposant, est liée à la taille maximale $p_{max} \times q_{max}$ que peut avoir une meta-cellule, puisqu’elle délimite à la fois l’aire de recherche dans les phases de nettoyage, et la zone d’interférence avec d’autres meta-cellules.

Soit $Z_{i,j}^1$ la zone de recouvrement, entourant le sous-domaine $\Omega_{i,j}$, formée de :

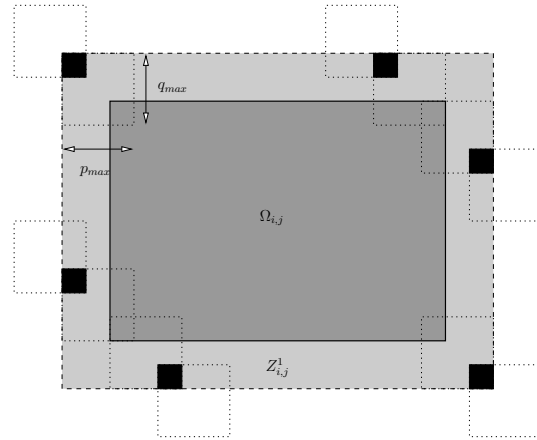
- $p_{max} - 1$ cellules à l’Est et à l’Ouest du sous-domaine,
- et de $q_{max} - 1$ cellules au Nord et au Sud.

Cette zone $Z_{i,j}^1$, ajoutée au sous-domaine $\Omega_{i,j}$, définit l’ensemble des cellules de tête $C_{x,y}$ pour lesquelles les meta-cellules $\text{Head}(\mathcal{M}_{x,y})$ permettent de créer la liste $\mathcal{L}_{i,j}$ (cf. figure 4.11(a)).

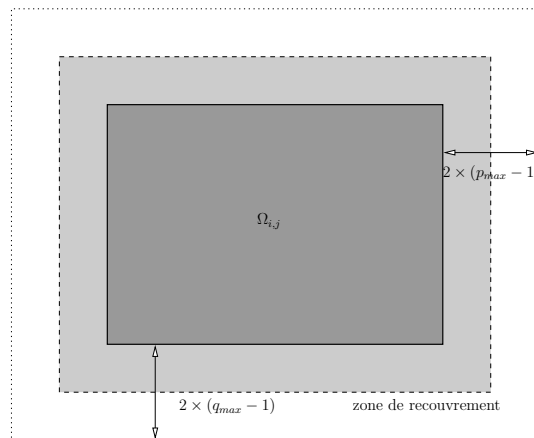
En ajoutant une deuxième zone de recouvrement $Z_{i,j}^2$, de taille identique, autour de $Z_{i,j}^1 \cup \Omega_{i,j}$, on garantit que les cellules nécessaires à la construction des meta-cellules de $\mathcal{L}_{i,j}$ ainsi qu’aux nettoyages avant et arrière, sont disponibles (cf figure 4.11(b)). On note $D_{i,j}$ le sous-domaine, illustré par la figure 4.11(c), formé de $\Omega_{i,j}$ ainsi que de ses deux zones de recouvrement $Z_{i,j}^1$ et $Z_{i,j}^2$, tel que $D_{i,j} = \Omega_{i,j} \cup Z_{i,j}^1 \cup Z_{i,j}^2$.



(a) $\Omega_{i,j}$ et sa zone de recouvrement $Z_{i,j}^1$ (en gris clair) permettant le calcul de $\mathcal{L}_{i,j}$.



(b) Visualisation des aires de nettoyage avant et arrière pour les meta-cellules en bordure du sous-domaine $\Omega_{i,j} \cup Z_{i,j}^1$.



(c) $\Omega_{i,j}$ et ses zones de recouvrement $Z_{i,j}^1$ (en gris clair) et $Z_{i,j}^2$ (en blanc), définissant le sous-domaine $D_{i,j}$.

FIG. 4.11 – Identification et construction de la zone de recouvrement $(Z_{i,j}^1 \cup Z_{i,j}^2)$ du maillage $\Omega_{i,j}$.

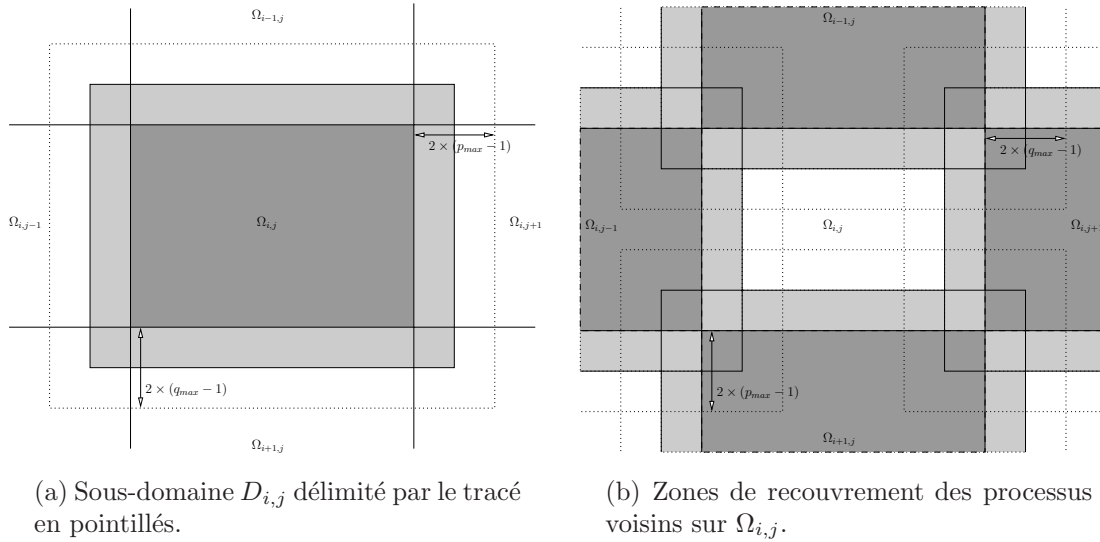


FIG. 4.12 – Sous-domaine $D_{i,j}$ ainsi que les zones de recouvrement liées aux domaines voisins.

4.2.3 Construction de la liste $\mathcal{L}_{i,j}$

La liste $\mathcal{L}_{i,j}$ est triée par meta-score décroissant et est construite sur le domaine défini par $\Omega_{i,j} \cup Z_{i,j}^1$ de la même façon que dans le mode séquentiel :

$$\mathcal{L}_{i,j} = \{\cup \text{Head}(\mathcal{M}_{x,y}), \forall x, y \mid C_{x,y} \in \Omega_{i,j} \cup Z_{i,j}^1\}$$

4.2.4 Sélection et nettoyage

L'opération de sélection et de nettoyage (avant et arrière) sur la liste $\mathcal{L}_{i,j}$, effectuée par chaque processeur $Proc_{i,j}$, se comporte principalement de la même façon qu'en mode séquentiel.

Cependant, à chaque fois que le processeur $Proc_{i,j}$ sélectionne une meta-cellule $M_{x,y}^{p,q}$, dont la cellule de tête est située dans $\Omega_{i,j}$ et dont le nettoyage interfère avec au moins un des domaines adjacents à $\Omega_{i,j}$ (cf. figure 4.12), il doit en informer le ou les processeurs adjacents, cette sélection ayant une influence sur l'évolution de la liste \mathcal{L} de ses voisins. Pour cela, il envoie un message permettant de définir la meta-cellule sélectionnée, puis poursuit son traitement, l'envoi étant non bloquant. Ce message est un quintuplet $(x, y, p, q, F(s_{x,y}^{p,q}))$ caractérisant la meta-cellule $M_{x,y}^{p,q}$, ayant $C_{x,y}$ pour cellule de tête. La taille de la meta-cellule est de $p \times q$, avec un meta-score $F(s_{x,y}^{p,q})$. Les processeurs qui reçoivent ce message

provenant de $Proc_{i,j}$ insèrent la configuration de la meta-cellule correspondante dans leur liste \mathcal{L} respective, tout en la maintenant triée. Lorsque cette configuration sera l'élément courant de \mathcal{L} , les opérations de nettoyage associées seront réalisées.

D'autre part, lorsque le processeur $Proc_{i,j}$ sélectionne une meta-cellule de $\mathcal{L}_{i,j}$ ayant sa cellule de tête située dans la zone de recouvrement $Z_{i,j}^1$ de $\Omega_{i,j}$, il n'est pas autorisé à poursuivre son traitement, puisque c'est l'un de ses voisins qui en est chargé. Le processeur $Proc_{i,j}$ se met donc en attente d'information (c'est-à-dire la réception bloquante d'un message), provenant de son voisin, sur le devenir de cette meta-cellule. Celle-ci est soit *sélectionnée*, soit *invalidée* lors d'une phase de nettoyage. Si la meta-cellule est sélectionnée par le processeur voisin, $Proc_{i,j}$ recevra le message correspondant, et il pourra donc poursuivre la phase de sélection sur cette meta-cellule. Par contre, si la meta-cellule est invalidée par le processeur voisin, c'est que celui-ci a trouvé une meilleure meta-cellule à sélectionner dans son sous-domaine. Il enverra donc un message au processeur $Proc_{i,j}$ concernant cette configuration sélectionnée. Le processeur $Proc_{i,j}$ recevra alors un message dont le meta-score $F(s_{x,y}^{p,q})$ sera plus élevé que celui de la configuration de l'élément courant. La configuration associée à ce message sera donc placée en amont de l'élément courant dans la liste $\mathcal{L}_{i,j}$, et deviendra le nouvel élément courant de $\mathcal{L}_{i,j}$. Sa sélection aura pour effet d'invalider la meta-cellule sur laquelle le processeur $Proc_{i,j}$ était en attente précédemment.

Étant donné que les différentes listes $\mathcal{L}_{i,j}$ sont systématiquement gardées triées par les différents processeurs tout au long du processus décrit ci-dessus, celui-ci se déroule sans inter-blocage. Rappelons qu'un processeur $Proc_{i,j}$ est en attente, si la configuration de son élément courant, qui est la meilleure localement à ce processeur, est dans sa zone de recouvrement. Plusieurs processeurs peuvent donc se trouver en attente si leur configuration courante est sur leur zone de recouvrement. Mais tous les processeurs ne peuvent être simultanément en attente puisque l'un d'eux possède dans son sous-domaine la meilleure configuration globalement c'est à dire la meilleure sur l'ensemble des configurations courantes des processeurs. Ce processeur peut donc traiter cette configuration et éventuellement les suivantes, si elles sont toujours les meilleures globalement, et amorcer le déblocage de ses voisins.

Dans le cas extrême où tous les processeurs auraient à considérer à un instant donné des configurations ayant toutes le même meta-score, l'inter-blocage n'est pas possible. En effet, l'ordre d'insertion dans la liste triée \mathcal{L} étant renforcé en cas d'égalité de meta-score par l'ordre lexicographique sur les coordonnées globales des cellules de tête associées à chaque meta-cellule, l'un au moins des processeurs aura comme élément courant une configuration de son propre sous-domaine $\Omega_{i,j}$. Ce processeur ne sera pas en attente et les traitements qu'il réalisera débloquent ultérieurement les autres processeurs.

Une fois le processus de sélection-nettoyage terminé, les listes $\mathcal{L}_{i,j}$ sont fusionnées pour constituer le maillage irrégulier $\Omega'_{i,j}$. Certaines meta-cellules peuvent alors appartenir à

plusieurs sous-domaines, lorsqu'elles sont situées dans des zones de recouvrement, et l'étape de fusion doit bien évidemment ne les ajouter qu'une seule fois dans le maillage irrégulier final.

L'ensemble de ce processus est décrit par l'algorithme 4.3, qui utilise plusieurs fonctions agissant soit sur un message msg ou sur une liste de messages l , soit sur une meta-cellule m tel que :

GetMetaCell(msg)

restitue la meta-cellule dont les informations de position, de taille et de meta-score sont contenues dans le message msg .

GetScore(msg)

restitue la valeur du meta-score contenue dans le message msg .

SendMsg(msg, n)

envoie, en mode non bloquant, le message msg au processus n .

RcvMsg(n)

réceptionne le message provenant du processeur n .

AddMsg(msg, l)

ajoute à la liste de messages l le message msg .

SortByScore(l)

trie les messages de la liste l par meta-scores décroissants, et en cas d'égalité de meta-scores par ordre lexicographique sur les coordonnées de la cellule de tête.

isInvalidated(m)

retourne la valeur *vrai* si la meta-cellule m a été invalidée, sinon retourne la valeur *faux*. La meta-cellule m est invalidée si la liste courante \mathcal{M} des configurations associées à m est telle que $Card(\mathcal{M}) = 0$.

4.3 Résultats expérimentaux

Afin d'analyser le comportement de cette application, en termes de performance et de scalabilité, une série de tests a été effectuée sur deux architectures parallèles, un cluster de 6 bi-processeurs PC/Xeon à 1,7 Ghz et 1 Go de RAM, et un cluster de bi-processeurs Itanium 2 à 1,3 Ghz et 8 Go de RAM dont les descriptions complètes sont présentées en section 3.4.2 (cluster EOST et HPC).

Ces tests correspondent à l'exécution de l'application sur plusieurs processeurs, en utilisant un même jeu de données, et en définissant la taille maximale $p_{max} \times q_{max}$ d'une meta-cellule à 10×10 et 20×20 . Les résultats des tests sont présentés dans les tableaux 4.1 et 4.2. Tous les temps mesurés sont en secondes, et les temps reportés sont les temps minimum et maximum correspondant aux processeurs le plus rapide et le plus lent parmi l'ensemble des processeurs. De même, les tailles mémoires mesurées sont en méga-octets et les valeurs reportées correspondent à l'occupation mémoire (application et données) la plus faible et la plus forte parmi l'ensemble des processeurs.

```

Input : le sous-domaine  $D_{i,j}$  avec les cellules  $C_{x,y}$ , les listes de meta-cellules  $\mathcal{M}_{x,y}$  et la liste  $\mathcal{L}_{i,j}$  initiale.
Output : la liste  $\mathcal{L}_{i,j}$  nettoyée, ne contenant que les meta-cellules qui constituent le nouveau maillage irrégulier  $\Omega'_{i,j}$ .
MsgList  $\leftarrow \emptyset$ 
 $M_{x,y}^{p,q} \leftarrow \text{GetFirst}(\mathcal{L}_{i,j})$ 
while  $M_{x,y}^{p,q} \neq \text{null}$  do
  // La meta-cellule  $M_{x,y}^{p,q}$  est sélectionnée
  if  $C_{x,y} \in \Omega_{i,j}$  then
    // le processus  $P_{i,j}$  est responsable de la création de la meta-cellule  $M_{x,y}^{p,q}$ .
    foreach  $\{(k,l) \mid k \in [i-1, i+1] \wedge l \in [j-1, j+1] \wedge (k,l) \neq (i,j)\}$  do
      if  $M_{x,y}^{p,q} \cap D_{k,l} \neq \emptyset$  then
        // la meta-cellule "déborde" sur le sous-domaine voisin
         $msg \leftarrow (x, y, p, q, S_{x,y}^{p,q})$ 
        // envoi non bloquant au processeur adjacent
        SendMsg( $msg, P_{k,l}$ )
      endif
    endforeach
    // suppression des éléments de  $\mathcal{L}_{i,j}$  ayant des cellules en commun avec  $M_{x,y}^{p,q}$ 
    SelectClean( $M_{x,y}^{p,q}, \mathcal{L}_{i,j}$ )
  else
    // le proc.  $P_{i,j}$  a dû/doit recevoir un msg du proc. responsable de la création de  $M_{x,y}^{p,q}$ .
    reached  $\leftarrow \text{False}$ 
    while reached = False do
      // réception des messages des processeurs voisins
      foreach  $\{(k,l) \mid k \in [i-1, i+1] \wedge l \in [j-1, j+1] \wedge (k,l) \neq (i,j)\}$  do
        while  $\exists$  messages de  $P_{k,l}$  do
           $msg \leftarrow \text{RcvMsg}(P_{k,l})$ 
          AddMsg( $msg, \text{MsgList}$ );
        endw
      endforeach
      // trie la liste de messages MsgList, par meta-scores décroissants
      SortByScore(MsgList)

      // traitement des messages
       $msg \leftarrow \text{Head}(\text{MsgList})$ 
      while  $msg \neq \emptyset$  and  $S_{x,y}^{p,q} \leq \text{GetScore}(msg)$  and reached = False do
        // traitement des configurations "en amont" de l'élément courant
        // nettoyage avant et arrière, selon msg, dans le domaine  $D_{i,j}$ 
         $M_{x',y'}^{p',q'} \leftarrow \text{GetMetaCell}(msg)$ 
        SelectClean( $M_{x',y'}^{p',q'}, \mathcal{L}_{i,j}$ )
        if isInvalidated( $M_{x,y}^{p,q}$ ) or  $msg \equiv M_{x,y}^{p,q}$  then reached  $\leftarrow \text{True}$ 
        else reached  $\leftarrow \text{False}$ 
        MsgList  $\leftarrow \text{Tail}(\text{MsgList})$ 
         $msg \leftarrow \text{Head}(\text{MsgList})$ 
      endw
    endw
  endif
   $M_{x,y}^{p,q} \leftarrow \text{GetNext}(\mathcal{L}_{i,j})$ 
endw

```

Algorithme 4.3 : Maillage parallèle pour le processus $P_{i,j}$.

Les différentes valeurs mesurées, en fonction du nombre de processeurs, sont :

1. le temps d'exécution total de l'application (colonne **total**),
2. les tailles mémoires minimum et maximum utilisées par l'application après le chargement des données,
3. les temps minimum et maximum nécessaires pour calculer les configurations des meta-cellules et construire la liste \mathcal{L} (colonne **configuration**),
4. les temps minimum et maximum pour sélectionner et nettoyer les éléments de la liste \mathcal{L} (colonne **sélection**),
5. les temps cumulés minimum et maximum d'attente de messages, précisés dans la colonne **inactivité** (ce temps d'attente est donc inclus dans la phase de sélection),
6. l'**accélération**.

4.3.1 Implémentation

Les temps mesurés proviennent de l'application compilée avec le compilateur `icc8.1` en utilisant la bibliothèque `MPI-GM` de Myricom sur le cluster Itanium 2, et `gcc2.96` avec `LAM/MPI` sur le cluster Xeon. Pour ces deux architectures, une optimisation `03` a été utilisée.

4.3.2 Jeu de données et paramètres de l'application

Le jeu de données sur lequel les tests ont été effectués, provient du tracé de rais sismiques dans un maillage régulier, calculés selon la méthode décrite au chapitre 3. Il s'agit d'environ 350 000 rais tracés qui proviennent de la sismicité mondiale de l'année 1999, extraits des bases de données de l'ISC³. Le maillage initial utilisé est régulier sur chacune des ses 11 couches telles que définies dans le tableau 3.3 dans le chapitre précédent, et est constitué par couche de 64 800 cellules dont la taille est de $1^\circ \times 1^\circ$.

Les tests ont été menés sur la couche 7 qui possède, parmi l'ensemble des couches du maillage, le plus de cellules impactées par des rais, et par conséquent conduit à des temps de traitement et une occupation mémoire plus importants. On rappelle que les meta-cellules sans aucun rai sont ignorées par l'algorithme de maillage.

La fonction utilisée pour caractériser l'illumination des meta-cellules est la *densité de rais*. Une densité de 2.5 (*ie.* $10^{2.5} \approx 316$ rais) est utilisé dans l'algorithme de maillage comme score optimal à atteindre dans chacune des meta-cellules du maillage irrégulier final de la couche 7 ([SPAKMAN & BIJWAARD 01] utilisent une densité de 2.7 à cette

³ISC : International Seismic Center.

même profondeur). De plus, un facteur de forme sur la géométrie des meta-cellules fixé à 2.5, a été utilisé.

Deux séries d'expériences ont été réalisées avec des meta-cellules ayant comme taille maximale $p_{max} \times q_{max} = 10 \times 10$ et $p_{max} \times q_{max} = 20 \times 20$.

4.3.3 Les résultats

Les résultats, permettant de comparer les performances entre les deux clusters, sont présentés dans le tableau 4.1 et les figures 4.13 pour une exécution utilisant $p_{max} \times q_{max} = 10 \times 10$, et dans le tableau 4.2 et les figures 4.14 pour $p_{max} \times q_{max} = 20 \times 20$.

np	total	configuration min – max	sélection min – max	inactivité min – max	accélération
1	211	203–203	7–7	0–0	1.0
2	150	96–143	6–53	1–49	1.4
4	121	39–116	4–82	0–80	1.7
6	106	25–101	4–79	2–76	2.0
8	97	20–93	4–77	2–77	2.2
10	81	19–76	4–61	1–60	2.6

(a) Cluster PC/Xeon 1.7Ghz.

np	total	configuration min – max	sélection min – max	inactivité min – max	accélération
1	138	133–133	5–5	0–0	1.0
2	103	65–99	4–37	1–34	1.3
4	85	24–82	3–61	1–59	1.6
8	64	12–62	2–52	1–51	2.2
16	49	4–47	1–45	0–45	2.8
32	44	2–43	1–42	0–42	3.1

(b) Cluster Itanium2/1.3Ghz.

TAB. 4.1 – Performances obtenues pour $p_{max} \times q_{max} = 10 \times 10$.

En comparant les résultats de performances globales obtenus sur les deux architectures (cf. figures 4.13(a) et 4.14(a)), on constate que la scalabilité est similaire pour un $p_{max} \times q_{max}$ donné. En effet le type de communication, constitué de nombreux messages de faible taille, ne permet pas de donner un avantage au cluster Itanium2 (réseau myrinet) par rapport au cluster Xeon (réseau gigabit).

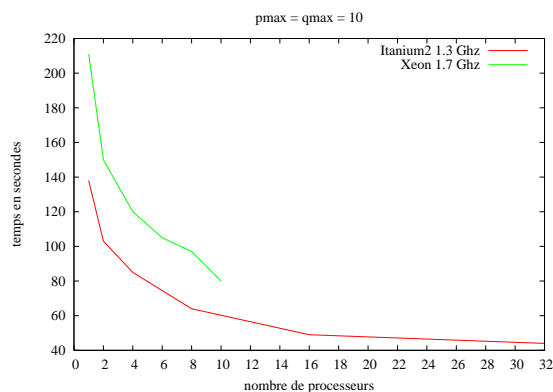
np	total	configuration min – max	sélection min – max	inactivité min – max	accélération
1	1528	1482–1482	46–46	0–0	1.0
4	995	387–955	40–607	14–587	1.5
6	920	271–881	37–647	18–632	1.7
8	851	224–819	32–627	16–613	1.8
10	773	200–740	32–571	18–558	2.0

(a) Cluster PC/Xeon 1.7Ghz.

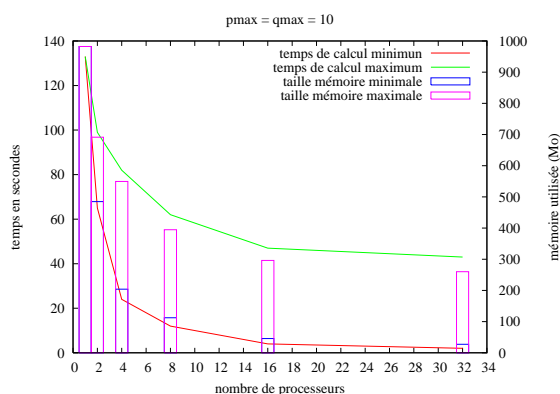
np	total	configuration min – max	sélection min – max	inactivité min – max	accélération
1	1186	1156–1156	30–30	0–0	1.0
2	885	662–857	28–223	6–205	1.3
4	778	278–755	22–500	6–487	1.5
8	627	157–609	18–470	9–464	1.9
16	565	61–552	13–504	5–500	2.1

(b) Cluster Itanium2/1.3Ghz.

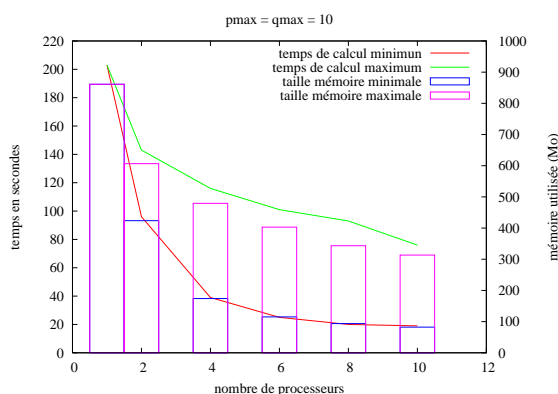
TAB. 4.2 – Performances obtenues pour $p_{max} \times q_{max} = 20 \times 20$.



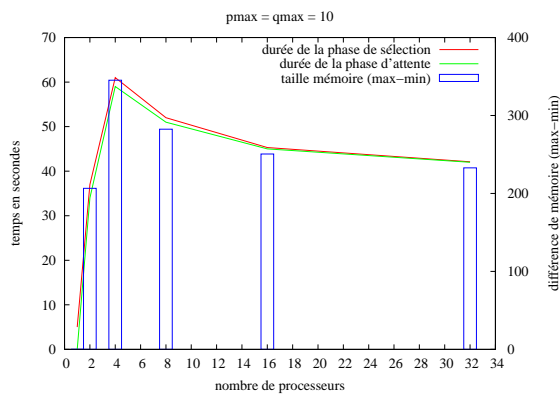
(a) Performances globales de l'application.



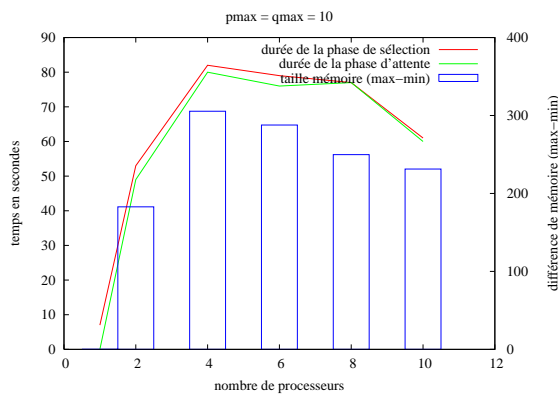
(b) Temps de calcul des configurations et occupation mémoire sur le cluster Itanium2.



(c) Temps de calcul des configurations et occupation mémoire sur le cluster Xeon.

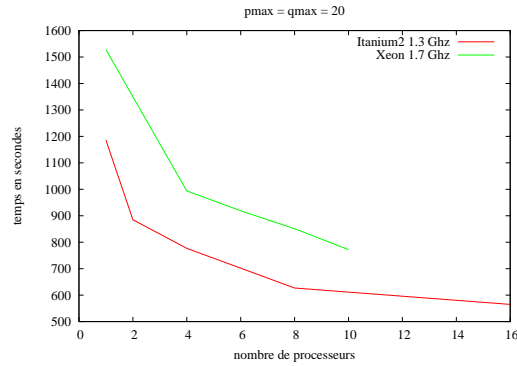


(d) Durée de la phase de sélection sur le cluster Itanium2 et différence maximale d'occupation mémoire.

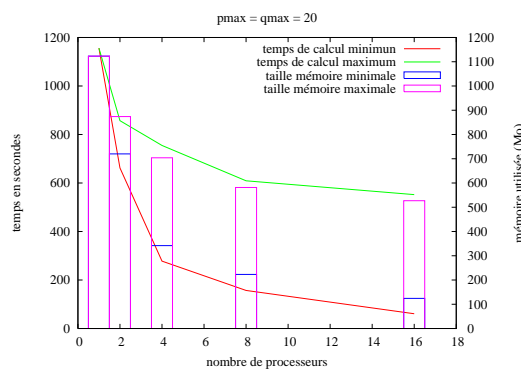


(e) Durée de la phase de sélection sur le cluster Xeon et différence maximale d'occupation mémoire.

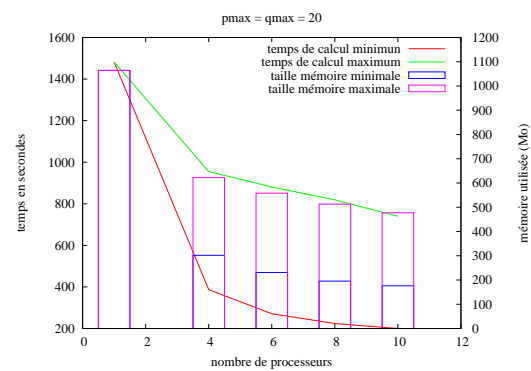
FIG. 4.13 – Performances de l'application obtenues sur le cluster Itanium2 et le cluster Xeon pour $p_{max} \times q_{max} = 10 \times 10$ en fonction du nombre de processeurs.



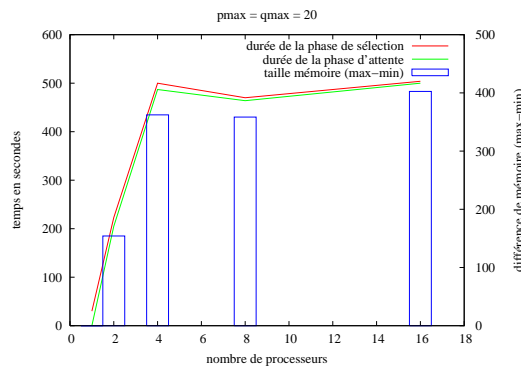
(a) Performances globales de l'application.



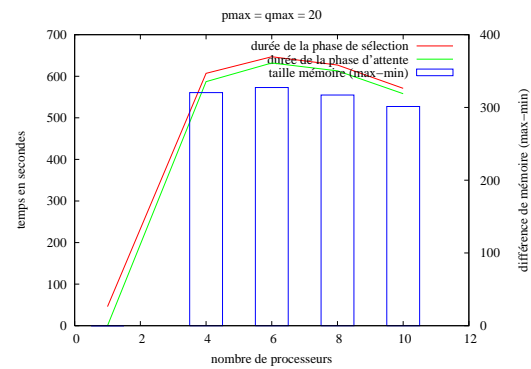
(b) Temps de calcul des configurations et occupation mémoire sur le cluster Itanium2.



(c) Temps de calcul des configurations et occupation mémoire sur le cluster Xeon.



(d) Durée de la phase de sélection sur le cluster Itanium2 et différence maximale d'occupation mémoire.



(e) Durée de la phase de sélection sur le cluster Xeon et différence maximale d'occupation mémoire.

FIG. 4.14 – Performances de l'application obtenues sur le cluster Itanium2 et le cluster Xeon pour $p_{max} \times q_{max} = 20 \times 20$ en fonction du nombre de processeurs.

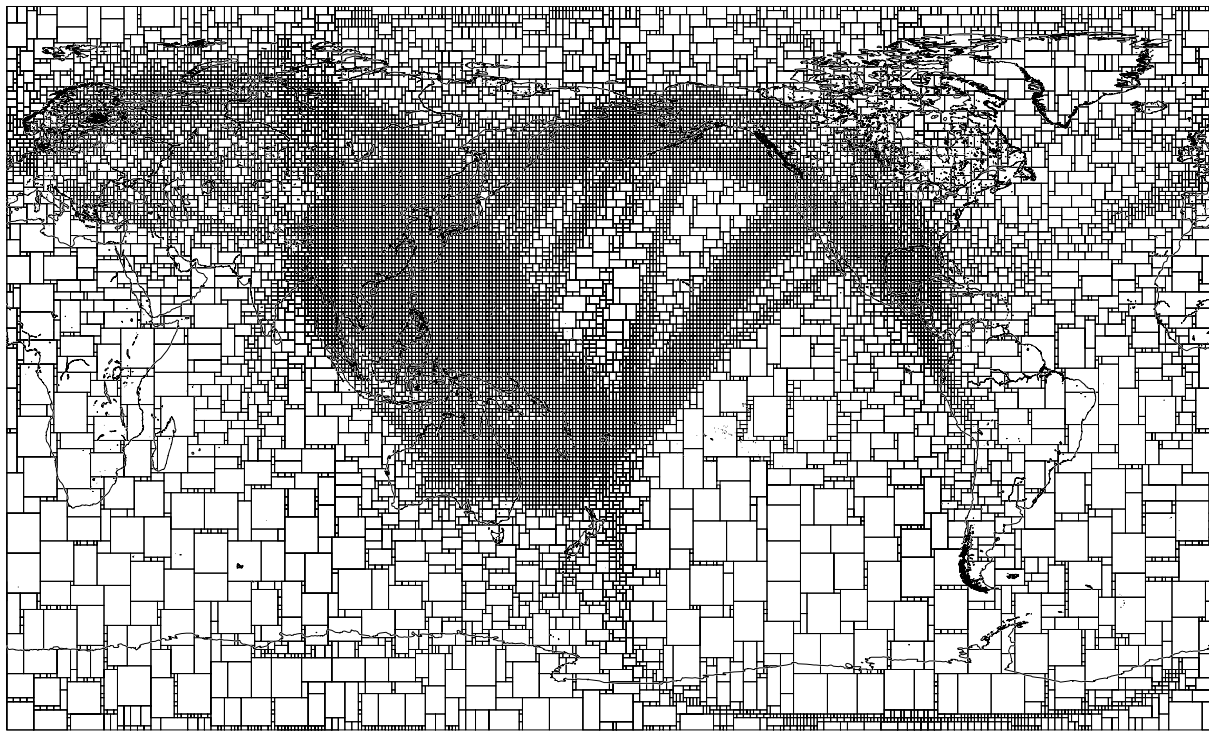
Cependant, la scalabilité de l'application est limitée par deux facteurs principaux. Tout d'abord, concernant la phase de calcul des configurations de meta-cellules, plus le nombre de processeurs augmente plus la zone de recouvrement prend relativement du poids dans le calcul par rapport au sous-domaine. Pour estimer ce surcoût relatif de la phase de calcul, lié à l'augmentation du nombre de processeurs, le tableau 4.3 montre la part du nombre de cellules constituant la zone de recouvrement ($Z_{i,j}^1 \cup Z_{i,j}^2$) du sous-domaine $D_{i,j}$ par rapport au nombre de cellules total de ce sous-domaine. Ainsi en utilisant 10 processeurs avec $p_{max} \times q_{max} = 10 \times 10$, la zone de recouvrement occupe 50% du sous-domaine traité par chaque processeur alors que pour deux processeurs elle n'occupe que 16%. D'autre part, bien que les domaines aient un nombre de cellules comparables, la taille des données associées sur les différents processeurs est très variable. Ainsi les figures 4.13(b,c) et 4.14(b,c) montrent l'occupation mémoire du processeur ayant le moins de données à traiter (mémoire minimale) et de celui en ayant le plus (mémoire maximale) parmi l'ensemble des processeurs. La durée de la phase de calcul associée à cette quantité de données minimale et maximale correspond aux temps de traitement du processeur ayant terminé le plus tôt et de celui ayant terminé le plus tard. L'écart constaté dans les durées de traitement des configurations⁴ montre que le processeur ayant terminé le plus tôt commence la phase de sélection-nettoyage en ayant de grandes chances de se trouver rapidement en attente d'un message d'un processeur étant encore dans la phase de calcul des configurations. Il en résulte que le temps d'attente maximal dans la phase de sélection est, en première approximation, bien corrélé avec les écarts en occupation mémoire (cf. figures 4.13(d,e) et 4.14(d,e)).

np	$p_{max} \times q_{max} = 10 \times 10$	$p_{max} \times q_{max} = 20 \times 20$
2	16%	28%
4	28%	44%
6	37%	54%
8	44%	62%
10	50%	67%
16	60%	80%
32	69%	86%

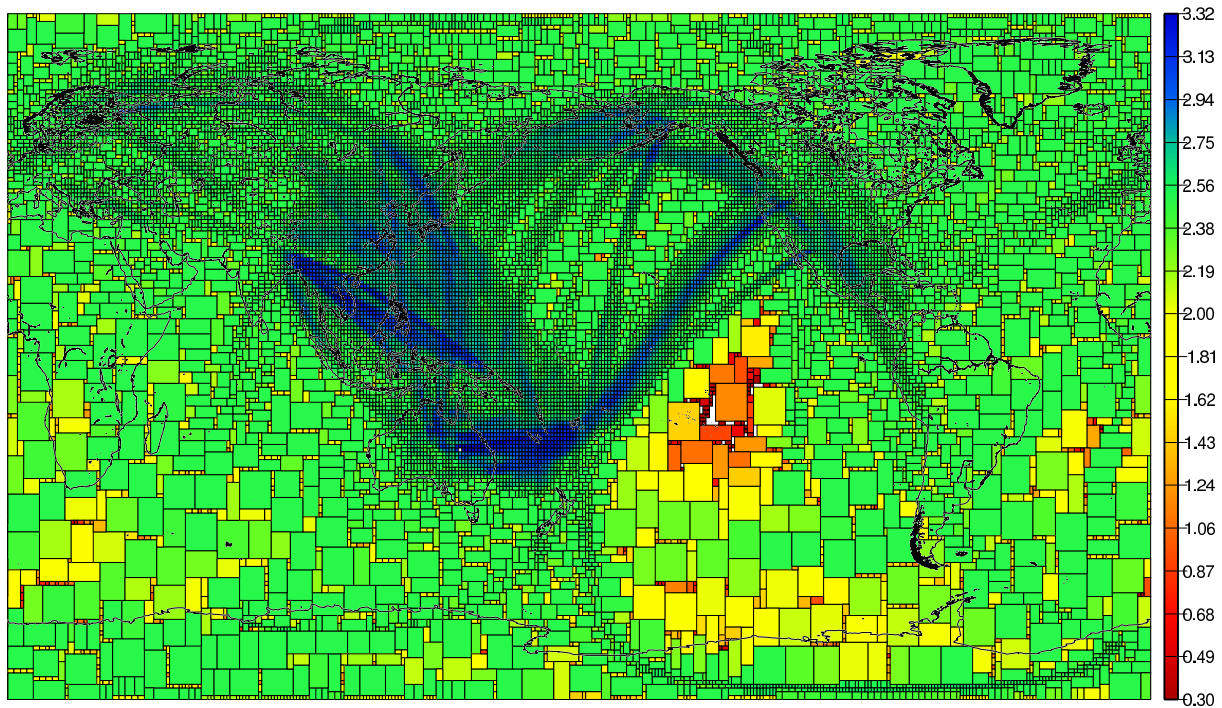
TAB. 4.3 – Pourcentage représentant la part du nombre de cellules contenues dans zone de recouvrement ($Z_{i,j}^1 \cup Z_{i,j}^2$) par rapport au nombre total de cellules du maillage $D_{i,j}$, en fonction du nombre de processeurs et de la taille maximale $p_{max} \times q_{max}$ d'une meta-cellule.

Les différentes meta-cellules formant le maillage irrégulier final sont présentées en figure 4.15(a) et les scores associés à celles-ci en figure 4.15(b) pour $p_{max} \times q_{max} = 10 \times 10$. Pour évaluer de façon *formelle* la qualité de ce maillage final, la distribution des scores est représentée en figure 4.16 sous la forme d'un histogramme (figure 4.16(a)) et d'un histo-

⁴Cet écart est amplifié par le découpage effectué dans cette expérimentation, certains processeurs étant affectés à l'hémisphère sud ou à des zones couvrant les pôles et ayant de ce fait moins de données à traiter.



(a) Tailles des meta-cellules du maillage irrégulier.



(b) Scores (densité de rais) associés aux meta-cellules du maillage irrégulier.

FIG. 4.15 – Maillage irrégulier de la couche 7 (de 760 à 2740 km) constitué de 17827 meta-cellules avec $p_{max} \times q_{max} = 10 \times 10$. La fonction de score utilisée (f) est la densité de rais, et le score optimal est 2.5 (*ie.* le nombre de rais souhaité dans les meta-cellules est $10^{2.5}$).

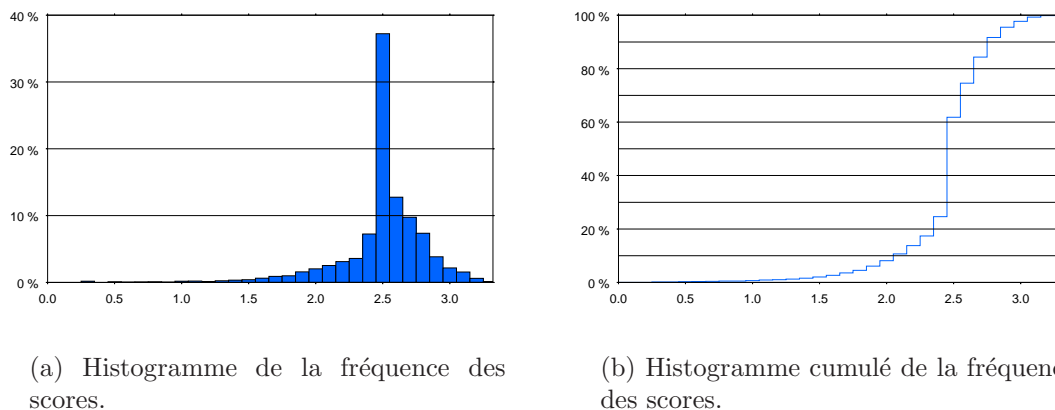


FIG. 4.16 – Distribution des scores associés aux meta-cellules du maillage final.

gramme cumulé (figure 4.16(b)). A partir de ces figures on constate une large dominance des meta-cellules ayant un score optimal fixé à 2.5 (plus de 37%). On observe également que le nombre de meta-cellules optimales (en vert sur la figure 4.15(b)) et sur-optimales (en bleu) prédomine largement, à environ 80%, sauf dans les zones peu impactées par les rais où l’optimal ne peut naturellement pas être atteint.

4.4 Conclusion

La parallélisation de l’algorithme de maillage conduit à une accélération plutôt limitée, cependant elle a l’avantage de permettre la gestion de grandes masses de données en les distribuant sur l’ensemble des processeurs. Ainsi lors d’une exécution utilisant un jeu de 350 000 rais sur un seul processeur, la taille mémoire nécessaire pour construire le maillage irrégulier d’une couche est de l’ordre du giga-octet. L’utilisation d’un jeu de données plus important, tel que celui de [ENGDAHL *et al.* 98] comportant plusieurs millions de rais, est donc tout à fait envisageable sur les deux clusters à notre disposition.

Les améliorations possibles de l’algorithme concernent essentiellement la réduction du temps d’attente des messages dans la phase de sélection-nettoyage. Un découpage du maillage initial en domaines de tailles comparables, non pas en terme de nombre de cellules mais plutôt en terme d’occupation mémoire, permettrait de réduire significativement le temps d’attente d’un message provenant d’un processeur en train de terminer sa phase de calcul de configurations. De plus, en attribuant plusieurs sous-domaines non adjacents (éventuellement sur plusieurs couches différentes) à un processeur, on minimise la probabilité que le processeur soit en attente de messages sur l’ensemble de ses sous-domaines

simultanément. Le processeur effectuerait donc la phase de sélection et de nettoyage de façon préemptive, en basculant d'un sous-domaine à l'autre après un certain nombre d'appels à la procédure `SelectClean` ou bien après un certain nombre de messages envoyés.

Chapitre 5

Résultats tomographiques

Dans cette partie nous présentons les résultats d'une tomographie globale en onde P de la croûte et du manteau. Ce choix a été fait afin de comparer notre travail à des travaux similaires que nous appelons *études de référence*. Il s'agit des publications [VAN DER HILST *et al.* 97], [BIJWAARD *et al.* 98] et [SAMBRIDGE & FALETIC 03].

Les différentes étapes, décrites dans ce chapitre, conduisent à la création d'un modèle tomographique global. Elles ont été réalisées à l'aide des différentes applications développées dans le cadre de cette thèse, plus un solveur, LSQRSOLVE (c.f. section 5.5.2), adapté pour s'intégrer dans notre suite logicielle. Au final, on peut récapituler l'enchaînement des traitements nécessaires pour la tomographie complète, à travers l'imbrication des applications, présentée en Figure 5.1.

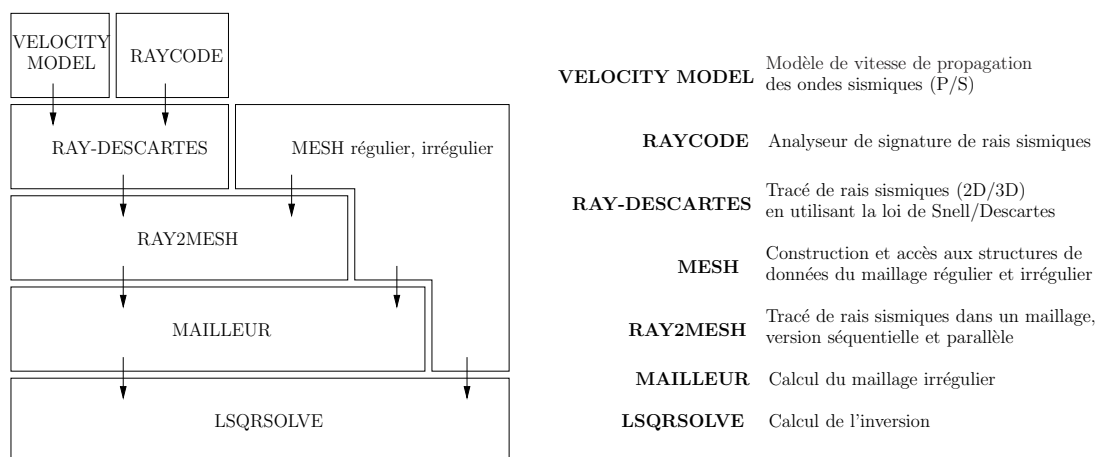


FIG. 5.1 – Organisation des briques logicielles pour la tomographie complète.

Ce chapitre débute par la définition de la zone géographique à tomographier, ainsi que par la construction du maillage initial associé. La deuxième partie expose les détails de l'extraction de notre propre jeu de rais à tracer à partir d'un jeu de données fréquemment utilisé en tomographie sismique ([ENGDAHL *et al.* 98]). Notons que ce dernier est également utilisé par les travaux préalablement cités. La troisième partie présente le tracé des rais ainsi sélectionnés dans le maillage initial. Dans la quatrième partie, grâce aux informations collectées sur la répartition géométrique de ces rais dans les différentes cellules, le maillage adaptatif est construit. Finalement, la solution du problème tomographique est calculée par inversion dans le maillage irrégulier et nous présentons les résultats du modèle de vitesse obtenu.

Comme il a été vu dans les chapitres précédents, un certain nombre de paramètres, ayant un impact sur le résultat tomographique final, doivent être définis en tenant compte des contraintes géophysiques. Cependant, le choix optimal de ces paramètres nécessiterait une étude approfondie qui dépasse le cadre de cette thèse. C'est pourquoi la plupart de ceux-ci sont directement inspirés des travaux cités précédemment.

5.1 Construction du maillage initial

La tomographie globale présentée dans ce chapitre a pour objectif d'imager la structure interne de la Terre depuis la surface jusqu'à la base du manteau inférieur située à 2889 km de profondeur. Le maillage de cette zone est celui de la figure 5.2. Il est composé de 18 couches d'épaisseurs variables, variant grossièrement de 100 km pour les couches plutôt proches de la surface, à 200 km pour les couches les plus profondes. Chaque couche est constituée de 64 800 cellules dont la taille en latitude et en longitude est de 1 degré, ce qui fait que le maillage comporte au total 1 166 400 cellules.

Dans les autres études citées ci-dessus les profondeurs des couches sont similaires. Cependant [SAMBRIDGE & FALETIC 03] maillent chaque couche par des tétraèdres, alors que [BIJWAARD *et al.* 98] utilisent des blocs réguliers dont la taille augmente avec la profondeur (de 0.6 pour la croûte à 1.8 pour le manteau inférieur) et [VAN DER HILST *et al.* 97] des blocs réguliers de 2 degrés à toutes les profondeurs.

5.2 Constitution du jeu de données à tracer

La qualité des données utilisées, pour réaliser une tomographie sismique, a une grande influence sur le résultat final. Par exemple, l'utilisation de pointés¹ peu précis, de phases

¹L'analyse d'un sismogramme où l'on détermine le temps d'arrivée d'une onde est appelée *pointé*.

```

<?xml version="1.0"?>
<mesh lat-min="-90.00" lat-max="90.00"
      lon-min="0.00" lon-max="360.00"
      lat-unit-size="1" lon-unit-size="1">

  <model>
    <layer name="layer 0" zstart="-10.00" zend="100.00" lat-unit="1" lon-unit="1" />
    <layer name="layer 1" zstart="100.00" zend="200.00" lat-unit="1" lon-unit="1" />
    <layer name="layer 2" zstart="200.00" zend="300.00" lat-unit="1" lon-unit="1" />
    <layer name="layer 3" zstart="300.00" zend="410.00" lat-unit="1" lon-unit="1" />
    <layer name="layer 4" zstart="410.00" zend="520.00" lat-unit="1" lon-unit="1" />
    <layer name="layer 5" zstart="520.00" zend="660.00" lat-unit="1" lon-unit="1" />
    <layer name="layer 6" zstart="660.00" zend="820.00" lat-unit="1" lon-unit="1" />
    <layer name="layer 7" zstart="820.00" zend="1000.00" lat-unit="1" lon-unit="1" />
    <layer name="layer 8" zstart="1000.00" zend="1200.00" lat-unit="1" lon-unit="1" />
    <layer name="layer 9" zstart="1200.00" zend="1400.00" lat-unit="1" lon-unit="1" />
    <layer name="layer 10" zstart="1400.00" zend="1600.00" lat-unit="1" lon-unit="1" />
    <layer name="layer 11" zstart="1600.00" zend="1800.00" lat-unit="1" lon-unit="1" />
    <layer name="layer 12" zstart="1800.00" zend="2000.00" lat-unit="1" lon-unit="1" />
    <layer name="layer 13" zstart="2000.00" zend="2200.00" lat-unit="1" lon-unit="1" />
    <layer name="layer 14" zstart="2200.00" zend="2400.00" lat-unit="1" lon-unit="1" />
    <layer name="layer 15" zstart="2400.00" zend="2600.00" lat-unit="1" lon-unit="1" />
    <layer name="layer 16" zstart="2600.00" zend="2750.00" lat-unit="1" lon-unit="1" />
    <layer name="layer 17" zstart="2750.00" zend="2889.00" lat-unit="1" lon-unit="1" />
  </model>
</mesh>

```

FIG. 5.2 – Description du maillage utilisé pour la tomographie qui est constitué de 18 couches de la surface jusqu’à l’interface manteau-noyau à 2889 km de profondeur.

mal identifiées, ou de localisations approximatives d’hypocentres, conduit à des résultats biaisés. C’est pourquoi de nombreuses méthodes de pré-traitements sont habituellement appliquées sur les données sismiques, de façon à réduire ou à éliminer les effets des erreurs et des incertitudes inhérentes à ce type de données.

Aussi, après avoir décrit la composition du jeu de données initial, nous décrivons l’étape de sélection des rais “à retenir” parmi ceux qui composent le jeu de données initial. Nous évoquons aussi le traitement par *ray bundles* qui permet de regrouper un ensemble de rais similaires en un unique rai “moyen”.

5.2.1 Jeu de données initial

Le jeu de données utilisé est celui de [ENGDAHL *et al.* 98] qui correspond à la sismicité mondiale de la période allant du début de l’année 1964 à la fin de l’année 1995. Il est extrait des bases de données de l’ISC² et du NEIC³. Cette sismicité a été revue et corrigée par les auteurs, en effectuant plusieurs traitements dont une re-identification des phases sismiques et une relocalisation des hypocentres. Ceci permet d’évacuer, en partie, les données de

²International Seismological Centre.

³US Geological Survey’s National Earthquake Information Center.

mauvaise qualité évoquées ci-dessus. Ce jeu de données contient environ 100 000 séismes enregistrés par plus de 7 000 stations, et totalise 11.7 millions de phases sismiques et de temps de trajet.

5.2.2 Sélection des rais

Étant donné que nous réalisons une tomographie globale à ondes P, une sélection sur les phases pertinentes doit être effectuée. Celle-ci consiste tout d'abord à éliminer du jeu de données initial les phases comportant une propagation de type S sur une partie de leur chemin, et à ne conserver que celles se propageant en compression. Cette étape réduit le jeu de données initial à 5 millions de phases de type P.

D'autres filtres, dont les paramètres proviennent de [SAMBRIDGE & FALETIC 03], sont ensuite appliqués sur ce dernier jeu de données afin d'éliminer plusieurs catégories de rais indésirables.

Les rais pour lesquels les erreurs de localisation de foyer de séisme sont significatives, ainsi que ceux qui proviennent de séismes ayant moins de 20 observations sont supprimés. Il en est de même pour ceux qui ont une incertitude sur leur temps d'arrivée supérieure à la seconde. En outre, les rais pouvant se propager dans le noyau, donc en dehors de la zone d'étude (qui s'étend de la surface à la base du manteau) sont aussi supprimés en rejetant tous ceux dont la distance angulaire entre la source et le récepteur excède une distance de 95 degrés.

5.2.3 Ray bundles

Un *ray bundle* regroupe plusieurs rais similaires, c'est-à-dire des rais qui partent et qui arrivent à peu près aux mêmes endroits (*ie.* dans un certain volume que l'on se fixe), tout en ayant des trajets très proches. Le *ray bundle* peut donc être assimilé à un rai "moyen", auquel on peut éventuellement affecter un poids dans l'inversion. Ce poids est lié à la dispersion des différents temps de propagation au sein d'un même bundle. Concrètement plus la dispersion des temps est faible plus le poids de ce bundle sera fort dans l'inversion. Dans ce travail, nous n'avons pas implémenté cette pondération. Les rays bundles ont donc tous le même poids dans l'étape d'inversion.

Les *ray bundles* sont couramment utilisés en sismologie pour réduire :

- les erreurs dûes aux incertitudes sur les localisations des hypocentres,
- l'influence des petites hétérogénéités géologiques au niveau des stations,
- ainsi que les erreurs liées à l'analyse des temps d'arrivée sur les sismogrammes.

Pour plus de précision on pourra se référer à [WIDIYANTORO & VAN DER HILST 97].

Dans cette tomographie en définissant un volume de taille $1^\circ \times 1^\circ \times 30km$, 2.6 millions de *ray bundles* ont pu être construits à partir de 5 millions de rais. Cependant pour augmenter encore la qualité de notre jeu de données, seuls les *bundles* possédant un minimum de deux rais sont conservés, ce qui revient à ignorer les rais uniques. Ainsi, le jeu de données que nous avons constitué pour cette tomographie contient 1 171 197 rais. Notons que [VAN DER HILST *et al.* 97] et [SAMBRIDGE & FALETIC 03] utilisent un volume légèrement plus grand de $1^\circ \times 1^\circ \times 50km$ ce qui conduit à un nombre de rais de l'ordre de 550 000.

5.3 Tracé de rais

Le modèle de vitesse initial, dans lequel nous traçons les rais, est le modèle radial *ak135* de [KENNETT *et al.* 95] (cf. figure 2.3). Les 1 171 197 rais ont été tracés en mode itératif (cf. chapitre 3.2.2), avec en moyenne 12 itérations par rai et un pas d'échantillonnage de 4 km, en un peu moins de 9 heures sur 4 processeurs (un mini-cluster de deux machine bi-xeon à 3.2 Ghz).

La forme générale des rais tracés est représentée dans la figure 5.3, et la répartition par type de phase est illustrée par le tableau 5.1. Ce dernier montre que plus de 90% des rais tracés correspondent à des ondes directes P, le reste étant des phases de profondeur de type pP, et enfin dans une moindre mesure des phases PP.

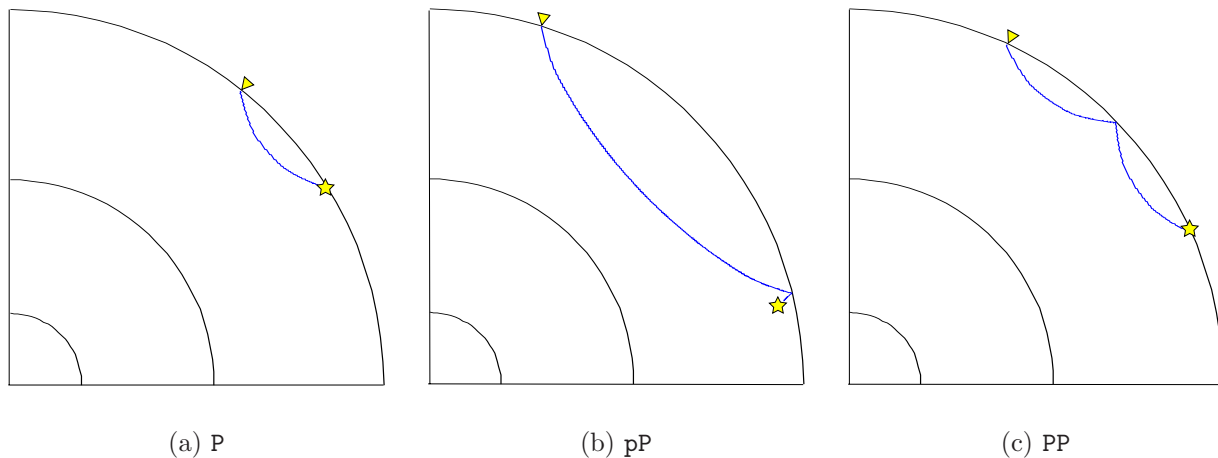


FIG. 5.3 – Exemples types de trajets en 2D des rais tracés.

phase	nb	%
P	1 095 137	93%
pP	80 809	≈ 6%
PP	271	< 1%

TAB. 5.1 – Répartition des rais tracés en fonction du type de phases

5.4 Génération du maillage irrégulier

La taille minimale d’une meta-cellule est celle d’une cellule du maillage initial, fixée à $1^\circ \times 1^\circ$, alors que sa taille maximale a été fixée arbitrairement, en latitude et en longitude, à 10 fois celle d’une cellule du maillage initial. Le facteur de forme utilisé pour contrôler la géométrie d’une meta-cellule est 2.5.

Le critère, ou score, retenu par [BIJWAARD *et al.* 98] et [SAMBRIDGE & FALETIC 03] est la *densité de rais*. A des fins de comparaison avec ces travaux nous l’avons également utilisé, et nous avons fixé comme objectif à atteindre 500 rais par meta-cellules pour les 9 couches dont la profondeur est inférieure à 1200 km et 1000 rais par meta-cellules pour les 9 suivantes (comme [BIJWAARD *et al.* 98]).

La génération du maillage irrégulier des 18 couches a été réalisée en 74 minutes, séquentiellement couche par couche, et en un peu moins de 45 minutes sur 4 processeurs. Le temps de traitement du maillage d’une couche (chargement des données et calculs) est très variable car il dépend de la quantité de rais et du nombre de cellules impactées dans celles-ci. Ainsi, le temps de traitement maximum pour mailler une couche en séquentiel est de 6 minutes pour la plus impactée (couche 7) alors que le temps de traitement minimum est de 15 secondes pour la moins impactée (couche 17, la plus profonde).

L’outil de génération du maillage irrégulier annote la description XML du maillage régulier avec des sections `data` qui désignent les fichiers de résultats de l’opération de maillage. Dans le cas de cette tomographie, le fichier annoté est⁴ celui de la figure 5.4. Les annotations de type `irm` désignent les fichiers de description du maillage irrégulier produit (un fichier par couche), et celles de type `sparse` désignent les longueurs des rais dans les meta-cellules.

Le résultat de l’opération de maillage peut être visualisé pour chaque couche. La figure 5.5 montre le maillage calculé ainsi que le nombre de meta-cellules pour chaque couche. Le maillage irrégulier global comporte 141 772 meta-cellules soit une réduction de pres de 88% du nombre de cellules du maillage initial.

⁴Pour des raisons de concision, certaines lignes répétitives ont été remplacées par des points de suspensions.

```

<?xml version="1.0"?>
<mesh lat-min="-90.00" lat-max="90.00"
      lon-min="0.00" lon-max="360.00"
      lat-unit-size="1" lon-unit-size="1">

  <model>
    <layer name="layer 0" zstart="-10.00" zend="100.00" lat-unit="1" lon-unit="1" />
    ...
    <layer name="layer 17" zstart="2750.00" zend="2889.00" lat-unit="1" lon-unit="1" />
  </model>

  <data format="irm" directory=".">
    <file name="global-00.irm"/>
    ...
    <file name="global-17.irm"/>
  </data>

  <data format="sparse" directory=".">
    <file name="global-00.sparse"/>
    ...
    <file name="global-17.sparse"/>
  </data>
  ...
</mesh>

```

FIG. 5.4 – Description XML du maillage irrégulier.

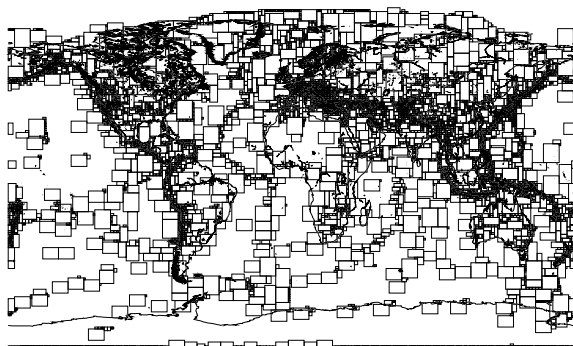
5.5 Résolution du problème inverse

5.5.1 Construction de la matrice

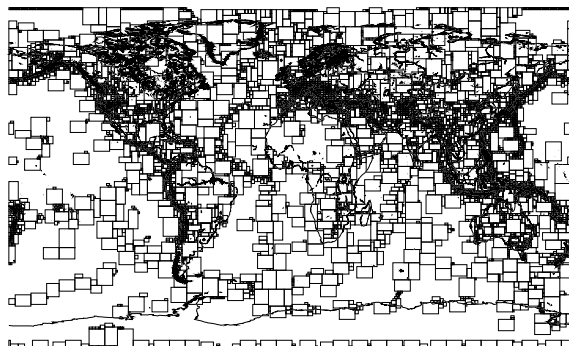
Les informations de longueurs de rais calculées dans les meta-cellules du maillage irrégulier permettent de constituer la matrice à inverser. Celle-ci est composée de 1 171 197 lignes, une pour chacun des rais tracés, et de 141 772 colonnes, une pour chaque meta-cellule traversée par au moins un rai. Cette matrice est creuse à 95%.

Comme il est précisé dans le chapitre 2.1.6, nous “régularisons” le système d’équations d’une part par le *norm damping* et d’autre part par le *gradient damping*. Cela contribue à faire converger la solution du système vers une solution géophysique. Le *gradient damping* que nous avons utilisé pour chacune des couches est une approximation de l’opérateur de dérivée seconde telle que présentée par [NOLET 87]. Le nombre de lignes de la matrice enrichie de ces nouvelles contraintes atteint alors 1 313 748.

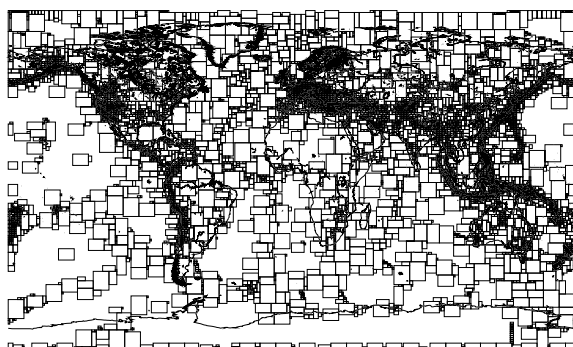
Les coefficients de *norm damping* $\lambda_1 = 4000$ et de *gradient damping* $\lambda_2 = 2000$ ont été choisis empiriquement en réalisant une dizaine de tomographies.



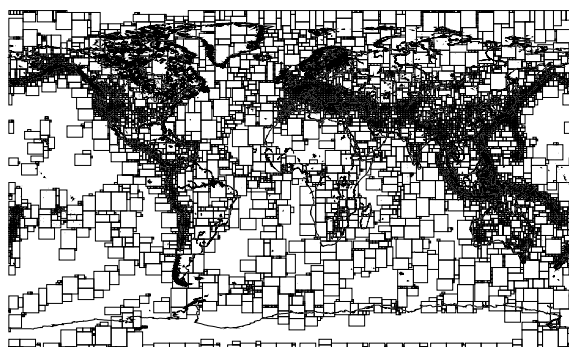
(a) 0 à 100 km, 6481 meta-cellules



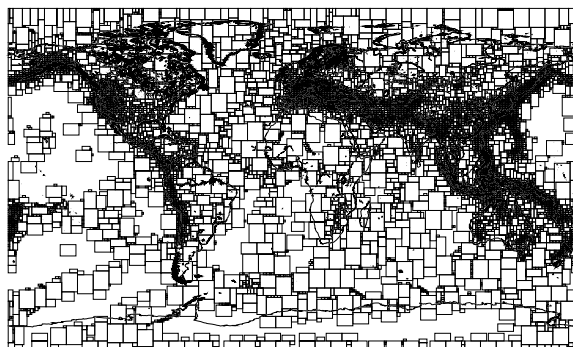
(b) 100 à 200 km, 6744 meta-cellules



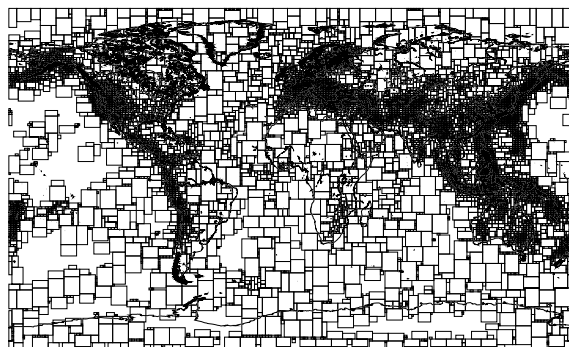
(c) 200 à 300 km, 7434 meta-cellules



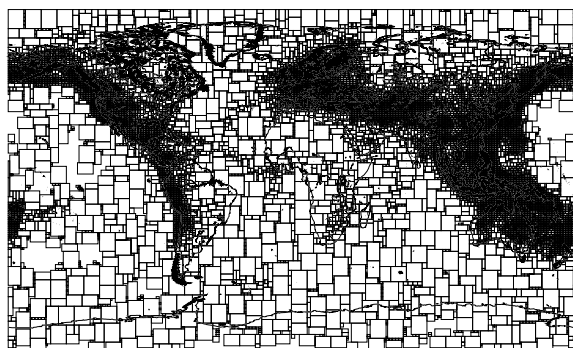
(d) 300 à 410 km, 8048 meta-cellules



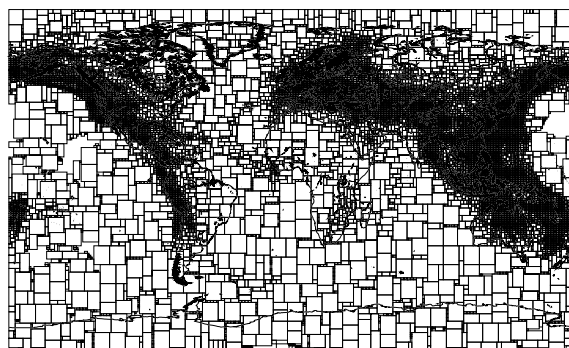
(e) 410 à 520 km, 9257 meta-cellules



(f) 520 à 660 km, 9916 meta-cellules

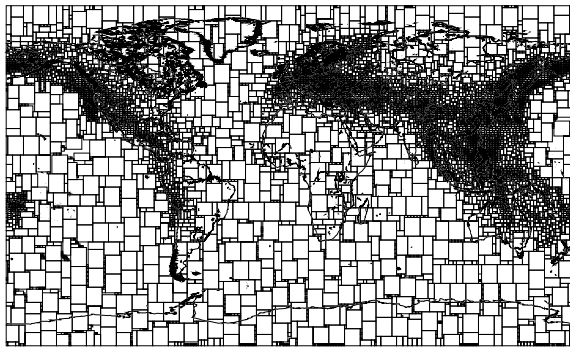


(g) 660 à 820 km, 13505 meta-cellules

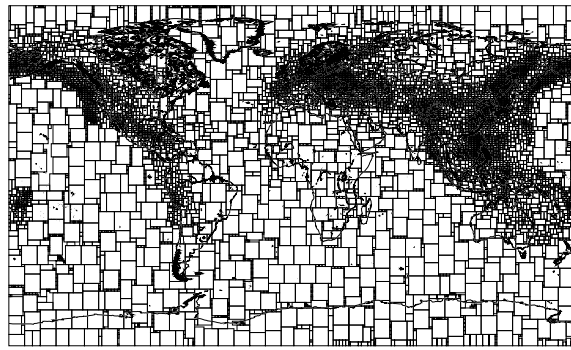


(h) 820 à 1000 km, 13891 meta-cellules

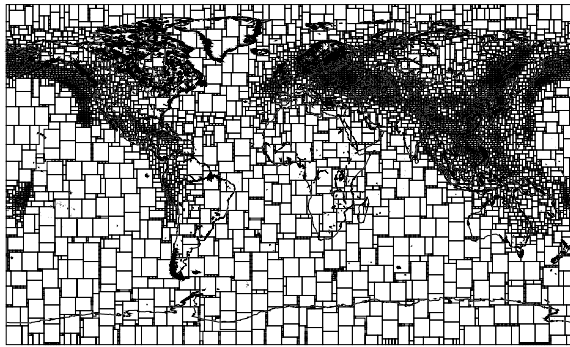
FIG. 5.5 – (Partie 1/3) – Visualisation des 18 couches du maillage irrégulier.



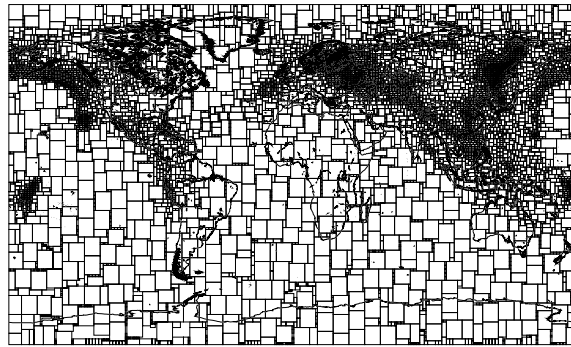
(i) 1000 à 1200 km, 8608 meta-cellules



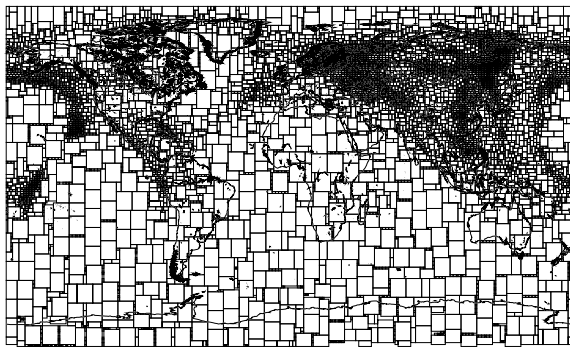
(j) 1200 à 1400 km, 8171 meta-cellules



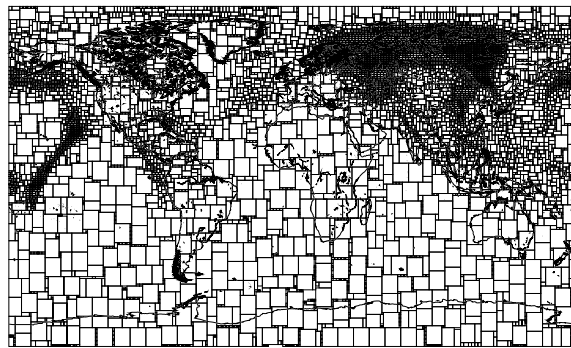
(k) 1400 à 1600 km, 7940 meta-cellules



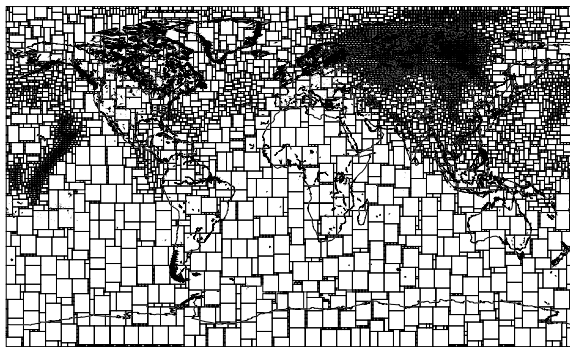
(l) 1600 à 1800 km, 7796 meta-cellules



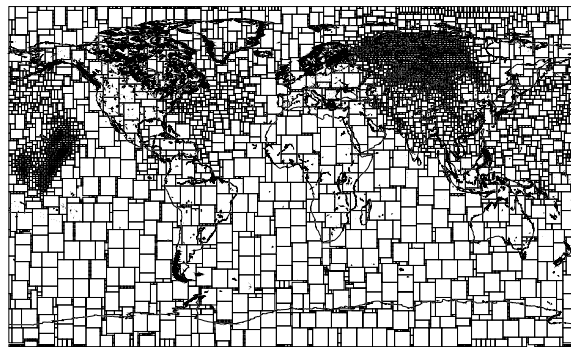
(m) 1800 à 2000 km, 7667 meta-cellules



(n) 2000 à 2200 km, 7804 meta-cellules



(o) 2200 à 2400 km, 7382 meta-cellules



(p) 2400 à 2600 km, 5309 meta-cellules

FIG. 5.5 – (Partie 2/3) – Visualisation des 18 couches du maillage irrégulier.

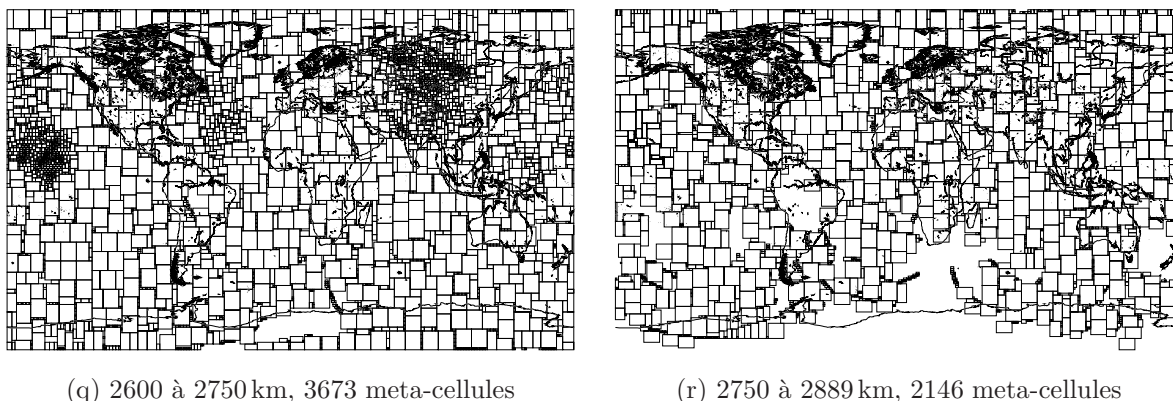


FIG. 5.5 – (Partie 3/3) – Visualisation des 18 couches du maillage irrégulier.

5.5.2 Calcul de la solution

Le calcul de la solution tomographique peut être réalisé avec l’une des méthodes décrites dans le chapitre 2.1.7. Cependant, afin de pouvoir comparer les résultats obtenus avec ceux des études de référence, notre résolution du problème inverse utilise la même méthode d’inversion, à savoir LSQR.

Dans ce travail, nous avons adapté un solveur LSQR existant [PAIGE & SAUNDERS 82]. La première adaptation du solveur consiste à fournir une implémentation optimisée d’une opération de produit “vecteur ligne” par “vecteur colonne”. De plus, le logiciel cité ci-dessus n’est qu’un moteur de résolution et ne fournit pas de primitive d’accès aux données. Nous avons donc ajouté des primitives de gestion des matrices creuses. L’utilisation de matrices creuses est très avantageuse : le stockage en creux nous coûte 20% de la taille de la matrice initiale contenant les zéros (creuse à 95%). En ne gérant pas le creux nous n’aurions pas pu résoudre de systèmes de cette taille, à la fois pour des problèmes de mémoire et de temps de calcul.

La solution du problème inverse dans cette tomographie a été calculée en 17 minutes en séquentiel sur un PC x86-64 à 3.4 Ghz ayant 8 Go de RAM, au bout d’environ 70 itérations. Le calcul de cette solution a nécessité près de 5 Go de RAM.

5.6 Résultats

Le tableau 5.6 résume les différents paramètres que nous avons utilisés pour réaliser la tomographie ainsi que ceux utilisés par les auteurs des études de référence.

	[VAN DER HILST <i>et al.</i> 97]	[BIJWAARD <i>et al.</i> 98]	[SAMBRIDGE & FALETIC 03]	dans ce travail
phases	P, pP	P, pP	P, pP	P, pP, PP
nombre de rais	550 000	inconnu	550 000	1 171 197
paramétrisation	blocs, régulier	blocs, irrégulier	tétraédrique, irrégulier	hexaédrique, irrégulier
taille du maillage initial/final	291 600/291 600	4 680 000/277 000	259 418/600 000	1 166 400/141 772
algorithme inversion	gradient conjugué	LSQR	LSQR	LSQR

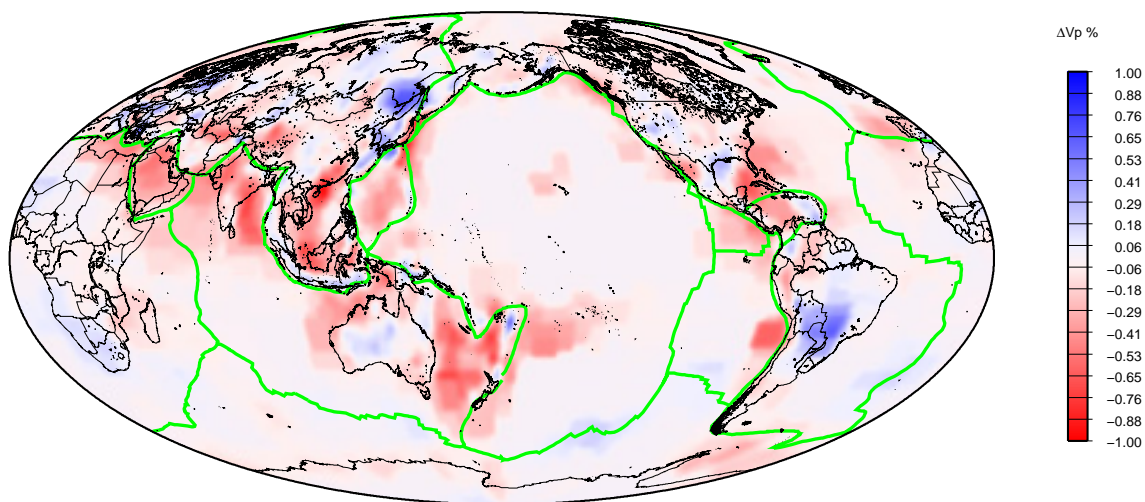
TAB. 5.2 – Tableau comparatif des paramètres de tomographie utilisés.

Les résultats tomographiques obtenus dans ce travail sont présentés dans la figure 5.6 et ceux des études de référence dans les figures 5.7, 5.8 et 5.9. Ces figures sont des cartes de pourcentages de perturbation par rapport au modèle de vitesse *ak135* (ΔV_p) à une profondeur donnée de 500 km et de 1300 km. La couleur bleue (respectivement rouge) représente les zones où les ondes sismiques se propagent plus rapidement (respectivement plus lentement) que dans le modèle de vitesse *ak135*. Notons tout de même que pour la figure 5.7, les blocs initialement de taille $2^\circ \times 2^\circ$ ont été ré-échantillonnés à posteriori par les auteurs par des blocs de $1^\circ \times 1^\circ$ pour un meilleur résultat visuel.

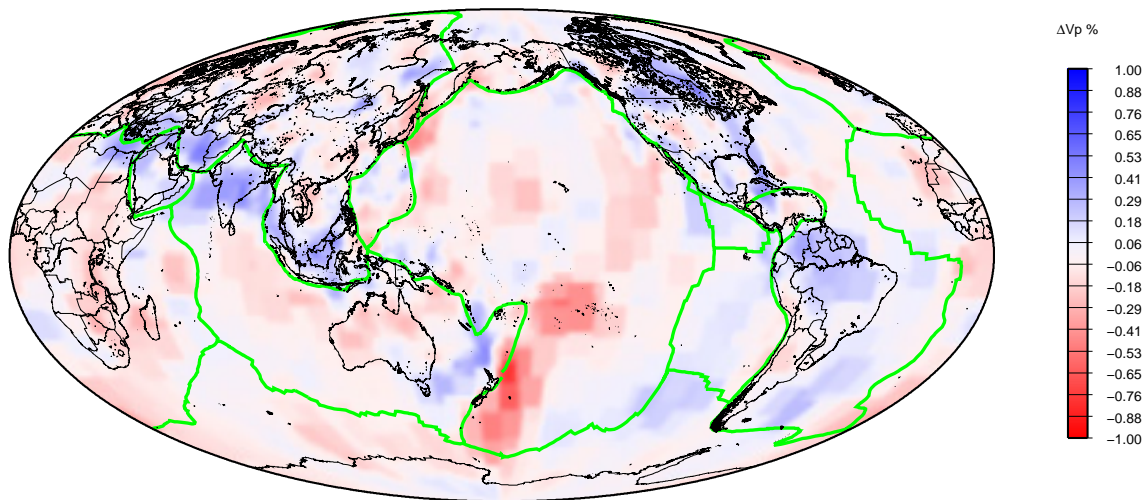
Au regard de ces figures, on constate que le modèle obtenu dans ce travail décrit les mêmes structures géologiques que les modèles des travaux de référence. Par exemple à 1300 km on distingue clairement les perturbations positives (en bleu) qui mettent en évidence l’anomalie de la Thethys [GRAND *et al.* 97] qui s’étend de l’est de la Méditerranée à Bornéo. On constate également de nombreuses autres similitudes pour des anomalies de taille plus petite. De plus, les anomalies négatives (en rouge) sont bien corrélées à cette même profondeur, avec les points chauds de la surface : les Iles des Canaries, les Iles de la Société, l’est de l’Afrique, et Yellowstone. Toutes ces observations montrent que notre tomographie est cohérente avec celles réalisées dans les études de référence.

En revanche certaines différences entre le modèle obtenu et les modèles des travaux de référence apparaissent, et peuvent s’expliquer notamment par :

- les différences de traitements sur le jeu de rais initial, et plus particulièrement dans la construction, la sélection et le nombre de rays bundles utilisés,
- une pondération identique pour tous les rays bundles dans l’inversion tomographique,
- le choix des facteurs de damping identiques pour toutes couches du maillage, alors que ceux-ci sont différents par couche dans le travail de [SPAKMAN & BIJWAARD 01].

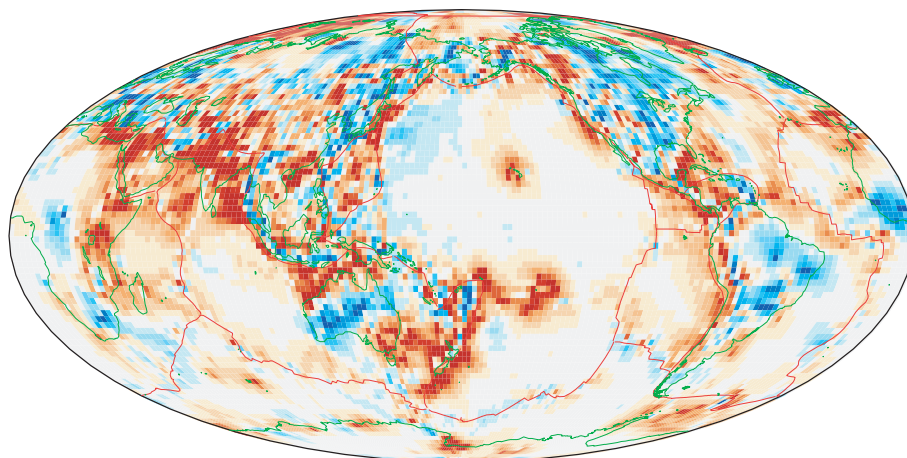


(a) 500 km

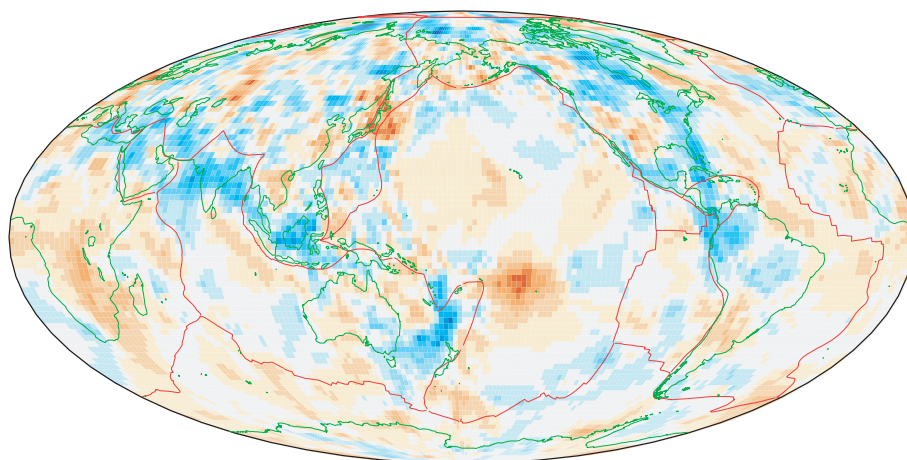


(b) 1300 km

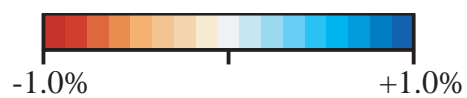
FIG. 5.6 – Résultats tomographiques obtenus dans cette thèse à 500 km et 1300 km de profondeur.



(a) 500 km



(b) 1300 km



(c) Légende

FIG. 5.7 – Résultats tomographiques obtenus par [VAN DER HILST *et al.* 97] à 500 km et 1300 km de profondeur. Pour un meilleur résultat visuel, les blocs initialement de taille $2^\circ \times 2^\circ$, ont été ré-échantillonnés à posteriori par les auteurs par des blocs de $1^\circ \times 1^\circ$.

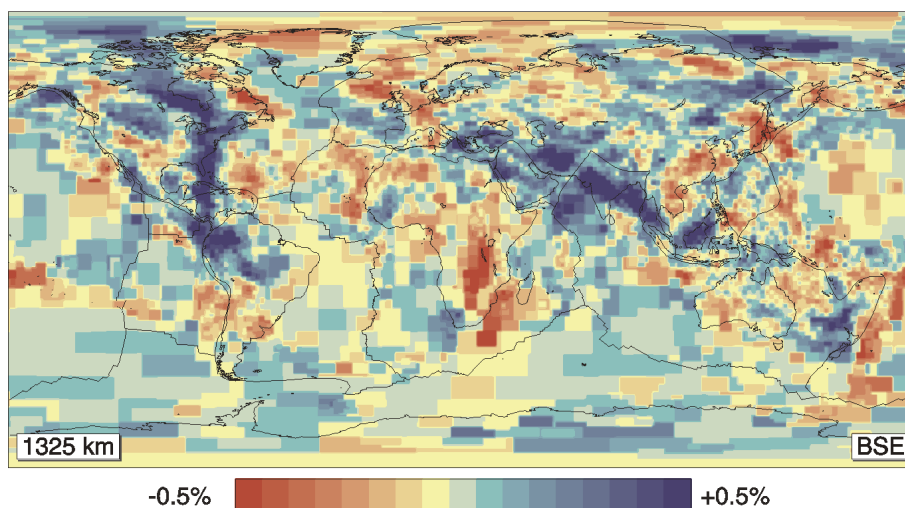


FIG. 5.8 – Résultats tomographiques obtenus par [BIJWAARD *et al.* 98] à 1325 km de profondeur.

5.6.1 Conclusion

Dans ce dernier chapitre nous avons réalisé une tomographie globale, avec une quantité de rais équivalente et une résolution comparable aux autres études publiées dans le domaine. Les résultats géophysiques obtenus sont similaires à ceux des études de référence. Ainsi, nous avons pu valider d'un point de vue géophysique, la méthode de tomographie proposée dans ce travail.

Une perspective immédiate et originale dans le domaine de la tomographie sismique, est la construction d'un nouveau modèle utilisant un maillage adaptatif basé sur un score composite, à la place du *traditionnel* score de densité de rais utilisé à la fois dans les travaux de référence et dans les expérimentations de ce travail.

Cependant, afin de pouvoir utiliser un jeu de rais encore plus important et utiliser un maillage initial plus fin que $1^\circ \times 1^\circ$ pour les couches les moins profondes, il est nécessaire de paralléliser l'étape d'inversion. On peut envisager de paralléliser LSQRSOLVE ou adapter des méthodes parallèles d'inversion existantes, telle que le logiciel libre HYPRE [FALGOUT *et al.* 06].

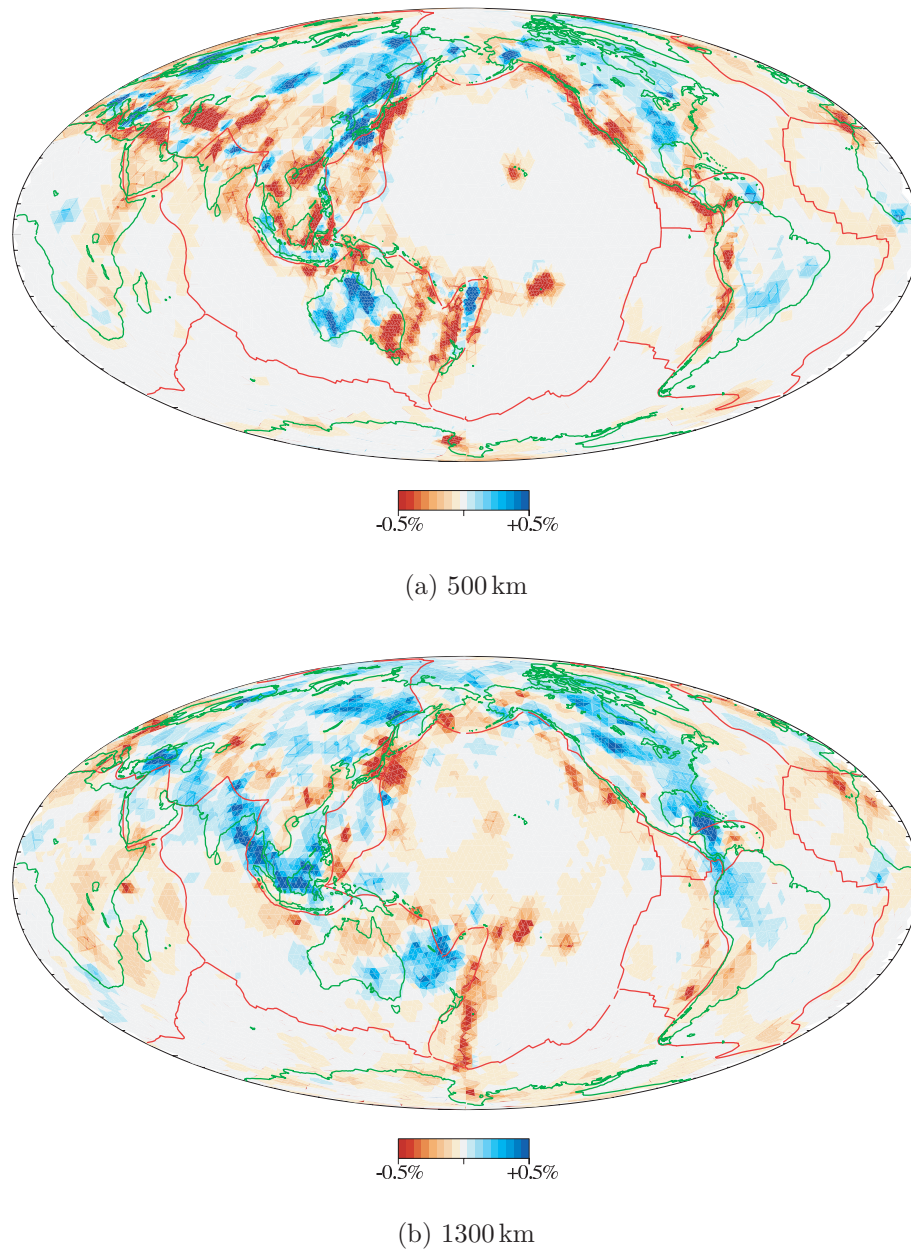


FIG. 5.9 – Résultats tomographiques obtenus par [SAMBRIDGE & FALETIC 03] à 500 km et 1300 km de profondeur.

Chapitre 6

Conclusion et perspectives

Dans cette thèse, nous avons étudié le problème de la tomographie d'un modèle global de la Terre. Ce problème est caractérisé par de grandes quantités de données – les rais sismiques – inégalement réparties dans le volume de la Terre.

La répartition inégale des informations nous a conduit à proposer une méthode de maillage adaptatif permettant d'obtenir une *bonne illumination* de l'ensemble des cellules du maillage final. Ce maillage est composé de plusieurs couches (correspondant aux différentes structures géologiques) lesquelles sont constituées de cellules hexaédriques de taille variable en fonction de la quantité d'information apportée par les rais sismiques.

La taille des données à considérer, à la fois le nombre de rais et le nombre de cellules, nécessite des ressources importantes tant en puissance de calcul qu'en mémoire. Ces contraintes nous ont naturellement conduit à proposer des méthodes parallèles d'une part pour le tracé des rais, et d'autre part pour la construction du maillage adaptatif.

L'objectif final de cette thèse était de mettre en oeuvre l'ensemble des méthodes proposées et donc de mettre à la disposition de la communauté des géophysiciens un ensemble d'outils logiciels dédiés à la tomographie sismique. Il visait également à se donner les moyens de résoudre des problèmes de taille significativement supérieure à celle des problèmes étudiés actuellement :

- en traitant plus de rais sismiques (*ie.* passer de l'analyse de quelques millions de rais à plusieurs dizaines de millions),
- en augmentant le nombre de cellules du maillage afin d'accroître la résolution du modèle tomographique.

La chaîne logicielle développée durant cette thèse a été conçue de façon modulaire et adaptable. La modularité tient dans l'architecture des différents modules pensée en termes de "briques" logicielles. Un module est composé de bibliothèques de primitives et d'utilitaires. L'intérêt de ces modules imbriqués est évident dans ce type d'application. D'une

part, plusieurs programmes peuvent utiliser des primitives provenant d'une bibliothèque de plus bas niveau (les primitives du module de gestion du maillage MESH sont par exemple utilisées à la fois par le module de tracé de rais dans un maillage RAY2MESH et par le module de maillage irrégulier MAILLEUR). D'autre part, la modularité permet d'intégrer et de tester plus facilement d'autres méthodes dans le processus de tomographie : par exemple, un autre type de tracé de rais pourrait être substitué au module RAYDESCARTES, pourvu qu'il fournisse la même interface aux modules qui l'utilisent. Nous souhaitons ainsi que notre suite logicielle puisse accueillir des modules écrits par d'autres équipes, ou être réutilisée entièrement ou en partie. Pour favoriser cela, nous diffusons ces logiciels sous licence libre.

Les logiciels développés ont été conçus pour s'adapter à différentes architectures, et en priorité aux architectures parallèles. En effet, l'objectif de réaliser une tomographie en utilisant de très grandes quantités de données nous imposait de développer des algorithmes parallèles. Nous avons parallélisé les deux applications principales, à savoir le tracé de rais dans un maillage et la génération du maillage adaptatif. Dans le cas du tracé de rais, le problème autorise une parallélisation qui a une bonne scalabilité, et nous obtenons des gains de temps très importants. Dans le cas du maillage adaptatif, la scalabilité est médiocre mais l'emploi du parallélisme permet de s'affranchir de certaines limites liées à la mémoire. Nous avons analysé le comportement de ces applications dans leurs versions parallèles dans différents contextes comme sur une machine parallèle, des clusters d'entrée de gamme ou de haut de gamme, ou encore sur une grille de calcul. Outre la démonstration de la portabilité des applications, ces expériences permettent de mieux comprendre quel type d'application parallèle se prête à quel type d'architecture matérielle. Enfin, l'adaptabilité de notre solution permet d'exécuter les mêmes applications en séquentiel, y compris sur des données volumineuses grâce à des techniques de type *out of core*.

Afin de valider, d'un point de vue géophysique, la méthode de tomographie proposée dans ce travail, nous avons réalisé une tomographie globale, avec une quantité de rais similaire et une résolution comparable aux autres études publiées dans le domaine. La tomographie a pu être calculée en quelques heures, et les résultats géophysiques sont similaires à ceux des études de référence.

La suite logique de ce travail est de conduire, en étroite collaboration avec les géophysiciens, des expérimentations. Elles permettront d'étudier les différents paramètres qui influent sur le résultat tomographique et ainsi en améliorer la qualité. Parmi les points qui influencent la tomographie, on peut citer :

- le choix des facteurs de damping, éventuellement différents par couche, comme dans le travail [SPAKMAN & BIJWAARD 01],
- les interactions entre la forme ou le voisinage des cellules du maillage et le damping,
- la définition de nouveaux scores pertinents, qui peuvent être des scores composites (en utilisant l'ensemble des données de la cellule),

- la détermination de “meilleures” tailles de cellule pour le maillage initial (selon les couches) ainsi que les tailles maximales des meta-cellules et leur rapport de forme.

Hormis ce travail avec les sismologues, il reste dans la chaîne de traitements un maillon séquentiel qui est la résolution du problème inverse. Deux alternatives nous semblent possibles à ce niveau : paralléliser LSQRSOLVE ou adapter des méthodes parallèles d’inversion existantes, telle que le logiciel libre HYPRE [FALGOUT *et al.* 06].

Finalement, ce travail inter-disciplinaire a permis de concevoir de nouvelles méthodes pour la tomographie sismique et de mettre à disposition des sismologues une suite logicielle permettant de réaliser des tomographies sur un vaste éventail d’architectures. La suite logicielle est en effet utilisable sur une simple station de travail, mais tirera avantage d’architectures parallèles. Dans ce cas, des tomographies globales de plusieurs dizaines de millions de rais avec une résolution basée sur des cellules de moins d’un degré de côté peuvent être réalisées en quelques heures. C’est donc une voie ouverte à plus d’expérimentations et à plus d’études dans le domaine de la tomographie sismique.

Bibliographie

- [DAS] {[Http://www.cs.vu.nl/das/](http://www.cs.vu.nl/das/)}
- [TER] {[Http://www.teragrid.org/](http://www.teragrid.org/)}
- [CST 90] *C* Programming Guide*, novembre 1990.
- [HPF 93] HPF Forum, *High Performance Fortran Language Specification*, 1^{re} édition, May 3 1993.
- [MPI 95] Message Passing Interface Forum, *MPI : A message-passing Interface Standard*, June 1995.
- [COR 03] « Common object request broker architecture (CORBA/IIOP) », Rapport technique, OMG, novembre 2003. {Technical report - document formal/02-11-03}
- [BIJWAARD *et al.* 98] H. BIJWAARD, W. SPAKMAN & E. R. ENGDAHL, « Closing the gap between regional and global travel time tomography », *Journal of Geophysical Research*, vol. 103, n° B12, 30 055–30 078, December 1998.
- [BODEN *et al.* 95] N. J. BODEN, D. COHEN, R. E. FELDERMAN, A. E. KULAWIK, C. L. SEITZ, J. N. SEIZOVIC & W.-K. SU, « Myrinet : A Gigabit-per-Second Local Area Network », *IEEE Micro*, vol. 15, n° 1, 29–36, 1995, URL citeseer.ist.psu.edu/boden95myrinet.html.
- [BOSCHI 03] L. BOSCHI, « Measures of resolution in global body wave tomography », *Geophysical Research Letters*, vol. 30, n° 19, 2003.
- [BOUTEILLER *et al.* 03] A. BOUTEILLER, F. CAPPELLO, T. HÉRAULT, G. KRAWEZIK, P. LEMARINIER & F. MAGNIETTE, « MPIch-V2 : a Fault Tolerant MPI for Volatile Nodes based on the Pessimistic Sender Based Message Logging », in *SuperComputing 2003*, Phoenix USA, novembre 2003.
- [CAPPELLO 05] F. CAPPELLO, « Grid'5000 : A Large Scale, Reconfigurable, Controlable and Monitorable Grid Platform », in *Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing Grid'2005*, novembre 2005, URL <http://grid5000.org>.
- [CAREY 97] G. F. CAREY, *Computational Grids : Generations, Adaptation and Solution Strategies*, Taylor & Francis, Washington, DC., 1mai 1997.
- [CARON *et al.* 02] E. CARON, F. DEPREZ, F. FRÉDÉRIC LOMBARD, J.-M. NICOD, M. QUINSON & F. SUTER, « A Scalable Approach to Network Enabled Servers », in *8th EuroPar Conference*, vol. 2400 de *LNCS*, pages 907–910. Springer-Verlag, août 2002.

- [CERVENY *et al.* 77] V. CERVENY, I. A. MOLOTKOV & I. PSENCIK., *Ray method in seismology*, University of Karlova Press, Prague, 1977.
- [CHRISOCHOIDES 05] N. CHRISOCHOIDES, *Parallel Mesh Generation, In Numerical Solution of Partial Differential Equations on Parallel Computers*, vol. 51 de *Lecture Notes in Computational Science and Engineering*, Springer-Verlag, 2005.
- [COUGNY & SHEPHARD 99] H. COUGNY & M. SHEPHARD, « Parallel volume meshing using face removals and hierarchical repartitioning », *Comp. Meth. Appl. Mech. Engng.*, vol. 174, n° 3-4, 275–298, 1999.
- [DAGUMI & MENON 98] L. DAGUMI & R. MENON, « "OpenMP : An Industry-Standard API for Shared-Memory Programming" », *IEEE Computational Science and Engineering*, vol. 5, n° 1, 46–55, 1998.
- [DESCARTES 37] R. DESCARTES, « La dioptrique », 1637.
- [DIJKSTRA 59] E. W. DIJKSTRA, « A note on two problems in connection with graphs », *Numer. Math.*, n° 1, 269–271, 1959.
- [DZIEWONSKI & ANDERSON 81] A. M. DZIEWONSKI & D. L. ANDERSON, « Preliminary reference Earth model », *Physics of The Earth and Planetary Interiors*, vol. 25, n° 4, 297–356, juin 1981.
- [DZIEWONSKY 84] A. M. DZIEWONSKY, « Mapping the lower mantle : determination of lateral heterogeneity in P velocity up to degree and order 6 », *Journal of Geophysical Research*, vol. 89, n° B7, 5929–5952, 10juillet 1984.
- [EL-GHAZAWI *et al.* 05] T. EL-GHAZAWI, W. CARLSON, T. STERLING & K. YELICK, *UPC : Distributed Shared Memory Programming*, Wiley, July 2005.
- [ENGDAHL *et al.* 98] E. R. ENGDAHL, R. D. V. DER HILST & R. P. BULAND, « Global teleseismic earthquake relocation with improved travel times and procedures for depth determination », *Bulletin of Seismol. Society Amer.*, n° 88, pp-722–743, 1998.
- [FALGOUT *et al.* 06] R. FALGOUT, J. JONES & U. YANG, *The Design and Implementation of hypre, a Library of Parallel High Performance Preconditioners*, vol. 51 de *Lecture Notes in Computational Science and Engineering*, pages 267–294, Springer-Verlag, 2006. {Aussi disponible en rapport LLNL Technical Report UCRL-JRNL-205459, 2004.}
- [FEDAK *et al.* 01] G. FEDAK, C. GERMAIN, V. NÉRI & F. CAPPELLO, « XtremWeb : A Generic Global Computing System », in *CCGRID*, pages 582–587. IEEE Computer Society, 2001.
- [FERMAT 91] P. d. FERMAT, « Oeuvre de Fermat », *Paris*, vol. 2, 354, 1891.
- [FLYNN 66] M. J. FLYNN, « Very high-speed computing systems », *Proceedings of the IEEE*, vol. 54, 1901–1909, décembre 1966.
- [FOSTER & KARONIS 98] I. FOSTER & N. T. KARONIS, « A Grid-Enabled MPI : Message Passing in Heterogeneous Distributed Computing Systems », in *Proc. of the 1998 ACM/IEEE conference on Supercomputing*, Nov. 1998, URL http://www.supercomp.org/sc98/TechPapers/sc98_FullAbstracts/Foster1125/index.htm.

- [FOSTER & KESSELMAN 97] I. FOSTER & C. KESSELMAN, « Globus : A Metacomputing Infrastructure Toolkit », *The International Journal of Supercomputer Applications and High Performance Computing*, vol. 11, n° 2, 115–128, 1997.
- [FOSTER & KESSELMAN 98] I. FOSTER & C. KESSELMAN, *The Grid, Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers, Inc., 1998.
- [FREEMAN *et al.* 99] E. FREEMAN, S. HUPFER & K. ARNOLD, *JavaSpaces Principles, Patterns, and Practice*, Pearson Education, June 1999.
- [FREY & GEORGE 99] P. J. FREY & P.-L. GEORGE, *Maillages : applications aux éléments fins*, Hermès Sciences Publications, Paris, 1999.
- [GELERNTER 85] D. GELERNTER, « Generative communications in Linda », *ACM Transactions on Programming Languages and Systems*, vol. 7, 80–112, January 1985.
- [GENAUD & RATTANAPOKA 05] S. GENAUD & C. RATTANAPOKA, « A Peer-to-Peer Framework for Robust Execution of Message Passing Parallel Programs », in B. D. M. ET AL. (dir.), *EuroPVM/MPI 2005*, vol. 3666 de *LNCS*, pages 276–284. Springer-Verlag, September 2005.
- [GEORGE & BOROUCAKI 98] P.-L. GEORGE & H. BOROUCAKI, *Triangulation de Delaunay et maillage - application aux éléments finis*, Hermès, 1998.
- [GILBERT 72] P. GILBERT, « Iterative methods for the three-dimensional reconstruction of an object from projections », *J. Theor. Biol.*, n° 36, 105–117, 1972.
- [GORDON *et al.* 70] R. GORDON, R. BENDER & G. T. HERMAN, « Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and x-ray photography », *J. Theor. Biol.*, n° 29, 471–481, 1970.
- [GRAND *et al.* 97] S. P. GRAND, R. D. VAN DER HILST & S. WIDIYANTORO, « Global seismic tomography : A snapshot of convection in the Earth », *GSA Today*, vol. 7, 1–7, 1997.
- [GRANET & TRAMPERT 89] M. GRANET & J. TRAMPERT, « Large-scale P-velocity structures in the Euro-Mediterranean area », *Geophys. J. Int.*, vol. 99, 583–594, 1989.
- [GROPP *et al.* 96] W. GROPP, E. LUSK, N. DOSS & A. SKJELLUM, « A high-performance, portable implementation of the MPI message passing interface standard », *Parallel Computing*, vol. 22, n° 6, 789–828, septembre 1996.
- [HESTENES 80] M. R. HESTENES, *Conjugate directions methods in optimization*, Springer, 1980.
- [HOARE 78] C. A. R. HOARE, « Communicating sequential processes », *Communications of the ACM*, vol. 21, n° 8, 1978.
- [HOARE 85] C. A. R. HOARE, *Communicating Sequential Processes*, Prentice-Hall, 1985.
- [HOEFLINGER 06] J. P. HOEFLINGER, « Extending OpenMP* to Clusters », 2006. {White Paper}
- [HUYGENS 90] C. HUYGENS, « Traité de la lumière », *Leyden*, 1690.

- [KENNETT *et al.* 95] B. L. KENNETT, E. R. ENGDAHL & R. BULAND, « Constraints on seismic velocities in the Earth from traveltimes », *Geophys. J. Int.*, vol. 122, 108–124, 1995.
- [KOELBEL *et al.* 94] C. H. KOELBEL, D. B. LOVEMAN, R. S. SCHREIBER, G. L. STEELE & M. E. ZOSEL, *The High Performance Fortran Handbook*, The MIT Press, 1994.
- [MARQUES *et al.* 03] O. MARQUES, T. DRUMMOND & D. VASCO, « A Computational Strategy for the Solution of Large Linear Inverse Problems in Geophysics », in *International Parallel and Distributed Processing Symposium (IPDPS'03)*, page 16b, 2003.
- [MOSER 91] T. J. MOSER, « Shortest path calculation of seismic rays », *Geophysics*, vol. 56, n° 1, 59–67, 1991.
- [NAKADA *et al.* 02] H. NAKADA, S. MATSUOKA, K. SEYMOUR, J. DONGARRA, C. LEE & H. CASANOVA, « GridRPC : A Remote Procedure Call API for Grid Computing », in *Proceedings of Grid'02*, LNCS. Springer, novembre 2002.
- [NAKANISHI & YAMAGUCHI 86] I. NAKANISHI & K. YAMAGUCHI, « A numerical experiment on non linear image reconstruction from first-arrival times for two-dimensional island arc structure », *J. Phys. Earth*, n° 34, 195–201, 1986.
- [NELDER & MEAD 65] A. NELDER & R. MEAD, « A simplex method for function minimization. », *Comput. J.*, n° 7, 308–313, 1965.
- [NOLET 87] G. NOLET, *Seismic Tomography : With Application in Global Seismology and Exploration Geophysics*, chap. Seismic wave propagation and seismic tomography, pages 1–23, D. Reidel, Norwell, Mass., 1987.
- [OLIKER & BISWAS 98] L. OLIKER & R. BISWAS, « Plum : Parallel load balancing for adaptive unstructured meshes », *Journal of Par. and Dist. Comp.*, vol. 52, n° 2, 150–177, 1998.
- [PAIGE & SAUNDERS 82] C. PAIGE & M. SAUNDERS, « LSQR : An algorithm for sparse linear equations and sparse least squares », *ACM Transactions on Mathematical Software*, vol. 8, 43–71, mars 1982.
- [PÉREZ *et al.* 03] C. PÉREZ, T. PRIOL & A. RIBES, « A Parallel CORBA Component Model for Numerical Code Coupling », *The International Journal of High Performance Computing Applications (IJHPCA)*, vol. 17, n° 4, 417–429, 2003.
- [PERRONNET 98] A. PERRONNET, « Interpolation transfinie sur le triangle, le tétraèdre et le pentaèdre. Application à la création de maillage et à la condition de Dirichlet. », *C. R. Acad. Sci.*, vol. t 326, n° Serie I, 117–122, 1998.
- [ROWBOTHAM PETER S. & PRATT R. GERHARD 97] ROWBOTHAM PETER S. & PRATT R. GERHARD, « Improved inversion through use of the null space », *Geophysics*, vol. 62, n° 3, 869–883, mai 1997.
- [SAMBRIDGE & FALETIC 03] M. SAMBRIDGE & R. FALETIC, « Adaptive whole Earth tomography », *Geochem., Geophys., Geosyst.*, vol. 4, n° 3, March 2003.
- [SCHROEDER *et al.* 97] W. SCHROEDER, K. MARTIN & B. LORENSEN, *The Visualization Toolkit, An Object-Oriented Approach To 3D Graphics*, Prentice Hall, December 1997.

- [SPAKMAN & BIJWAARD 01] W. SPAKMAN & H. BIJWAARD, « Optimization of Cell Parametrizations for Tomographic Inverse Problems », *Pure and Applied Geophysics*, n° 158, 1401–1423, 2001.
- [SQUYRES *et al.* 00] J. M. SQUYRES, A. LUMSDAINE, W. L. GEORGE, J. G. HAGEDORN & J. E. DEVANEY, « The Interoperable Message Passing Interface (IMPI) Extensions to LAM/MPI », in *Proceedings of MPIDC'2000*, March 2000.
- [STORCHAK *et al.* 03] D. A. STORCHAK, J. SCHWEITZER & P. BORMANN, « The IASPEI Standard Seismic Phase List », *Seismol. Res. Lett.*, vol. 74, n° 6, 761–772, 2003.
- [SUNDERAM 90] V. S. SUNDERAM, « PVM : A Framework for Parallel Distributed Computing », *Concurrency, Practice and Experience*, vol. 2, n° 4, 315–339, décembre 1990.
- [THOMPSON *et al.* 85] J. F. THOMPSON, Z. U. A. WARSI & C. W. MASTIN, *Numerical grids generation, foundations and applications*, North Holland, 1985.
- [TKALCIC *et al.* 02] H. TKALCIC, B. ROMANOWICZ & N. HOUY, « Constraints on D' structure using PKP(AB-DF), PKP(BC-DF) and PcP-P travelttime data from broadband records », *Geophys. J. Int.*, vol. 148, 599–616, 2002.
- [TREFETHEN & BAU 97] N. TREFETHEN & D. BAU, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
- [VAN DER HILST *et al.* 97] R. D. VAN DER HILST, S. WIDIYANTORO & E. R. ENGD AHL, « Evidence for deep mantle circulation from global tomography », *Nature*, vol. 386, 578–584, 1997.
- [VIDALE 90] J. E. VIDALE, « Finite-difference calculation of traveltimes in three dimensions », *geophysics*, vol. 55, n° 5, 521–526, mai 1990.
- [W. A. PROTHERO *et al.* 88] W. A. PROTHERO, W. J. TAYLOR & J. A. EICKEMEYER, « A fast, two-point, three-dimensional raytracing algorithm using a simple step search method », *Bulletin of the Seismological Society of America*, vol. 78, n° 3, 1190–1198, juin 1988.
- [WIDIYANTORO & VAN DER HILST 97] S. WIDIYANTORO & R. D. VAN DER HILST, « Mantle structure beneath Indonesia inferred from high-resolution tomographic imaging », *Geophys. J. Int.*, vol. 130, 167–182, 1997.
- [YAMAMURO 74] S. YAMAMURO, *Differential calculus in topological linear spaces*, vol. 374, Springer-Verlag, Berlin-New York, 1974.
- [ZIV & LEMPEL 77] J. ZIV & A. LEMPEL, « A Universal Algorithm for Sequential Data Compression », *IEEE Transactions on Information Theory*, vol. 23, n° 3, 337–343, 1977, URL citeseer.ist.psu.edu/ziv77universal.html.

Annexe A

Nomenclature des phases sismiques

IASPEI Standard Seismic Phase List

The new list modifies and complements the phase nomenclature, published in the previous edition of the Manual of Seismological Observatory Practice (Willmore,1979) and the ISC bulletins. It is more in tune with the modern earth models and accounts for the significantly increased detection capabilities of modern (digital) seismic stations. The new phase list also uses the larger flexibility of the new data exchange format (ISF).

After numerous consultations with the seismological community this list has been adopted by the IASPEI Commission on Seismological Observation and Interpretation and approved by the IASPEI Assembly at its meeting in Sapporo on July 04, 2003.

The paper explaining the list and presenting ray path diagrams of major phases was published by Storchak, D.A., J. Schweitzer, P. Bormann (2003). The IASPEI Standard Seismic Phase List, Seismol. Res. Lett. 74, 6, 761-772.

IASPEI Working Group members

Adams R.D.
Bormann P.
Engdahl R.E.
Havskov J.
Kennett B. L. N.
Schweitzer J.
Storchak D.A. (Chairman)

Notations used:

alt - acceptable alternative name;
old - name used in the past, no longer acceptable.

CRUSTAL PHASES

- Pg At short distances, either an upgoing P wave from a source in the upper crust or a P wave bottoming in the upper crust. At larger distances also arrivals caused by multiple P-wave reverberations inside the whole crust with a group velocity around 5.8 km/s.
- Pb (alt:P*) Either an upgoing P wave from a source in the lower crust or a P wave bottoming in the lower crust
- Pn Any P wave bottoming in the uppermost mantle or an upgoing P wave from a source in the uppermost mantle

PnPn	Pn free surface reflection
PgPg	Pg free surface reflection
PmP	P reflection from the outer side of the Moho
PmPN	PmP multiple free surface reflection; N is a positive integer. For example, PmP2 is PmPPmP
PmS	P to S reflection from the outer side of the Moho
Sg	At short distances, either an upgoing S wave from a source in the upper crust or an S wave bottoming in the upper crust. At larger distances also arrivals caused by superposition of multiple S-wave reverberations and SV to P and/or P to SV conversions inside the whole crust.
Sb	(alt:S*) Either an upgoing S wave from a source in the lower crust or an S wave bottoming in the lower crust
Sn	Any S wave bottoming in the uppermost mantle or an upgoing S wave from a source in the uppermost mantle
SnSn	Sn free surface reflection
SgSg	Sg free surface reflection
SmS	S reflection from the outer side of the Moho
SmSN	SmS multiple free surface reflection; N is a positive integer. For example, SmS2 is SmSSmS
SmP	S to P reflection from the outer side of the Moho
Lg	A wave group observed at larger regional distances and caused by superposition of multiple S-wave reverberations and SV to P and/or P to SV conversions inside the whole crust. The maximum energy travels with a group velocity around 3.5 km/s
Rg	Short period crustal Rayleigh wave
MANTLE PHASES	
P	A longitudinal wave, bottoming below the uppermost mantle; also an upgoing longitudinal wave from a source below the uppermost mantle
PP	Free surface reflection of P wave leaving a source downwards
PS	P, leaving a source downwards, reflected as an S at the free surface. At shorter distances the first leg is represented by a crustal P wave.
PPP	analogous to PP
PPS	PP to S converted reflection at the free surface; travel time matches that of PSP
PSS	PS reflected at the free surface
PcP	P reflection from the core-mantle boundary (CMB)
PcS	P to S converted reflection from the CMB
PcPN	PcP multiple free surface reflection; N is a positive integer. For example PcP2 is PcPPcP
Pz+P	(alt:PzP) P reflection from outer side of a discontinuity at depth z; z may be a positive numerical value in km. For example P660+P is a P reflection from the top of the 660 km discontinuity.
Pz -P	P reflection from inner side of discontinuity at depth z. For example, P660 -P is a P reflection from below the 660 km discontinuity, which means it is precursory to PP.
Pz+S	(alt:PzS) P to S converted reflection from outer side of discontinuity at depth z
Pz -S	P to S converted reflection from inner side of discontinuity at depth z
PScS	P (leaving a source downwards) to ScS reflection at the free surface
Pdif	(old:Pdiff) P diffracted along the CMB in the mantle
S	A shear wave, bottoming below the uppermost mantle; also an upgoing shear wave from a source below the uppermost mantle
SS	Free surface reflection of an S wave leaving a source downwards
SP	S, leaving source downwards, reflected as P at the free surface. At shorter distances the second leg is represented by a crustal P wave.
SSS	analogous to SS
SSP	SS to P converted reflection at the free surface; travel time matches that of SPS
SPP	SP reflected at the free surface
ScS	S reflection from the CMB
ScP	S to P converted reflection from the CMB

ScSN ScS multiple free surface reflection; N is a positive integer.
For example ScS2 is ScSScS

Sz+S (alt:SzS) S reflection from outer side of a discontinuity at depth z; z may be a positive numerical value in km. For example S660+S is an S reflection from the top of the 660 km discontinuity.

Sz -S S reflection from inner side of discontinuity at depth z. For example, S660 -S is an S reflection from below the 660 km discontinuity, which means it is precursory to SS.

Sz+P (alt:SzP) S to P converted reflection from outer side of discontinuity at depth z

Sz -P S to P converted reflection from inner side of discontinuity at depth z

ScSP ScS to P reflection at the free surface

Sdif (old:Sdiff) S diffracted along the CMB in the mantle

CORE PHASES

PKP (alt:P') unspecified P wave bottoming in the core

PKPab (old:PKP2) P wave bottoming in the upper outer core; ab indicates the retrograde branch of the PKP caustic

PKPbc (old:PKP1) P wave bottoming in the lower outer core; bc indicates the prograde branch of the PKP caustic

PKPdf (alt:PKIKP) P wave bottoming in the inner core

PKPpre (old:PKhKP) a precursor to PKPdf due to scattering near or at the CMB

PKPdif P wave diffracted at the inner core boundary (ICB) in the outer core

PKS Unspecified P wave bottoming in the core and converting to S at the CMB

PKSab PKS bottoming in the upper outer core

PKSbc PKS bottoming in the lower outer core

PKSdf PKS bottoming in the inner core

P'P' (alt:PKPPKP) Free surface reflection of PKP

P'N (alt:PKPN) PKP reflected at the free surface N -1 times; N is a positive integer. For example P'3 is P'P'P'

Pz -P' PKP reflected from inner side of a discontinuity at depth z outside the core, which means it is precursory to P'P'; z may be a positive numerical value in km

P'S' (alt:PKPSKS) PKP to SKS converted reflection at the free surface; other examples are P'PKS, P'SKP

PS' (alt:PSKS) P (leaving a source downwards) to SKS reflection at the free surface

PKKP Unspecified P wave reflected once from the inner side of the CMB

PKKPab PKKP bottoming in the upper outer core

PKKPbc PKKP bottoming in the lower outer core

PKKPdf PKKP bottoming in the inner core

PNKP P wave reflected N -1 times from inner side of the CMB; N is a positive integer

PKKPpre a precursor to PKKP due to scattering near the CMB

PKiKP P wave reflected from the inner core boundary (ICB)

PKNIKP P wave reflected N -1 times from the inner side of the ICB

PKJKP P wave traversing the outer core as P and the inner core as S

PKKS P wave reflected once from inner side of the CMB and converted to S at the CMB

PKKSab PKKS bottoming in the upper outer core

PKKSbc PKKS bottoming in the lower outer core

PKKSdf PKKS bottoming in the inner core

PcPP' (alt:PcPPKP) PcP to PKP reflection at the free surface; other examples are PcPS', PcSP', PcSS', PcPSKP, PcSSKP

SKS (alt:S') unspecified S wave traversing the core as P

SKSac SKS bottoming in the outer core

SKSdf (alt:SKIKS) SKS bottoming in the inner core

SPdifKS (alt:SKPdifs) SKS wave with a segment of mantle side Pdif at the source and/or the receiver side of the raypath

SKP Unspecified S wave traversing the core and then the mantle as P

SKPab SKP bottoming in the upper outer core

SKPbc SKP bottoming in the lower outer core

SKPdf	SKP bottoming in the inner core
S'S'	(alt:SKSSKS) Free surface reflection of SKS
S'N	SKS reflected at the free surface N -1 times; N is a positive integer
S'z -S'	SKS reflected from inner side of discontinuity at depth z outside the core, which means it is precursory to S'S'; z may be a positive numerical value in km
S'P'	(alt:SKSPKP) SKS to PKP converted reflection at the free surface; other examples are S'SKP, S'PKS
S'P	(alt:SKSP) SKS to P reflection at the free surface
SKKS	Unspecified S wave reflected once from inner side of the CMB
SKKSac	SKKS bottoming in the outer core
SKKSdf	SKKS bottoming in the inner core
SNKS	S wave reflected N -1 times from inner side of the CMB; N is a positive integer
SKiKS	S wave traversing the outer core as P and reflected from the ICB
SKJKS	S wave traversing the outer core as P and the inner core as S
SKKP	S wave traversing the core as P with one reflection from the inner side of the CMB and then continuing as P in the mantle
SKKPab	SKKP bottoming in the upper outer core
SKKPbc	SKKP bottoming in the lower outer core
SKKPdf	SKKP bottoming in the inner core
ScSS'	(alt:ScSSKS) ScS to SKS reflection at the free surface; other examples are: ScPS', ScSP', ScPP', ScSSKP, ScPSKP
NEAR SOURCE SURFACE REFLECTIONS (Depth phases)	
pPy	All P-type onsets (Py) as defined above, which resulted from reflection of an upgoing P wave at the free surface or an ocean bottom; WARNING: The character "y" is only a wild card for any seismic phase, which could be generated at the free surface. Examples are: pP, pPKP, pPP, pPcP, etc
sPy	All Py resulting from reflection of an upgoing S wave at the free surface or an ocean bottom; For example: sP, sPKP, sPP, sPcP, etc
pSy	All S-type onsets (Sy) as defined above, which resulted from reflection of an upgoing P wave at the free surface or an ocean bottom; for example: pS, pSKS, pSS, pScP, etc
sSy	All Sy resulting from reflection of an upgoing S wave at the free surface or an ocean bottom; for example: sSn, sSS, sScS, sSdif, etc
pwPy	All Py resulting from reflection of an upgoing P wave at the ocean's free surface
pmPy	All Py resulting from reflection of an upgoing P wave from the inner side of the Moho
SURFACE WAVES	
L	Unspecified long period surface wave
LQ	Love wave
LR	Rayleigh wave
G	Mantle wave of Love type
GN	Mantle wave of Love type; N is integer and indicates wave packets traveling along the minor arcs (odd numbers) or major arc (even numbers) of the great circle
R	Mantle wave of Rayleigh type
RN	Mantle wave of Rayleigh type; N is integer and indicates wave packets traveling along the minor arcs (odd numbers) or major arc (even numbers) of the great circle
PL	Fundamental leaking mode following P onsets generated by coupling of P energy into the waveguide formed by the crust and upper mantle
SPL	S wave coupling into the PL waveguide; other examples are SSPL, SSSPL
ACOUSTIC PHASES	
H	A hydroacoustic wave from a source in the water, which couples in the ground
HPg	H phase converted to Pg at the receiver side
HSg	H phase converted to Sg at the receiver side
HRg	H phase converted to Rg at the receiver side
I	An atmospheric sound arrival, which couples in the ground
IPg	I phase converted to Pg at the receiver side

ISg I phase converted to Sg at the receiver side
IRg I phase converted to Rg at the receiver side
T A tertiary wave. This is an acoustic wave from a source in the solid earth, usually trapped in a low velocity oceanic water layer called the SOFAR channel (SOund Fixing And Ranging)
TPg T phase converted to Pg at the receiver side
TSg T phase converted to Sg at the receiver side
TRg T phase converted to Rg at the receiver side
AMPLITUDE MEASUREMENT PHASES
A Unspecified amplitude measurement
AML Amplitude measurement for local magnitude
AMB Amplitude measurement for body wave magnitude
AMS Amplitude measurement for surface wave magnitude
END Time of visible end of record for duration magnitude
UNIDENTIFIED ARRIVALS
x (old:i,e,NULL) unidentified arrival
rx (old:i,e,NULL) unidentified regional arrival
tx (old:i,e,NULL) unidentified teleseismic arrival
Px (old:i,e,NULL,(P),P?) unidentified arrival of P-type
Sx (old:i,e,NULL,(S),S?) unidentified arrival of S-type

Annexe B

Liste des publications

Articles dans des revues internationales à comité de lecture

1. Seismic ray-tracing and Earth mesh modeling on various parallel architectures
M. GRUNBERG, S. GENAUD, C. MONGENET
The Journal of Supercomputing, Kluwer Academic Pub., pages 27–44, vol. 29(1), juillet 2004

Articles dans des revues nationales à comité de lecture

2. Calcul de rais en tomographie sismique : exploitation sur la grille
S. GENAUD, M. GRUNBERG
Technique et Science Informatiques, pages 591-608, vol. 24, num. 5, 2005

Communications à des manifestations internationales avec comité de lecture

3. Experiments in running a scientific MPI application on Grid'5000
M. GENAUD, S. GRUNBERG, C. MONGENET
International Symposium on Grid computing, high-performance and Distributed Applications (GADA'06), Montpellier, France, Oct 29 - Nov 3, 2006
4. Parallel Adaptive Mesh Coarsening for Seismic Tomography
M. GRUNBERG, S. GENAUD, C. MONGENET
SBAC-PAD 2004, 16th Symposium on Computer Architecture and High Performance Computing, IEEE Computer Society Press, Foz do Iguaçu, Brazil - octobre 2004
5. Parallel Seismic Ray-tracing in a Global Earth Mesh
M. GRUNBERG, S. GENAUD, C. MONGENET
2002 Parallel and Distributed Processing Techniques and Applications (PDPTA'02), pages 1151-1157, Las Vegas - juin 2002

Communications à des manifestations nationales à comité de lecture

6. Calcul de rais en tomographie sismique : exploitation sur la Grille
M. GRUNBERG, S. GENAUD
Renpar 15 INRIA, pages 179–186, La Colle sur Loup - octobre 2003

Communications à des manifestations internationales

7. Geographical ISC data Characterization with Parallel Ray-tracing
M. GRUNBERG, S. GENAUD, M. GRANET
Eos Trans. AGU, vol. 84(46), San Francisco - décembre 2003

Rapports de recherche

8. Experiments in running a scientific MPI application on Grid'5000
S. GENAUD, M. GRUNBERG, C. MONGENET
Rapport de recherche no 06-03, ICPS-LSIIT, Université Louis Pasteur, Strasbourg, France, juillet 2006.