Imagerie ultrasonore 2D et 3D sur GPU : application au temps réel et à l'inversion de forme d'onde complète

Etienne Bachmann

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *07/10/2016* **par :**

ETIENNE BACHMANN

Imagerie ultrasonore 2D et 3D sur GPU : application au temps réel et à l'inversion de forme d'onde complète.

JURY

ANNE-SOPHIE BONNET-BENDHIA	Directeur de recherche CNRS, ENSTA ParisTech	Présidente
DIDIER CASSEREAU	Maître de conférences, HDR, ESPCI Paris	Rapporteur
MARC BONNET	Directeur de recherche CNRS, ENSTA ParisTech	Rapporteur
VINCENT GIBIAT	Professeur des Universités, Université Toulouse 3	Directeur de thèse
XAVIER JACOB	Maître de conférences, Université Toulouse 3	Examinateur
DIMITRI KOMATITSCH	Directeur de recherche CNRS, LMA Marseille	Examinateur
SAMUEL RODRIGUEZ	Maître de conférences, Université de Bordeaux	Examinateur

---

École doctorale et spécialité :
MEGEP : Dynamique des fluides
Unité de Recherche :
Laboratoire PHASE
Directeur de Thèse :
Vincent GIBIAT
Rapporteurs :
Didier CASSEREAU et *Marc BONNET*

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *07/10/2016* **par :**

ETIENNE BACHMANN

Imagerie ultrasonore 2D et 3D sur GPU : application au temps réel et à l'inversion de forme d'onde complète.

JURY

ANNE-SOPHIE BONNET-BENDHIA	Directeur de recherche CNRS, ENSTA ParisTech	Présidente
DIDIER CASSEREAU	Maître de conférences, HDR, ESPCI Paris	Rapporteur
MARC BONNET	Directeur de recherche CNRS, ENSTA ParisTech	Rapporteur
VINCENT GIBIAT	Professeur des Universités, Université Toulouse 3	Directeur de thèse
XAVIER JACOB	Maître de conférences, Université Toulouse 3	Examinateur
DIMITRI KOMATITSCH	Directeur de recherche CNRS, LMA Marseille	Examinateur
SAMUEL RODRIGUEZ	Maître de conférences, Université de Bordeaux	Examinateur

---

École doctorale et spécialité :
MEGEP : Dynamique des fluides
Unité de Recherche :
Laboratoire PHASE
Directeur de Thèse :
Vincent GIBIAT
Rapporteurs :
Didier CASSEREAU et *Marc BONNET*

Remerciements

Après ces trois années de thèse, je suis très heureux de pouvoir écrire ces remerciements et saluer ceux qui ont contribué, de près ou de plus loin, à ce vaste travail.

Mes premiers remerciements s'adressent tout naturellement à mon directeur de thèse Vincent Gibiat, qui m'a d'abord en temps que professeur introduit l'univers de l'acoustique, puis offert la possibilité d'y poursuivre l'aventure en taillant sur mesure ce sujet de thèse. Si les aléas politiques sud américains et la prospection scientifique extra-acoustique en ont sérieusement altéré le contenu original, la liberté et la confiance qu'il m'a accordé pour en redéfinir les contours auront été déterminantes.

Un grand merci également à Xavier Jacob, dont les discussions quotidiennes sur l'avancement des différents projets, qu'ils soient scientifiques et pédagogiques, m'auront permis d'avancer plus loin. Un grand merci !

Je tiens à remercier Dimitri Komatitsch pour la spontanéité et l'enthousiasme avec lesquelles il a accueilli les perspectives d'une collaboration PHASE-LMA, qui a sans conteste donné à ces travaux une toute autre dimension.

Je remercie aussi les autres membres du jury, Anne-Sophie Bonnet, Samuel Rodriguez ainsi que Didier Cassereau et Marc Bonnet pour avoir accepté d'être rapporteurs du mémoire.

Merci à Pierre De Guibert, qui a toujours été très disponible et n'a jamais reculé devant les difficultés software/hardware que j'ai rencontrées, et notamment ces relations houleuses entre drivers graphiques et noyaux Linux qui m'ont invité à le solliciter plus d'une fois.

Je souhaite remercier aussi tous les autres membres du laboratoire PHASE pour ces trois ans passés ensemble.

A big thanks to all the Tromp's research team of Princeton University, and specially to Jeroen, Ryan, Yanhua, James and Heru who gave me all the right tools and confidence to obtain in such a short period new results.

Et au delà du travail au labo, je remercie tous les Toulousains ex-INSA et piliers du Frédéric Estèbe : Gweniche, Nicouille, Tucker, Victor, Torvi, La Fauste, Diouf, Normi, Peyo, Guégué & Cie. Le chantier est enfin fini !

Merci enfin à toute ma famille et mes proches qui m'ont toujours soutenu au cours de ces nombreuses années d'études. Merci beaucoup !

Liste des acronymes

API : Application Programming Interface

CHPI : Cycles d'Horloge Par Instruction

CMUT : Capacitive Micromachined Ultrasonic Transducer

CND : Contrôle Non Destructif

CPU : Central Processing Unit

CUDA : Compute Unified Device Architecture

DORT : Décomposition de l'Opérateur de Retournement Temporel

FLOP : FLoating-Point OPeration

FLOPS : FLoating-Point OPeration per Second

FMC : Full Matrix Capture

FTIM : Fast Topological IMaging

FWI : Full Waveform Inversion

GLL : Gauss-Lobatto-Legendre

GPU : Graphics Processing Unit

GPGPU : General-Purpose computing on Graphics Processing Unit

L-BFGS : Limited-memory Broyden-Fletcher-Golfarb-Shannon

PIV : Particle Image Velocimetry

SAFT : Synthetic Aperture Focusing Technique

SIMD : Single Instruction Multiple Data

SSD : Solid State Drive

TDTE : Time Domain Topological Energy

TFM : Total Focusing Method

TR-MUSIC : Time Reversal with MUltiple SIgnal Classification

Table des matières

Introduction

Exposition de la problématique

L'imagerie ultrasonore est utilisée dans de nombreux domaines, et notamment en imagerie médicale ou en contrôle non destructif. Néanmoins, dans chacun de ces domaines l'imagerie ultrasonore est souvent vue comme un moyen rapide et peu onéreux d'obtenir une image, et des techniques telles que le RADAR, les Rayons X ou encore l'IRM lui sont préférées lorsque qu'il s'agit d'obtenir une image de qualité. De fait, la grande majorité des procédés d'imagerie ultrasonore sont appréciés pour leur opération en temps réel, ce qui empêche un traitement numérique des données trop ambitieux. Ainsi, la plupart des échographes actuels affichent simplement l'évolution au cours du temps des données enregistrées par la barrette de transducteurs, ce qui limite significativement l'interprétation possible et la garantie qu'offre l'image obtenue. L'assimilation de l'échelle de temps à une échelle d'espace n'est possible qu'avec une connaissance a priori de la vitesse du son dans le milieu, qui est de plus par hypothèse la même partout. Cette estimation et cette approximation sont ensuite utilisées telles quelles dans ces méthodes et influencent directement le repositionnement des hétérogénéités du milieu inconnu. Enfin, les images obtenues sont généralement à deux dimensions, alors que le milieu à imager et la propagation des ondes dans ce milieu sont par nature à 3D, ce qui affaiblit à nouveau la qualité du résultat obtenu.

Par ailleurs, les GPU (Graphics Processing Unit), devenus les outils les plus performants pour effectuer du calcul scientifique, offrent une puissance de calcul qui n'est pas encore exploitée par la plupart des applications d'imagerie ultrasonore. Depuis une dizaine d'années, ces composants sont devenus programmables et leur popularité auprès de la communauté scientifique ne cesse de s'accroître. Ceci s'explique notamment par le fait que la vitesse d'exécution du programme peut être accélérée de plusieurs ordres de grandeur alors que le prix d'un GPU est du même ordre que celui d'un processeur, ou CPU. Ce bénéfice exige toutefois une réécriture parfois complexe de l'algorithme considéré dans le langage approprié, ce qui constitue la principale barrière à leur utilisation. Néanmoins, l'écart déjà conséquent de vitesse d'exécution sur les deux composants devrait continuer de s'accroître et justifie amplement cet effort de transition, et plus particulièrement dans des applications où le délai d'obtention de l'image est un facteur critique.

L'objectif de cette thèse est de tirer le plus grand bénéfice de ce nouvel outil de calcul, en ciblant deux applications complémentaires :

- Le premier objectif est d'offrir une imagerie en temps réel de type échographique, afin de répondre aux exigences de rapidité d'exécution nécessaire dans de nombreux domaines, tout en augmentant significativement la qualité de l'image par rapport aux applications concurrentes. A cette fin, nous avons poursuivi le développement du procédé d'imagerie FTIM (Fast Topological IMaging), héritage direct des travaux effectués au laboratoire PHASE ayant précédé cette thèse, pour aboutir au temps réel et à l'imagerie 3D. Cette méthode utilise d'une part la propriété de refocalisation inhérente au retournement temporel et à la propagation d'onde, qui est impliquée dans ce procédé et simulée numériquement de façon précise, par opposition à la plupart des autres méthodes qui utilisent une approximation géométrique de la propagation. Ce gain en précision numérique permet en particulier de s'affranchir de la nécessité d'utiliser un grand nombre d'acqui-

sitions pour obtenir une image. D'autre part, la corrélation des deux champs issus respectivement du signal source et des signaux enregistrés confère une seconde propriété de refocalisation à cette méthode. Enfin, la formulation dans le domaine fréquentiel du problème diminue d'environ 5 ordres de grandeurs le nombre d'opérations numériques nécessaires par rapport au domaine temporel, et l'algorithme de calcul suit le modèle 'Single Instruction Multiple Data' ce qui le rend propice à la parallélisation.

- Le second objectif est d'utiliser cette puissance de calcul pour obtenir la meilleure image possible. Pour cela, la notion même d'image dans un milieu fluide ou solide a été remise en question en se basant sur ce que reflètent réellement des données ultrasonores. Notre intérêt s'est alors tourné vers l'inversion de forme d'onde complète, ou Full Waveform Inversion (FWI) qui vise à reconstruire la carte des paramètres physiques du milieu inconnu, tels que la vitesse ou la densité. Le principe de la FWI repose sur la minimisation itérative de la différence entre données réelles et données simulées en modifiant localement les propriétés physiques du milieu numérique. Le premier intérêt de cette méthode est qu'en reconstruisant fidèlement la carte de vitesse du milieu, les interfaces présentes sont correctement repositionnées avec une grande robustesse. De fait, cette méthode d'inversion ne requiert pas de connaissance a priori sur le milieu, par opposition aux méthodes de type échographique. Le second intérêt est que la caractérisation quantitative des propriétés du milieu délivre une information bien plus riche que la plupart des autres applications, qui ne montrent que les interfaces internes. Cette information quantitative peut ainsi servir de solution à diverses applications, comme l'estimation de la densité des os, ce qu'aucune autre technique délivrant une simple image ne permet d'obtenir. Cette méthode issue de la géophysique a longtemps été impossible à mettre en œuvre sans un superordinateur à cause des nombreuses simulations de propagation d'onde qu'elle nécessite, mais l'augmentation constante de la puissance des GPUs rend à présent possible son application dans des délais raisonnables et à un coup matériel moindre.

Organisation du document

La première partie propose, en préambule au travail effectué, une vue d'ensemble sur les différents procédés d'imagerie ultrasonore existants, en se focalisant plus particulièrement sur les techniques issues du CND et de l'imagerie médicale. Nous remarquerons aussi comment l'image résultat répond à un besoin dans le domaine concerné, ainsi que les avantages et les limites de chacune de ces méthodes. Dans un deuxième temps, nous introduirons brièvement l'évolution du calcul scientifique, qui a suivi pendant plus de 40 ans l'évolution de la puissance des processeurs, et nous verrons comment le GPU a marqué une rupture technologique dans ce domaine.

La deuxième partie décrit la méthode d'imagerie FTIM, en explicitant son principe et son origine. Après avoir clairement défini le type de milieu que cette méthode permet d'imager et les hypothèses qui lui sont nécessaires, les détails de son implémentation sur GPU seront montrés et nous verrons à travers cet exemple les facteurs clés de l'optimisation d'un code sur GPU. Nous présenterons des résultats issus de données expérimentales, à 2D et à 3D. Enfin, nous aborderons les applications possibles de la méthode en valorisant les possibilités apportées par le temps réel.

La troisième partie introduit la Full Waveform Inversion, en décrivant dans un premier temps le problème inverse qu'elle tend à résoudre et sa formulation, puis sa résolution au travers des méthodes d'optimisation locale de type gradient et Newton. En raison de la nature non convexe et non linéaire de ce problème d'optimisation, les facteurs fondamentaux influant sur la convergence de la minimisation seront passés en revue. Enfin, comme la FWI repose fortement sur la précision de la simulation de la propagation d'onde, la méthode numérique des éléments spectraux que nous avons utilisée et son implémentation à travers le programme Specfem seront explicitées.

La quatrième partie abordera l'application de la FWI à l'échelle ultrasonore, où tout d'abord

les principales différences avec son utilisation en géophysique seront soulignées. Plus particulièrement, la relative liberté que l'on possède sur la géométrie des dispositifs expérimentaux à l'échelle ultrasonore sera mise à profit pour maximiser la quantité d'information détenue dans les données, en vue de favoriser la convergence de la FWI. Nous nous intéressons à la reconstruction des cartes de vitesse dans divers milieux, en distinguant les problèmes à faible et à forte variation de vitesse. Nous verrons que la stratégie d'inversion diffère entre ces deux catégories de problème et des résultats d'inversion à 2D et 3D seront présentés. Enfin, des tests numériques complémentaires seront menés en vue de l'application à des données réelles.

Chapitre 1

Etat de l'art

1.1 Panorama des procédés d'imagerie en acoustique ultrasonore

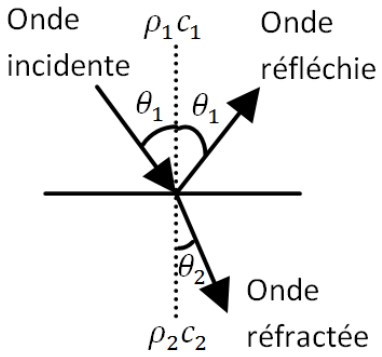
La découverte de l'effet piézoélectrique par Pierre et Jacques Curie en 1880 [1] a offert la possibilité d'émettre et d'enregistrer une onde sonore, et ne fut exploitée que 30 ans plus tard par Paul Langevin à des fins proches de l'imagerie. En démontrant la possibilité de détecter un objet distant sous l'eau avec des ondes acoustiques [2], cette expérience marque le début de l'histoire de l'imagerie ultrasonore, dont l'intérêt au sein de la communauté scientifique n'a cessé de croître depuis. L'expérience de Langevin illustre également un cas d'application où les ultrasons sont les seuls vecteurs d'information possibles du milieu, dans la mesure où les ondes électromagnétiques sont sujettes à une trop forte atténuation. Il est ainsi peu surprenant de constater que les principaux domaines où l'utilisation des ultrasons joue un rôle déterminant, tels que l'imagerie sous-marine, l'imagerie médicale ou encore le contrôle non destructif sont ceux qui impliquent des milieux opaques aux ondes électromagnétiques, empêchant ainsi l'imagerie "naturelle" photographique ou les techniques de type RADAR d'opérer. De manière générale, l'amélioration de la qualité des images délivrées en imagerie ultrasonore est grandement liée à l'outil d'émission et d'enregistrement de ces ondes, en particulier grâce à l'utilisation de fréquences de plus en plus élevées et d'un nombre croissant de capteurs. Néanmoins, l'avènement de l'électronique puis de l'informatique a également révolutionné la façon dont les données acquises peuvent être traitées, ce qui explique la relative jeunesse du panel de techniques décrites dans cette partie.

Le SONAR et l'échographie

Principe d'imagerie par scan

Le principe historique de fonctionnement des SONAR (SOund Navigation And Ranging) actifs utilisés pour la détection d'objets sous marins, et des échographes utilisés en imagerie médicale ou en contrôle non destructif est identique. Il repose sur l'émission d'une onde ultrasonore transitoire, et de son écoute à travers des hydrophones ou des transducteurs, qui utilisent pour la majorité l'effet piézoélectrique et peuvent à la fois transmettre ou enregistrer une onde. L'imagerie proposée par ces appareils révèle les contrastes d'impédance Z à l'intérieur du milieu inconnu, que l'on peut identifier à des interfaces ou des petites hétérogénéités, en vertu de la loi de Snell-Descartes, illustrée sur 1.1.

Problème monodimensionnel. L'exemple 1.2 ci-dessus montre le mode d'imagerie connu sous le nom de A-Scan à la base du fonctionnement des SONAR, et est aussi utilisé en imagerie médicale. Il constitue le mode d'imagerie existant le moins onéreux du marché. On peut ainsi voir sur 1.2(b) la signature acoustique de l'interface qui sépare les deux milieux, qui a engendré une onde réfléchie lors du passage de l'onde émise. L'image est construite en récupérant l'enveloppe



$$r = \frac{Z_2 \cos(\theta_1) - Z_1 \cos(\theta_2)}{Z_2 \cos(\theta_1) + Z_1 \cos(\theta_2)}$$

$$\frac{\sin(\theta_1)}{c_1} = \frac{\sin(\theta_2)}{c_2}$$

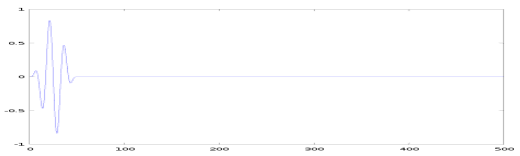
$Z_i = \rho_i c_i$ pour $i \in [1, 2]$

où :

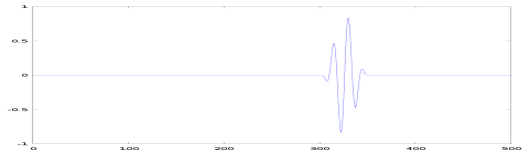
r est le coefficient de réflexion

Z_i est l'impédance caractéristique du milieu i

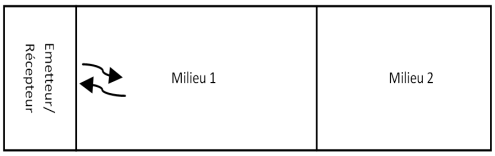
FIGURE 1.1 – Illustration de la loi de Snell-Descartes.



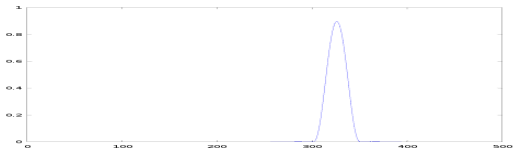
(a) Signal émis



(b) Signal reçu



(c) Milieu de propagation



(d) Image du milieu

FIGURE 1.2 – Principe d'imagerie par scan.

du signal enregistré, afin de retirer les oscillations dues à la nature ondulatoire du phénomène. En supposant une vitesse constante c_1 dans le milieu 1, on peut identifier l'échelle de temps à une échelle de distance par linéarité, et en considérant la durée de l'aller-retour de l'onde : $x = \frac{c_1 t}{2}$. Enfin, si l'onde transmise dans le milieu a une amplitude suffisante, elle peut à son tour révéler une interface située plus en profondeur. Comme l'onde transmise doit traverser une seconde fois la première interface pour revenir au transducteur, l'amplitude de l'onde écoutée par le transducteur est alors directement proportionnelle au contraste d'impédance entre le milieu 1 et le milieu 2. Ceci explique pourquoi des zones d'ombre apparaissent sur les échographies médicales, en particulier derrière les os qui possèdent une vitesse et une densité très supérieures aux autres tissus.

On peut aussi comprendre pourquoi l'aspect transitoire de l'onde émise, liée à sa longueur d'onde, influence la précision de la localisation de l'interface, et justifie la montée en fréquence des transducteurs. Outre la définition de l'interface, l'intérêt est de pouvoir distinguer la présence de deux interfaces proches l'une de l'autre, ce qui correspond à la résolution axiale. Cette montée en fréquence est toutefois limitée par deux facteurs :

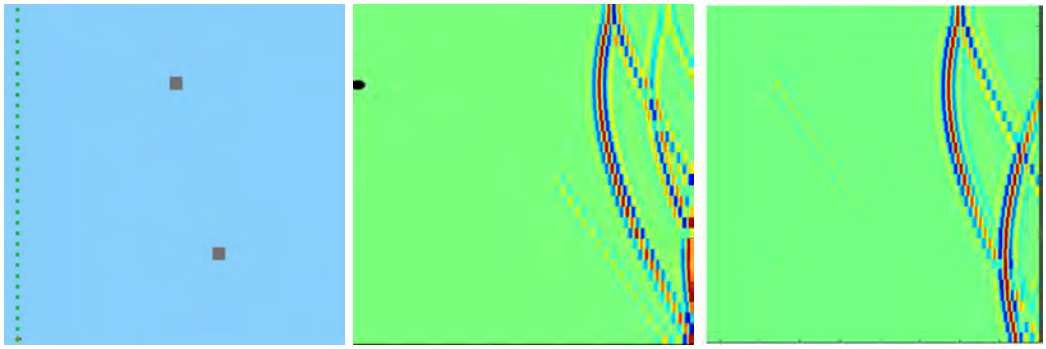
— Elle implique une miniaturisation des composants du transducteur et une augmentation de la fréquence d'échantillonnage qui se traduisent par un coût matériel substantiellement plus élevé.

— Les ondes ultrasonores sont soumises au phénomène d'atténuation qui est usuellement modélisé par un facteur exponentiel $e^{-\alpha x}$ où $\alpha \propto f^2$, ce qui rend le signal reçu inexploitable car indiscernable du niveau de bruit. Par ailleurs, la solution d'augmenter l'amplitude du signal émis a ses limites : dans le cas médical, elle génère un échauffement non négligeable des tissus traversés par l'onde qui est la conséquence du phénomène d'atténuation, et dans le cas sous-marin, elle peut nuire à la faune locale, comme en atteste la mort de plusieurs baleines lors d'une expérience de l'US Navy avec des amplitudes d'émission à 230dB. Dans

la pratique, les échographes actuels emploient le Time Gain Control (ou TGC), qui consiste à amplifier exponentiellement la sensibilité de l'appareil pour détecter les arrivées tardives. Cette solution n'est toutefois pas sans défauts : le bruit ambiant ne dépend pas du temps, et se trouve également amplifié ce qui réduit la qualité de l'enregistrement.

Ainsi, un compromis doit être trouvé en fonction de la géométrie du milieu à imager.

Problème à deux dimensions. L'extension à deux dimensions du procédé d'imagerie précédemment décrit est nommée B-Scan, qui implique soit un déplacement du capteur mono-élément soit l'utilisation une barrette linéaire de transducteurs. Il constitue encore aujourd'hui le principal mode de fonctionnement des échographes.



(a) Milieu de propagation

(b) Signaux RF avec une seule source, positionnée au point noir.

(c) Signaux RF avec un front d'onde plan

FIGURE 1.3 – Acquisition de données avec une barrette de transducteurs

L'exemple ci-dessus démontre que l'extension à deux dimensions n'est pas immédiate pour deux raisons :

1. La figure 1.3(b) illustre la conséquence de la divergence géométrique des ondes, l'énergie envoyée par la voie i se propage sur les deux dimensions et la signature acoustique de la deuxième hétérogénéité est visible, y compris sur le A-scan de la voie i. Cette considération déjà valable pour l'imagerie A-scan montre une première limite à ce principe d'imagerie. Dans le cas sous-marin, une première solution à ce problème a été apportée en contrôlant la directivité de l'onde émise au moyen de réflecteurs coniques à proximité de la source, afin de focaliser dans une direction précise le front d'onde émis. En imagerie médicale ou en CND, les contraintes géométriques ne permettent pas ce type de solution.
2. Les directions de propagation de l'onde réfléchie sur l'hétérogénéité dépendent de la géométrie de ce dernier. On peut voir sur les figures 1.3(b) et 1.3(c) que l'ensemble des voies enregistrent l'écho des deux hétérogénéités. Cette situation constitue en fait le cas idéal à imager : de petites hétérogénéités qui réfléchissent uniformément l'onde dans toutes les directions, avec un front d'onde enregistré qui évolue de façon continue d'une voie à l'autre. Comme nous le verrons par la suite, l'hypothèse de petites hétérogénéités est posée dans de nombreux algorithmes de résolution.

Ces deux raisons ne sont également pas sans conséquences sur le fait de traiter le problème d'imagerie en deux dimensions alors que l'onde se propage physiquement dans un milieu 3D. En particulier, l'onde émise peut se réfléchir sur des hétérogénéités situées en dehors du plan de résolution du problème, et l'onde réfléchie peut revenir dans le plan des transducteurs, causant ainsi des artefacts dans l'image obtenue. Cet effet peut être minimisé en maîtrisant la directivité de la source, surtout possible pour le cas sous-marin comme nous l'avons vu. L'unique alternative est de traiter le problème à 3 dimensions, ce qui délivre alors une information bien plus riche quant aux propriétés du milieu.

Acquisition des données

Pour le traitement des problèmes à deux ou trois dimensions, le schéma classique d'acquisition des données est le suivant :

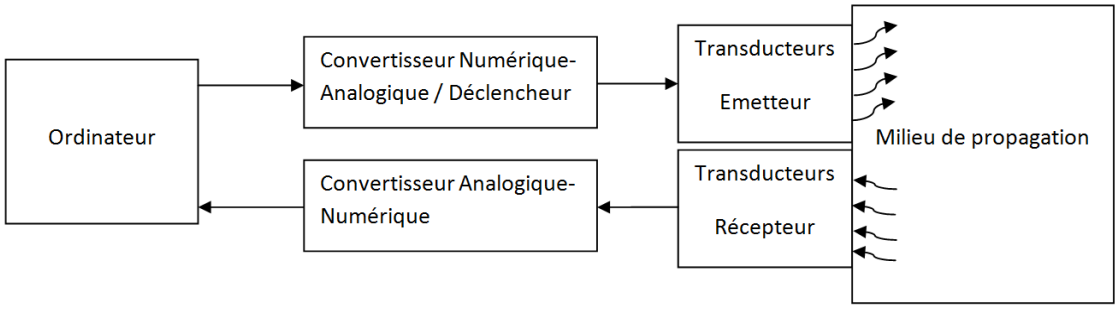


FIGURE 1.4 – Schéma de la chaîne d'acquisition des données ultrasonores. Dans une configuration typique d'imagerie échographique, les transducteurs servant à l'émission servent aussi à la réception.

La plupart des transducteurs utilisent le principe piézoélectrique, ce qui entraîne une bande passante assez limitée, de largeur approximativement équivalente à la fréquence centrale. L'onde de pression générée par le transducteur est déterminée par cette réponse en fréquence électro-mécanique. La plupart des dispositifs utilisent des impulsions carrées de tension, de durée de créneau réglable et souvent choisie comme étant égale à la demi-période de la fréquence centrale du transducteur, ou plus rarement des signaux programmables. Schématiquement, l'onde émise correspond à un sinus à la fréquence centrale ω_c du transducteur modulé par une enveloppe gaussienne ou à une ondelette de Ricker. D'autre part, la possibilité de contrôler le déclenchement du signal émis sur chacun des transducteurs de la barrette échographique offre une certaine liberté, qui est exploitée le plus souvent de trois façons différentes :

1. Le même signal est appliqué à toutes les voies, ce qui permet de générer un front d'onde plan. L'application d'un facteur de retard croissant ou décroissant linéairement selon le numéro de voie incline l'incidence de ce front d'onde proportionnellement au facteur de retard. Ainsi, en faisant varier les angles d'incidence des fronts d'onde, le milieu est insonifié différemment ce qui augmente la quantité d'information reçue. Cette technique est connue sous le nom de spatial compounding, et permet aussi de diminuer les effets d'ombre associés à la façon dont le milieu a été insonifié.
2. La divergence des ondes émises et réfléchies affaiblit l'amplitude du signal reçu. Les techniques de focalisation inversent l'effet de divergence de l'onde émise en créant une onde convergente en un point (x, y) donné du milieu à imager, en appliquant une loi de retard du type $\frac{\sqrt{y^2+x^2(n-n_0)^2}}{c}$, avec n_0 l'abscisse du transducteur le plus proche du point (x, y), et n l'abscisse du transducteur pour lequel est appliquée la loi de retard. En répétant cette procédure pour chacun des points du milieu à imager, le ratio signal sur bruit de l'image résultante est significativement amélioré. Cette technique possède cependant deux défauts : le grand nombre d'acquisitions rallonge le délai d'obtention d'une image, ce qui le rend incompatible avec des applications telles que l'estimation d'écoulements rapides ou l'échocardiographie, et en imagerie médicale induit un échauffement conséquent des tissus insonifiés. A cause de ces défauts, cette procédure qui a été très utilisée au début de l'ère des scanners à ultrason multi-éléments a été remplacée par le type d'acquisition (3).
3. Un seul transducteur émet au cours d'une acquisition alors que tous les autres écoutent. Cette méthode appelée réponse inter-élément, ou encore Full Matrix Capture (FMC) est à l'origine d'une multitude de traitements. Elle peut avoir le défaut d'offrir un rapport signal sur bruit assez faible car un seul transducteur est utilisé à la fois. Une solution consiste alors à utiliser plusieurs transducteurs adjacents lors d'une acquisition au lieu d'un seul.

Formation d'une image à partir de la Full Matrix Capture

La Total Focusing Method (TFM) consiste à focaliser virtuellement un front d'onde en chaque point du domaine à partir des données de la FMC. Cette technique a été reconnue comme le 'gold standard' [3] des méthodes d'imagerie linéaire. La focalisation est obtenue en calculant pour chaque point du domaine à imager le temps de trajet théorique de l'onde par rapport à chacun des transducteurs. Si le domaine à imager possède des interfaces connues avec des vitesses connues, le principe de Fermat, qui indique que le trajet de l'onde est celui qui minimise le temps de vol, est toujours appliqué. Ceci induit l'hypothèse que l'onde se comporte comme si elle avait une fréquence infinie. On peut alors construire la matrice de temps de trajet entre un point du domaine (i, j) et le transducteur k de coordonnées $(x_k, 0)$, qui dans le cas d'une vitesse constante dans le milieu est : $T(i, j; k) = \frac{\sqrt{(x_k - x_i)^2 + y_j^2}}{c}$. L'image est obtenue en sommant les contributions de chacune des voies en chaque point :

$$I(i,j)_{TFM} = \sum_{k,l} \mathrm{BSCAN}(T(i,j;k) + T(i,j;l), k, l) \tag{1.1}$$

où k représente un élément émetteur, l un élément récepteur, et $\mathrm{BSCAN}(t, k, l)$ la fonction qui associe la valeur de l'enregistrement à l'instant t du récepteur k lors de l'émission de l'élément k.

Parfois, afin de diminuer les effets inhérents à la directivité des transducteurs, une fonction d'apodisation $F(k, l)$ est incorporée. Dans la pratique, les images obtenues en présence de bruit montrent un meilleur ratio signal sur bruit lorsque le facteur d'apodisation est ajouté [4].

La Synthetic Aperture Focusing Technique (SAFT) se base sur la même relation, en utilisant moins d'information : Seul le signal enregistré par la voie qui a émis est considéré. La formule est donc la suivante :

$$I(i,j)_{SAFT} = \sum_{k} BSCAN(T(i,j;k) + T(i,j;k), k, l) \tag{1.2}$$

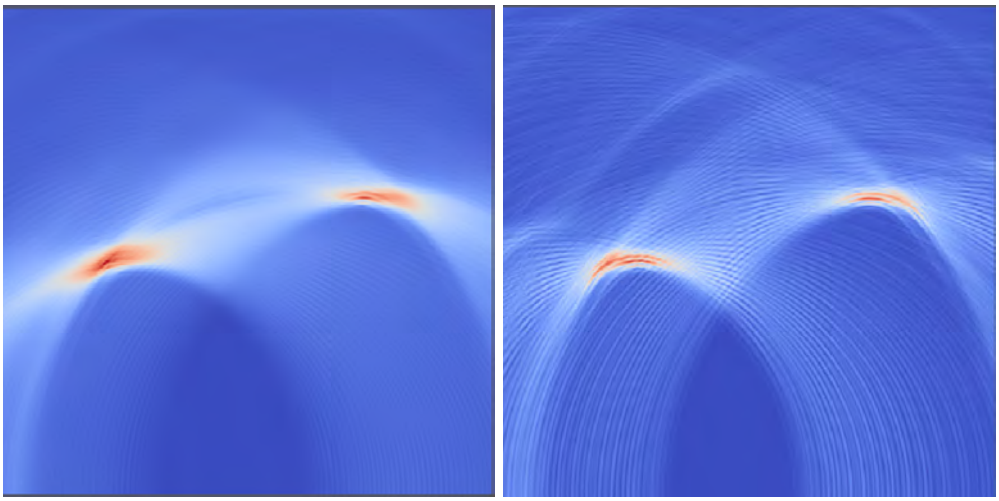
Comme moins d'information est utilisée, le résultat est également moins précis que pour la TFM comme on peut le voir sur 1.5. Cette méthode présente cependant deux intérêts :

- Elle peut être utilisée avec un seul transducteur qui sera translaté par la suite. Dans le cas d'une barrette de transducteurs, un convertisseur analogique numérique à une seule voie peut suffire. Le coût expérimental est alors significativement réduit.
- L'évaluation du critère I_{SAFT} requiert moins d'opérations mathématiques que le critère I_{TFM}. Cela peut permettre d'obtenir du temps réel lorsque les ressources matérielles effectuant le calcul sont limitées.

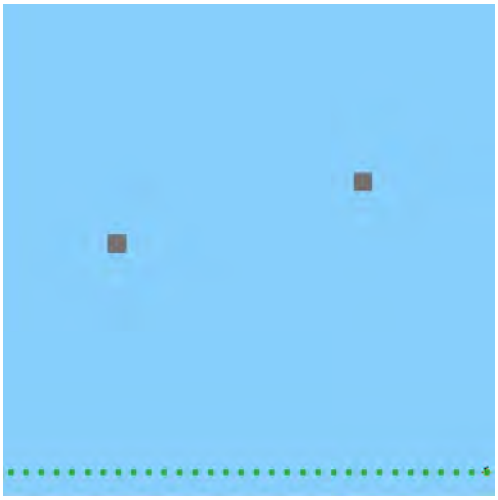
Ces méthodes, très utilisées en imagerie médicale et en CND, ont du fait de leur linéarité l'avantage de ne nécessiter que peu de calculs et d'être ainsi compatibles avec le temps réel. Néanmoins, elle ne sont destinées qu'à imager la présence de petites hétérogénéités dans un milieu. Elles utilisent la connaissance a priori des temps de vol en chaque point du milieu, ce qui suppose que la carte de vitesse est totalement connue. Ces modes d'imagerie reposent sur les hypothèses du lancer de rayon, et suivant l'approximation de Born, qui considère ici que l'onde enregistrée est soit l'onde émise, soit une réflexion de l'onde émise. Les effets de double réflexion et au-delà ne sont donc pas pris en compte, de même que ceux de la combinaison d'une onde réfractée puis réfléchie, alors qu'ils peuvent apparaître dans les données. Une extension de la TFM à l'identification d'une interface à 3D est présentée dans [5] avec des modifications de la carte de vitesse, mais en impliquant un temps de calcul bien plus élevé tout en conservant le lancer de rayon comme modèle de propagation.

Méthodes exploitant le retournement temporel

Les méthodes de retournement temporel utilisent la réversibilité de l'équation d'onde pour déterminer les contrastes d'impédance dans le milieu à imager. Cette propriété, connue sous le nom de symétrie CPT et commune à la plupart des phénomènes physiques, est particulièrement vérifiée avec l'équation d'onde, de par sa nature hyperbolique : dans le cas à deux dimensions,



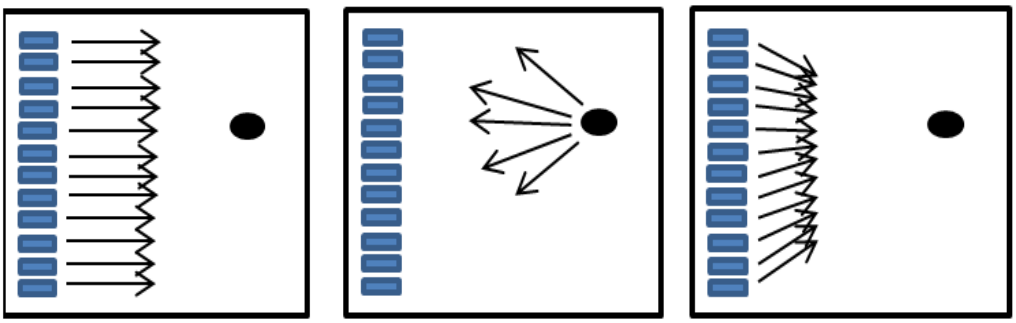
(a) Image avec TFM (b) Image avec SAFT



(c) Milieu "réel" pour la simulation des données

FIGURE 1.5 – Résultats d'expériences numériques avec les méthodes SAFT et TFM.

le seul enregistrement du champ acoustique sur un segment à une dimension suffit pour pouvoir inverser le processus, comme décrit sur le schéma 1.6.



(a) Emission d'un signal (b) Réception des échos (c) Retournement temporel des échos enregistrés

FIGURE 1.6 – Principe du retournement temporel.

La première mise en œuvre de ce principe fut expérimentale : en se basant sur une acquisition de type spatial compounding (SP) ou Full Matrix Capture (FMC), le signal reçu est inversé et est utilisé comme signal d'entrée. Lors de cette seconde acquisition, le front d'onde émis se focalise automatiquement sur les hétérogénéités présentes dans le milieu. L'opération peut être réitérée, afin de maximiser l'énergie envoyée sur les hétérogénéités. Cette approche, développée par le Laboratoire Ondes et Acoustique au début des années 90 avait été effectuée avec des composants électroniques dédiés (registres pour stocker les signaux enregistrés) ce qui permet une imagerie en temps réel [6]. Avec la démocratisation de l'informatique, l'inversion des signaux peut être gérée à partir d'un ordinateur comme illustré sur 1.4 pour un résultat similaire et éviter le coût possiblement élevé de l'électronique dédiée. Cette approche expérimentale est à présent utilisée à d'autres fins que l'imagerie, en particulier pour détruire les calculs rénaux [7].

Le retournement temporel a également été formalisé mathématiquement en se basant sur une acquisition de type FMC. Pour une barrette d'acquisition à N transducteurs et pour une fréquence ω donnée, les signaux enregistrés $S(\omega)$ peuvent s'exprimer dans le domaine fréquentiel comme une transformation linéaire du signal émis $E(\omega)$:

$$S(\omega) = R(\omega)E(\omega) \tag{1.3}$$

La matrice $R(\omega)$ peut être obtenue expérimentalement, en calculant le coefficient de Fourier des signaux enregistrés associé à la fréquence ω que l'on choisit généralement égale à la fréquence centrale des transducteurs, en utilisant un Dirac comme signal d'entrée, avec une acquisition de type FMC. $R_{ij}(\omega)$ contient le coefficient de Fourier à la fréquence ω caractérisant la réponse impulsionnelle émise par le transducteur i et enregistrée par le transducteur j. Dans le cas d'un milieu constitué de petites hétérogénéités, cet opérateur peut se décomposer en plusieurs morceaux, détaillant le processus physique :

$$R = A_r A_e \, {}^t HCH \tag{1.4}$$

Dans l'équation 1.4, H décrit la propagation dans le milieu de l'émetteur vers l'hétérogénéité, C l'interaction entre l'onde et l'hétérogénéité, et A_e et A_r caractérisent la réponse électro-acoustique de l'émetteur et du récepteur respectivement. Le retournement temporel peut s'exprimer avec ce formalisme, en prenant le conjugué de la transposée de la matrice R, afin de retourner le signal. Ainsi,

$$T = R^* R \tag{1.5}$$

La méthode de Décomposition de l'Opérateur de Retournement Temporel (DORT) se base sur l'équation 1.5 , en calculant les valeurs propres et les vecteurs propres de la matrice T, aussi appelée matrice ou opérateur de retournement temporel. La méthode montre que les valeurs propres les plus élevées sont les signatures des hétérogénéités les plus réfléchissantes, et que le vecteur propre ν_j associé à une valeur propre μ_j élevée est le signal qui serait reçu par chacune des voies si le milieu n'était constitué que de l'hétérogénéité associée à μ_j. Il est alors possible de retrouver la position spatiale de l'hétérogénéité j en simulant l'émission du vecteur propre ν_j, que l'on peut réaliser soit de façon expérimentale, en émettant le signal ν_j, soit simuler numériquement sa propagation. L'onde associée se focalisera alors sur la position de l'hétérogénéité.

Une autre méthode, nommée Time Reversal with MUltiple SIgnal Classification (TR-MUSIC) qui est très utilisée en CND se base aussi sur la décomposition de la matrice T. Elle utilise la réciprocité de la propagation d'onde du transducteur i au transducteur j, ce qui implique que la matrice R est symétrique, et ainsi que la matrice T est hermitienne. Il en découle alors l'orthogonalité des vecteurs propres de la matrice T, qui est à l'origine du critère d'imagerie défini par la méthode :

$$I(r) = \frac{1}{\sum_{i=M+1}^{N} \mid \langle \mu_i^* | G(R_i, r) \rangle \mid^2} \tag{1.6}$$

Le produit scalaire entre les vecteurs propres associés au bruit μ_i et l'opérateur de propagation $G(R_i, r)$ s'annule à la position des hétérogénéités du milieu, conduisant à une très haute valeur du critère d'imagerie. Il a été montré de façon expérimentale que cette méthode résiste très bien

à la présence de bruit, et que sa résolution et son pouvoir de séparation sont en dessous de la longueur d'onde de la fréquence principale de l'onde émise [8], ce qui explique sa popularité.

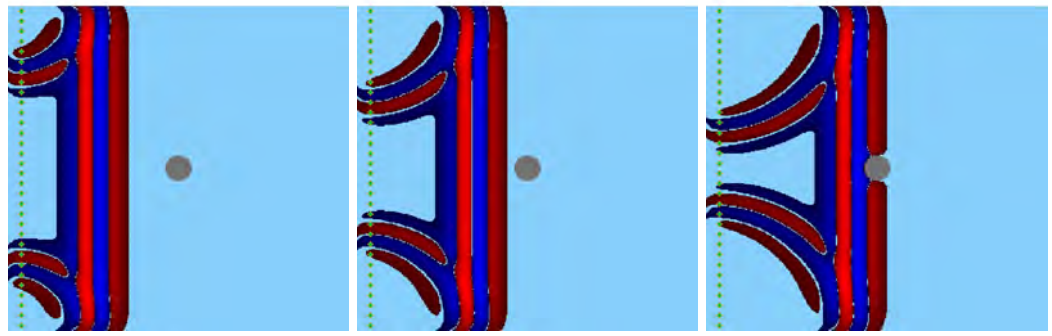
Ainsi, les méthodes DORT et TR-MUSIC permettent également d'imager la présence de petites hétérogénéités dans un milieu inconnu. L'utilisation de l'opérateur de propagation $G(R_i, r)$ permet d'incorporer un modèle de propagation complexe [9] si les propriétés du milieu sont connues à l'avance. Ces méthodes imposent en principe d'imager un milieu contenant moins d'hétérogénéités qu'il n'y a de transducteurs, mais cette difficulté peut être contournée en ayant recours au fenêtrage temporel des échantillons reçus, et en traitant chaque portion du domaine de façon indépendante. Ces deux méthodes utilisent des hypothèses et proposent une qualité d'imagerie comparables à la TFM, mais les calculs des valeurs et des vecteurs propres empêchent une utilisation en temps réel. Enfin, la connaissance a priori de la carte de vitesse est toujours nécessaire pour replacer correctement les hétérogénéités.

Remarques et conclusions

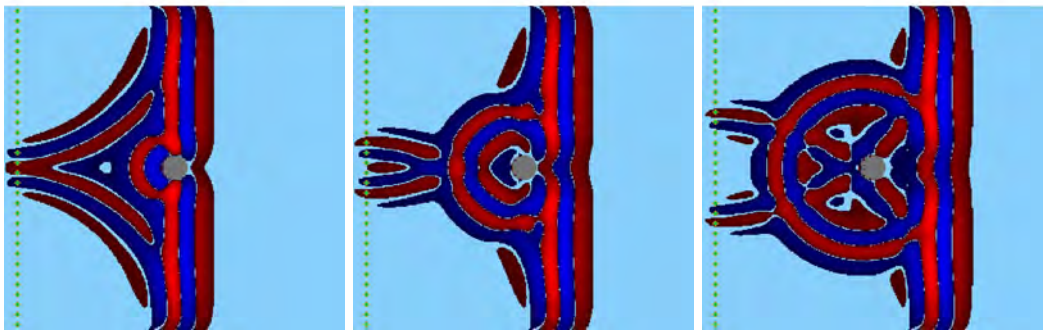
Imagerie des petites hétérogénéités Il est intéressant d'observer comment ces méthodes, qui visent à imager la présence de petites hétérogénéités, répondent à un besoin dans leur domaine d'application. Pour le contrôle non destructif, les matériaux utilisés sont souvent bien caractérisés acoustiquement, mais les procédés de fabrication ou l'usure peuvent engendrer pour des raisons diverses l'apparition de petites bulles d'air piégées dans un matériau, ou de micro-fissures. Lorsque la détection de ces imperfections est nécessaire, ces techniques échographiques répondent bien à ce besoin, de par leur faible coût matériel et leur rapidité. En imagerie médicale, le speckle désigne les très nombreuses hétérogénéités de tailles très inférieures à la longueur d'onde présentes dans tous les tissus du corps humain. La présence du speckle est souvent considérée comme néfaste car il affecte le contraste de l'image et rend les interfaces entre les différents tissus plus difficiles à observer, par opposition à un résultat d'imagerie de type IRM où un même tissu apparaît de façon homogène, ce qui explique pourquoi de nombreux articles sont consacrés à son élimination [10, 11]. Cependant, la notion même d'image échographique est intimement liée à la présence de ces nombreux diffuseurs, car la variation de leur nombre et de leur impédance acoustique d'un milieu à l'autre influence l'intensité, ou niveau de gris, associée à une zone donnée, ce qui permet l'identification d'un tissu ou d'un organe et ses contours. Par ailleurs, dans les cas où le milieu de référence est en mouvement, le speckle tracking, utilisé en échocardiographie [12] ou en imagerie des ondes de cisaillement, peut se révéler très utile pour obtenir une carte de déplacement de ce milieu en suivant l'évolution spatiale de chacune des petites hétérogénéités.

Limites et phénomènes non modélisés L'approximation de Born, nécessaire dans toutes ces méthodes, est sans doute l'hypothèse la plus contraignante. Elle empêche par exemple l'imagerie d'un milieu poreux, dont la structure a priori non connue influence fortement la trajet de l'onde dans ce milieu. Même si un modèle de propagation d'onde précis peut être utilisé, l'approximation de Born est à l'origine de plusieurs approximations sur la façon dont l'onde s'est propagée. Dans le cas des petites hétérogénéités, qui ne sont pas des points matériels dans la réalité, la propagation est grandement influencée par leur géométrie, et ainsi l'onde réfléchie comme celle réfractée sont réémises de façon inhomogènes. L'approximation géométrique de la propagation d'onde se combine souvent avec l'approximation de Born et conduit aussi à plusieurs simplifications du phénomène réel. Dans le cas d'hétérogénéités plus petites que la longueur d'onde locale, la combinaison de l'onde réfléchie par la face avant de l'hétérogénéité et celle réfléchie par la face arrière de l'hétérogénéité peut générer une interférence destructive dans certaines directions de repropagation. Enfin, la vitesse de l'onde à l'intérieur de l'hétérogénéité est généralement différente du milieu environnant, ce qui ralentit ou accélère légèrement le front d'onde qui traverse l'hétérogénéité. Dans le cas de très forts contrastes d'impédance, comme on peut le voir avec des bulles d'air, l'onde est localement totalement réfléchie. Ce phénomène est toutefois compensé par la diffraction, qui va venir 'combler' le trou du front d'onde une fois derrière l'hétérogénéité, avec un léger retard de phase, comme on peut l'observer sur la figure 1.7. Si tous ces effets ne sont pas

très importants un à un, leur accumulation, comme on peut le constater dans le domaine médical, peut s'avérer importante et parfois difficile à interpréter.



(a) $t = 6 \times 10^{-6}$ s (b) $t = 7.5 \times 10^{-6}$ s (c) $t = 9 \times 10^{-6}$ s



(d) $t = 1.05 \times 10^{-5}$ s (e) $t = 1.2 \times 10^{-5}$ s (f) $t = 1.35 \times 10^{-5}$ s

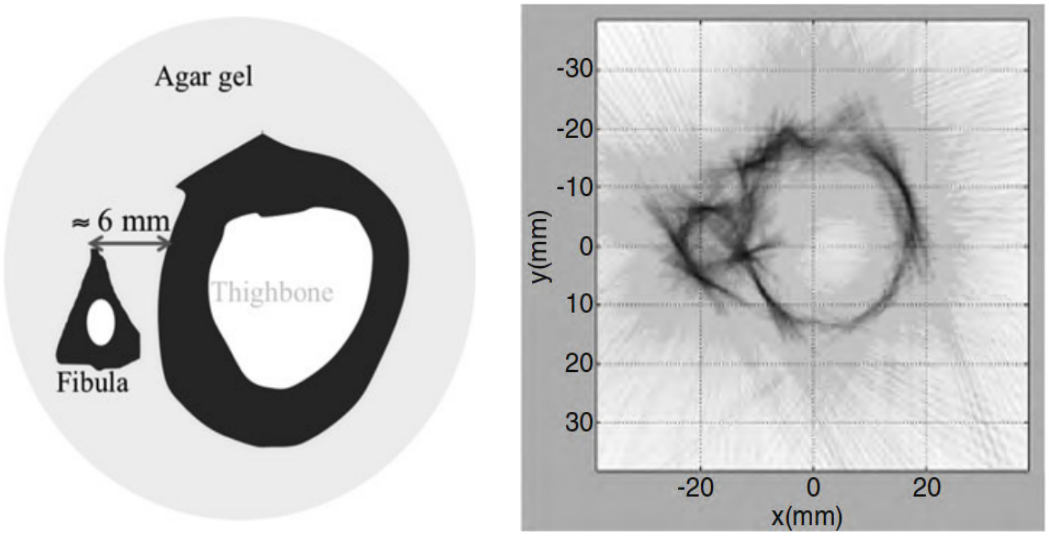
FIGURE 1.7 – Film de l'interaction entre une onde acoustique et une bulle d'air. Sur les dernières frames, le phénomène de diffraction compense la réflexion totale à l'entrée de la bulle.

Une autre conséquence de l'approximation géométrique de la propagation est la suivante : les interactions de l'onde avec des structures prédéfinies du milieu à imager, telles que la géométrie du fond d'un objet en CND, ou des os en imagerie médicale, ne sont pas bien prises en compte. Cette approximation se traduit dans les méthodes SAFT et TFM avec l'utilisation d'un temps de vol unique d'un point à un autre, basé sur le principe de Fermat. Pour les méthodes numériques de retournement temporel, le formalisme associé à la matrice de retournement temporel implique également l'unicité de ce temps de vol. De fait, l'intérêt d'un modèle de propagation élaboré pour ces méthodes se cantonne à l'incorporation du phénomène d'atténuation et à une meilleure modélisation de l'effet de la géométrie du transducteur sur la propagation. Ainsi, ces méthodes ne sont efficaces que pour des milieux globalement homogènes, ce qui restreint considérablement leur utilisation, ou expose alors l'image obtenue à la présence d'artefacts, comme l'effet miroir connu des échographistes illustré par la fig. 1.8.

La carte de vitesse, souvent restreinte à la connaissance de sa valeur moyenne, influence directement le temps de vol des ondes, est également supposée connue dans chacune de ces méthodes. Cette hypothèse peut se justifier par la bonne connaissance a priori du matériau à imager. Toutefois, cette connaissance reste sujette à une certaine incertitude, qui peut aussi dépendre des conditions de l'inspection : en CND, une variation de la température de l'objet ou de son taux d'humidité peuvent la modifier. Enfin, si la valeur moyenne de la vitesse est connue, les variations locales ne le sont généralement pas. L'influence de la vitesse sur le temps de vol se traduit par un mauvais repositionnement de l'hétérogénéité et une distorsion si une structure interne mesurant plusieurs longueurs d'onde de long est imagée. Dans la plupart des cas toutefois, ces erreurs sont minimes ou peu importantes vis-à-vis du problème étudié.

Il n'est pas toujours possible d'obtenir une onde réfléchie de la zone à imager. Dans le cas de

FIGURE 1.8 – Illustration de l'effet miroir sur une image échographique. Ici, le diaphragme apparaît en double. Image issue de [13].



(a) Modèle (b) Image échographique

FIGURE 1.9 – Exemple des os fémur-fibula, dont le rapprochement empêche l'échographie conventionnelle d'en obtenir les contours précis. Images issues de [14].

deux interfaces fortement réfléchissantes situées proches l'une de l'autre, il est possible qu'aucun angle d'approche ne puisse permettre l'insonification directe des interfaces, empêchant ainsi leur imagerie. Ce problème peut apparaître lors de l'imagerie d'os rapprochés l'un de l'autre, comme on peut le voir ici sur la figure 1.9.

Imagerie quantitative : exemple de l'élastographie

Si l'imagerie quantitative à partir d'ultrasons n'est pas à l'origine d'une production scientifique aussi abondante que pour l'échographie, les enjeux qui y sont associés n'en sont pas moins importants. Le terme imagerie quantitative fait ici référence à la reconstruction de la carte spatiale d'un ou de plusieurs paramètres physiques, qui peuvent être nombreux : la propagation ultrasonore est influencée par la vitesse des ondes de compression v_p et de cisaillement v_s ou de façon plus générale par le tenseur d'anisotropie, la densité, l'atténuation ou la porosité. Ces variables, utilisées ici pour leur lien direct avec la propagation sonore, peuvent à leur tour être transformées en d'autres

variables, comme les coefficients de Lamé λ, μ ou en module de compression K. De plus, ces variables peuvent être dépendantes d'autres variables selon des lois connues, comme nous l'avons évoqué la vitesse de compression ou la densité peuvent être fonctions de la température, ou de la concentration d'un certain fluide dans un autre, selon le problème considéré. Ainsi, l'imagerie quantitative fait figure de solution pour une large catégorie de problèmes. Cependant, l'obtention d'une image quantitative implique souvent un problème d'optimisation, dont la résolution itérative peut être coûteuse à la fois en temps et en ressources matérielles, ce qui contraste avec l'imagerie temps réel et peu onéreuse associée à l'échographie.

L'élastographie est sans doute l'application d'imagerie ultrasonore quantitative la plus répandue. Contrairement à la plupart des autres méthodes, elle n'implique pas nécessairement un problème d'optimisation, et est parfois compatible avec le temps réel. L'objectif associé est de déterminer le module de Young d'une zone donnée, dont une valeur localement élevée peut révéler la présence d'une tumeur. Dans le cas d'un milieu quasi fluide comme le corps humain, le module de Young est directement relié à la vitesse de cisaillement v_s par la relation $E = 2\rho v_s^2(1+\nu)$, avec le coefficient de Poisson ν qui est très proche de 0.5. L'obtention d'une carte de cette vitesse v_s résout alors le problème. Il est intéressant de remarquer que pour ce type de milieu le module de Young affecte uniquement les ondes de cisaillement et non les ondes de compression. En effet, le coefficient de Poisson, que l'on peut exprimer en fonction de v_p et v_s comme suit : $\nu = \frac{v_p^2 - 2v_s^2}{2(v_p^2 - v_s^2)}$, reste très proche de 0.5 car on a toujours $v_p >> v_s$. Ainsi, l'approximation d'une vitesse de compression constante dans le milieu reste une hypothèse acceptable. Il existe diverses méthodes ultrasonores résolvant ce problème. La première d'entre elles, l'élastographie statique, consiste en l'application d'une contrainte σ avec un faisceau ultrasonore et la mesure de la déformation ϵ permet de retrouver le module de Young grâce à la loi de Hooke $\sigma = E.\epsilon$. Elle reste limitée par la difficulté du contrôle de la contrainte appliquée et sa trop grande dépendance aux conditions aux limites. D'autres méthodes, dites dynamiques, se basent sur la force de radiation acoustique, qui exprime l'utilisation de l'onde de pression pour générer un déplacement dans le tissu. La méthode ARFI (Acoustic Radiation Force Impulse) génère un déplacement transitoire en un point donné et mesure ce déplacement au moyen du speckle tracking, dont est déduit une mesure locale du temps de relaxation qui dépend de l'élasticité. Une technique similaire en régime harmonique, la vibro-acoustographie, utilise deux ondes harmoniques de fréquences proches ω et $\omega + \Delta\omega$ et les focalise en un point donné afin de faire vibrer cette zone. L'écoute des vibrations engendrées par ce déplacement au moyen d'un hydrophone externe permet alors de déduire le module de Young. Ces deux techniques restent dépendantes de paramètres locaux non maîtrisés, tel que la géométrie précise de la zone qui vibre ou de l'absorption acoustique, et requièrent un balayage complet du milieu pour obtenir une image. Une nouvelle technique, apparue en 2004 [15], vise à directement imager les ondes de cisaillement avec le speckle tracking. La principale difficulté de cette méthode réside dans la génération d'une onde de cisaillement dont les caractéristiques sont contrôlées. La Supersonic Shear Wave Imaging technique développée par l'institut Langevin [16] utilise la force de radiation ultrasonore en focalisant des ultrasons à différentes profondeurs, afin de générer plusieurs sources de déplacement dans le milieu et dont les interférences constructives engendrent un cône de Mach. Ainsi, une onde de cisaillement 3D cylindrique et de grande amplitude se propage dans le milieu, et est vue à 2 dimensions comme deux ondes planes divergentes. Cette onde est assez énergétique pour traverser l'ensemble du milieu à imager. Grâce à l'observation de cette onde par le speckle tracking, la vitesse de cisaillement peut être retrouvée en corrélant les images, à partir d'un algorithme de temps de vol, se basant sur l'inversion directe de l'équation d'onde avec une formule du type :

$$\mu(x,z) = \frac{\rho}{N}\sum_{i=1}^{N} \frac{\left(\frac{\partial^2 u_z(x,z)}{\partial t^2}\right)_{x,z}}{\left(\frac{\partial^2 u_z(x,z)}{\partial x^2} + \frac{\partial^2 u_z(x,z)}{\partial z^2}\right)_{t=iT}} \tag{1.7}$$

Les termes dérivés $\frac{\partial^2 u_z(x,z)}{\partial t^2}$, $\frac{\partial^2 u_z(x,z)}{\partial x^2}_{t=iT}$ et $\frac{\partial^2 u_z(x,z)}{\partial z^2}_{t=iT}$ peuvent être obtenus très rapidement par différences finies, ce qui rend cette méthode compatible avec le temps réel. Le résultat obtenu est

alors assez précis, sauf sur l'axe de focalisation des ondes ultrasonores où est appliquée la force de radiation. Une alternative aux algorithmes de type temps de vol, basée sur l'inversion de la forme d'onde complète (FWI) que nous allons détailler dans le chapitre 3, a montré de très bons résultats [17], et résiste mieux à la présence de bruit.

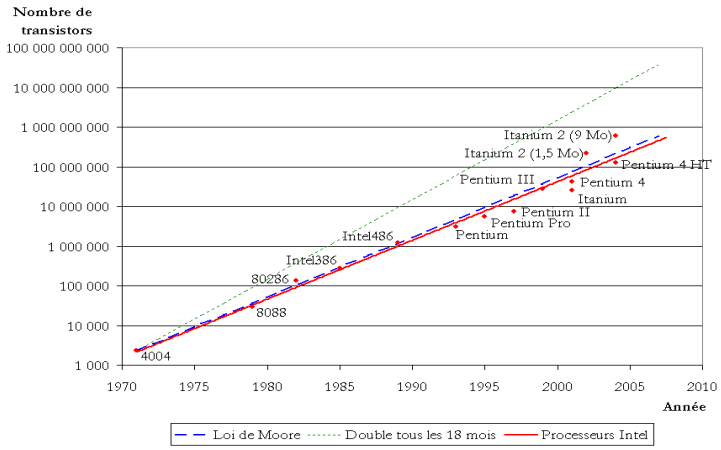
1.2 Introduction au calcul scientifique et au GPGPU

L'apparition de l'informatique et sa démocratisation croissante depuis la deuxième moitié du XXe siècle constitue sans doute le changement le plus important dans la démarche scientifique et ses applications industrielles. La faculté des ordinateurs à effectuer un très grand nombre d'opérations selon des instructions prédéfinies est en effet un atout majeur pour résoudre une multitude de problèmes formulés en langage scientifique. Si ses frontières ne sont pas immédiates à définir, le calcul scientifique désigne en général l'utilisation des ordinateurs pour résoudre un problème mathématique qui ne possède le plus souvent pas de solution analytique exacte, mais dont la solution peut s'exprimer au moyen d'un certain nombre d'équations à résoudre. Ces problèmes concernent en général la simulation, la compréhension ou le contrôle d'un système gouverné par les lois de la physique, et peuvent relever de la physique elle-même mais aussi de la biologie, de la chimie ou des sciences de l'ingénieur. La qualité et la précision de la solution obtenue est naturellement liée à la puissance de calcul disponible. Le plus souvent, une grandeur physique définie continûment dans la réalité est discrétisée en N morceaux, et le temps de résolution du problème est directement proportionnel à la finesse de cette discrétisation. Dans le cas de problèmes intrinsèquement discrets comme on peut les trouver en biologie moléculaire ou en mécanique quantique, la simulation du comportement d'un grand nombre de particules est souvent désirée et une fois encore une grande puissance de calcul est importante. Pour tous ces problèmes, un compromis entre qualité de la solution et temps de calcul associé est nécessaire. Par ailleurs, l'augmentation de la fidélité de la modélisation passe aussi par la prise en compte d'un phénomène observé expérimentalement et dont on va rendre compte par l'addition d'équations supplémentaires, ce qui contribue à l'augmentation du nombre d'opérations nécessaires à la résolution du problème. Ainsi donc, une puissance de calcul supérieure permet de résoudre plus fidèlement un problème donné, ou peut rendre accessible une méthode considérée comme trop longue auparavant, ce qui justifie l'intérêt de son augmentation continue.

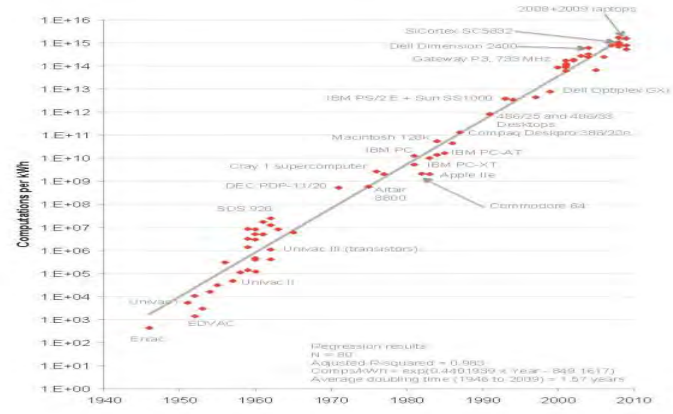
Évolution de la puissance de calcul des processeurs

Depuis le début des années 70, la puissance de calcul des processeurs, ou CPU (Central Processing Unit) n'a cessé de s'accroître. Très rapidement, plusieurs conjectures ont été formulées pour apprécier cet accroissement exponentiel. La plus célèbre d'entre elles, la loi de Moore [1.10(a)], qui a été énoncée en 1965 puis corrigée en 1975, prédit un doublement du nombre de transistors sur une puce de silicium tous les 2 ans. Cette conjecture s'est révélée exacte depuis 1971 jusqu'à ces dernières années (2016), même si l'industrie des semi-conducteurs reconnaît que cette cadence ne peut plus être maintenue à présent. Pour mieux appréhender ce qu'implique réellement cette loi en termes de performances, on peut exprimer le temps d'exécution d'un programme de la façon suivante T = N × CHPI / F où N est le nombre d'instructions du programme, CHPI représente le nombre de Cycles d'Horloge Par Instruction, et F est la fréquence du processeur, qui est inversement proportionnelle au nombre de cycles d'horloge.

L'augmentation du nombre de transistors a notamment permis la diminution des CHPI, en ajoutant de l'intelligence dans la gestion des différentes instructions à exécuter. Les microprocesseurs ont introduit le parallélisme du jeu d'instruction, qui permet d'exécuter certaines instructions indépendantes en simultané, ou d'en modifier d'ordre si les valeurs des quantités à additionner ou multiplier sont immédiatement accessibles. Pour la plupart des processeurs actuels, il est possible d'exécuter 4 Floating-Point OPerations (FLOP) lors d'un même cycle d'horloge sur le cœur de calcul d'un CPU. Les modèles dédiés au calcul comme les Intel Xeon Phi atteignent



(a) Loi de Moore



(b) Loi de Koomey

FIGURE 1.10 – Principales lois de l'informatique. Images issues de [18] et de [19].

jusqu'à 32 FLOP par cœur. Des algorithmes de prédiction des instructions à opérer peuvent être utilisés pour augmenter le nombre d'opérations exécutées en parallèle. De même, un grand nombre de transistors du processeur sont associés à la gestion de la mémoire cache et à la prédiction des données à utiliser, car l'accès à la mémoire RAM nécessite plusieurs centaines de cycles d'horloge et peut être la source d'un ralentissement considérable du programme. Ainsi, l'utilisation de l'augmentation du nombre de transistors T a essentiellement contribué à diminuer le nombre de cycles par instruction, et le gain en performance associé est estimé à $\sqrt{T}$ selon la loi de Pollack. Par ailleurs, grâce à la miniaturisation des transistors, la fréquence des processeurs a également augmenté de façon exponentielle, mais à un rythme inférieur car ces transistors sont des composants gravés par lithographie sur des plaques à deux dimensions, ce qui implique aussi une croissance en $\sqrt{T}$. Cependant, la fréquence d'horloge ne suit plus cette tendance depuis 2004 : la miniaturisation de ces composants a mis à mal la faculté à évacuer la chaleur que dégage ces transistors, et aucune solution n'a permis une réelle augmentation en fréquence, qui stagne depuis autour de 3GHz. De plus, un lien très fort unit fréquence et consommation énergétique : $C \propto U(f)^2 \times f$. Pour des fréquences de l'ordre du GigaHertz, la tension $U(f)$ est une fonction linéaire de la fréquence, ce qui induit une relation cubique entre fréquence d'horloge et consommation du processeur. La consommation électrique des processeurs a ainsi également augmenté continuellement pendant plusieurs décennies. Paradoxalement, l'efficacité énergétique a quant à elle aussi augmenté exponentiellement d'un facteur 1.57 tous les deux ans depuis 60 ans, comme le montre la loi de Koomey illustrée sur la figure 1.10(b).

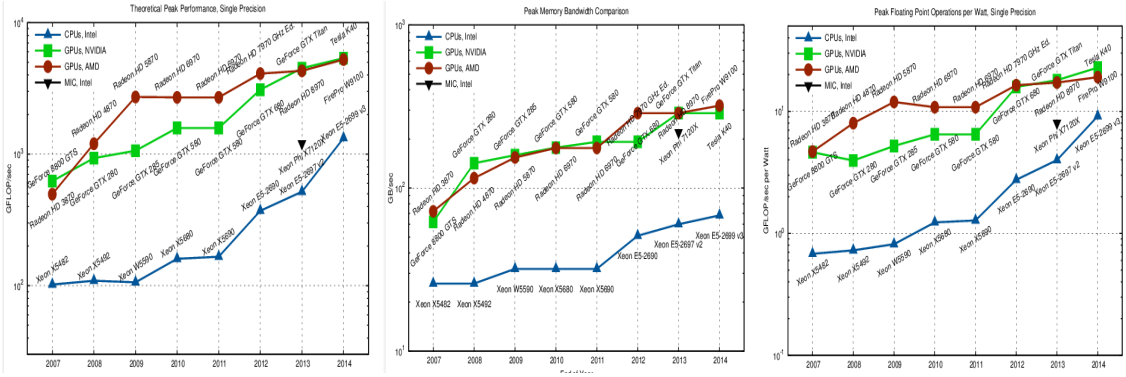
Pour faire face à la limitation en fréquence, les processeurs sont actuellement conçus avec plusieurs cœurs de calcul indépendants. D'un point de vue d'une utilisation grand public, cette approche permet surtout l'exécution simultanée de plusieurs programmes potentiellement exigeants. Elle permet également une véritable accélération des performances pour des programmes capables de se scinder en plusieurs processus, ce qui demande une réécriture de l'algorithme. Néanmoins, pour l'essentiel des applications, cette approche est sans effet sur ces programmes qui sont écrits de façon séquentielle, ce qui explique la multiplication très tempérée du nombre de cœurs des processeurs sur ces dix dernières années.

Ainsi, pendant près de 40 ans, un programme donné pouvait sans aucune modification nécessaire bénéficier d'une accélération significative en utilisant un processeur de nouvelle génération. Le changement de paradigme imposé par la programmation parallèle requiert quant à lui une réécriture au moins partielle d'un programme, en suivant des règles plus exigeantes, ce qui peut expliquer la relative inertie à la transcription de ces programmes, y compris dans le domaine scientifique. Cet effort de transition a toutefois été adopté par plusieurs solutions commerciales à but scientifique, et en particulier celles impliquant des modèles éléments finis, coûteux en temps de calcul. Le gain en temps d'exécution, restant tributaire du nombre de cœurs disponibles, demeure modéré. C'est pourquoi l'intérêt des scientifiques désireux d'une puissance de calcul plus importante, et tirant davantage parti de l'augmentation exponentielle du nombre de transistors, s'est porté vers un autre composant informatique, le processeur graphique.

General-Purpose computing on Graphics Processing Units (GPGPU)

Les processeurs graphiques, ou GPUs (Graphics Processing Unit) constituent le moteur de calcul des cartes graphiques. L'objectif de ces cartes est d'alléger le CPU des tâches relatives à l'affichage, lequel a beaucoup évolué depuis les années 80. Initialement, il se résumait à l affichage de caractères ASCII, avant de pouvoir contrôler l'allumage de chacun des pixels de l'écran. Avec l'apparition de la couleur, codée sur un nombre croissant de niveaux au cours du temps, l'augmentation de la résolution écran, et enfin la gestion autonome de la représentation d'objets à 3D, la tâche de l'affichage s'est rapidement complexifiée. Ceci s'est traduit par une évolution hardware suivant le modèle Single Instruction Multiple Data (SIMD) conduisant en particulier à l'augmentation du nombre d'unités logiques de calcul et de la bande passante mémoire. De plus, la complexification progressive du jeu d'instructions à exécuter a également influé sur l'architecture matérielle, se rapprochant un peu plus de celle d'un CPU avec l'apparition de plusieurs niveaux de mémoire cache. Cette modification hardware a également permis une évolution très importante côté software. Du côté de l'affichage, les fonctionnalités proposées par l'API OpenGL se sont assouplies, permettant au programmeur de définir lui-même le mouvement et la modification structurelle des objets à 3D à représenter. Ces possibilités de programmation ont rapidement été exploitées pour des objectifs autres que l'affichage écran : au début des années 2000, le terme General-Purpose computing on GPU (GPGPU), a été inventé par Mark Harris pour désigner cette pratique. En 2007 puis en 2009, l'apparition des plate-formes CUDA, développée par le fabriquant de GPU Nvidia, et OpenCL ont grandement contribué à faciliter l'écriture d'un code scientifique utilisant le GPU comme outil de calcul. De nombreuses publications faisant état des gains en performance [20, 21] sont alors parues, et le GPU s'est depuis positionné comme un acteur incontournable du calcul haute-performance. Depuis 10 ans, l'écart de performance entre CPU et GPU n'a cessé de s'accroître, comme on peut le voir sur la figure 1.11(a). On peut l'expliquer par la demande du principal acteur de ce domaine, le secteur des jeux vidéos, qui souhaite obtenir un rendu toujours plus réaliste et s'adapter aux normes de haute définition des écrans, ce qui passe inéluctablement par une augmentation du nombre d'opérations à effectuer par seconde. Du point de vue du composant, cette accélération est due à une augmentation drastique du nombre d'unités logiques de calcul présentes sur le GPU : les GPU actuels (2016) en contiennent plusieurs milliers. Une autre caractéristique non moins importante à observer est la bande passante du composant, qui va conditionner la vitesse à laquelle les données vont être acheminées vers l'unité logique de calcul, et est le principal frein à l'exploitation de la puissance de calcul du matériel. Elle affecte en

particulier les problèmes qui nécessitent l'accès à de larges tableaux, ce qui inclut la plupart des situations de simulation numérique, dépendantes de plusieurs variables d'entrée. Une fois encore, l'avantage est donné au GPU comme montré figure [1.11(b)], ce qui était prévisible dans l'optique de l'exploitation du très grand nombre d'unités de calcul qui le constitue. Enfin, la performance énergétique entre CPU et GPU, mesurée en nombre d'opérations flottantes par watt est aussi à la faveur des GPU [1.11(c)]. La fréquence d'horloge en moyenne trois fois moins élevée pour les GPU explique en grande partie ce phénomène, du fait de la dépendance cubique entre fréquence et consommation.



(a) Performance de calcul. (b) Bande passante. (c) Performance énergétique.

FIGURE 1.11 – Éléments de comparaison entre CPU et GPU. Images issues de [22]

Un autre avantage significatif des GPU est qu'en plus d'offrir une puissance de calcul très supérieure, d'environ 2 ordres de grandeur actuellement, le coût de ces deux composants est équivalent, pour des produits d'entrée, milieu ou haut de gamme. C'est sans doute cette caractéristique a priori surprenante qui est à l'origine du changement progressif de pratique dans le calcul scientifique. Et si de plus en plus de superordinateurs incorporent des GPUs, un gain substantiel peut être également obtenu pour un scientifique faisant l'acquisition d'une nouvelle carte graphique pour son ordinateur de travail : pour obtenir un gain du même ordre avec des CPUs, les factures matérielles et énergétiques seraient bien plus élevées. Il est intéressant d'observer que si actuellement la performance d'un GPU outrepasse de loin celle d'un CPU, le nombre de transistors utilisé pour les deux composants est sensiblement le même, ce qui explique la similarité du coût des deux composants. Pour un CPU, une augmentation du nombre de transistors est surtout mise à profit pour diminuer le nombre de cycles d'horloge par instruction, alors qu'elle est utilisée pour multiplier le nombre d'unités de calcul pour un GPU. Comme nous l'avons vu, la première stratégie a atteint ses limites et alors que la seconde continue de bénéficier linéairement de l'augmentation des transistors. De plus, l'architecture plus complexe des GPU requiert souvent plus de temps pour s'adapter à une nouvelle finesse de gravure, les GPU actuels (2016) étant gravés en 28nm alors que ceux des CPU le sont en 14nm. Les GPUs ont ainsi plus de potentiel quant aux limites de la gravure lithographique. Enfin, la stratégie de la multiplication du nombre de d'unités de calcul n'entraîne qu'une dépendance linéaire entre augmentation de la puissance de calcul et consommation électrique, et qui est de plus sous linéaire d'une génération à l'autre grâce à la miniaturisation des transistors. Le modèle GPU est donc tout à fait pertinent du point de vue énergétique.

Ainsi, la parallélisation d'un code apparaît comme une étape inévitable pour continuer à bénéficier du progrès informatique à chaque nouvelle génération de composants. Il demeure cependant quelques restrictions à son usage :

— Le défaut le plus pénalisant est incontestablement la nécessité de devoir coder dans un langage dédié, mais aussi avec un algorithme repensé. De fait, de par l'architecture parallèle des GPU, le calcul sur carte graphique utilise un modèle de programmation spécifique et son apprentissage ne relève pas simplement de la connaissance d'un nouveau langage.

Pour une programmation efficace, il est primordial que l'utilisateur ait conscience de la structure, des limitations et des propriétés du composant qu'il utilise et de leur évolution. Cet investissement d'apprentissage est important du côté des utilisateurs, et de plus le paradigme de programmation plus complexe oblige à consacrer plus de temps que pour une version séquentielle ou parallèle sur CPU. Enfin, pour des problèmes déjà implémentés sur CPU, tout le travail est à refaire.

— La parallélisation du code n'est pas toujours possible. Idéalement, le problème à résoudre doit suivre le modèle SIMD, ce qui constitue la plupart des cas.

— Le nombre d'instructions par donnée chargée doit aussi être suffisamment élevé pour que le temps de charge limité par la bande passante puisse être compensé. Le calcul d'un produit scalaire effectuant une seule addition pour deux données chargées n'offrira pas de bonnes performances. En revanche, pour un produit matriciel $N \times N$, chaque donnée est utilisée N fois, ce qui permet une utilisation totale des unités de calcul si N dépasse une certaine valeur. Un problème de grande taille est par ailleurs toujours souhaitable : si le problème est scindé en un nombre de processus supérieur au nombre d'unités logiques, le GPU peut compenser le temps de chargement de certaines données par des calculs sur les processus dont les données ont déjà été chargées. Cette situation reste la plus courante.

— La performance s'applique surtout à la simple précision, car c'est elle qui est utilisée en quantité par les jeux vidéos, ce qui incite les fabricants de GPU à augmenter ce type d'unité logique. La dernière génération de GPU Nvidia, Maxwell, a fait le sacrifice sur les unités de calcul en double précision : la performance théorique maximale passe de 7 TFLOPS en simple précision à 200 GFLOPS en double précision. Sur CPU, l'écart entre les deux précisions est habituellement d'un demi. Remarquons toutefois que si la double précision est indispensable dans certains cas, elle ne l'est pas le plus souvent, mais est un confort qui évite au programmeur de trop se soucier des contraintes de l'arithmétique finie. En effet, beaucoup de problèmes liés à la précision numérique ont une alternative algorithmique.

Chapitre 2

Le procédé d'imagerie FTIM

La méthode Fast Topological IMaging, ou imagerie topologique rapide, est une méthode d'imagerie ultrasonore de type échographique qui est l'aboutissement de plusieurs années de recherche au laboratoire PHASE. Comme nous allons le voir, cette méthode présente plusieurs avantages sur les autres techniques échographiques, et a prouvé son efficacité avec des données expérimentales.

Tout d'abord, nous allons décrire le formalisme mathématique qui est associé à son analogue temporel, la méthode de l'énergie topologique dans le domaine temporel, à laquelle nous référerons avec son acronyme anglais TDTE (Time Domain Topological Energy). Nous évoquerons l'interprétation physique de TDTE qui légitime sa création, puis nous introduirons la méthode FTIM. Nous pourrons alors la comparer théoriquement avec les méthodes vues dans le chapitre précédent, et discuter de son application à un milieu initial considéré comme homogène et semi-infini, comme c'est le cas pour les autres techniques échographiques. Ensuite, nous verrons que la relative simplicité de l'algorithme FTIM constitue pratiquement un cas d'école quant à l'illustration des critères déterminants pour une implémentation efficace sur GPU dans le cas à 2D, et le cas à 3D sera également présenté. Une application expérimentale de la méthode permettant une visualisation en temps réel sera exposée, qui nous permettra de conclure sur la valeur ajoutée de cette nouvelle implémentation.

2.1 Principe

Origine théorique

Les premiers travaux concernant l'imagerie topologique ultrasonore ont été initiés par la thèse de Nicolas Dominguez [23], soutenue en 2006. Son objectif était alors d'obtenir une image de la porosité dans les milieux solides composites, dans des perspectives d'application au contrôle non destructif. Pour ce faire, son attention s'est portée sur les méthodes de l'optimisation de forme, qui visent à déterminer les contours optimaux d'un milieu par rapport à un critère donné, comme la minimisation d'un coefficient de traînée ou la maximisation de la dissipation de la chaleur d'une pièce. L'optimisation topologique a quant à elle une portée encore plus générale, elle modifie non seulement la forme des contours du milieu, mais introduit aussi des degrés de liberté supplémentaires en autorisant l'insertion de variations géométriques (trous, fissures) dans ce milieu. Dans son contexte général, la formulation mathématique de l'optimisation topologique consiste en la minimisation d'une fonction coût. Comme beaucoup de problèmes d'optimisation, sa résolution peut s'effectuer avec une méthode de type gradient, ce qui implique le calcul des dérivées par rapport aux variables du problème, en l'occurrence des modifications de géométrie. Contrairement à la plupart des grandeurs physiques qui sont définies continûment dans l'espace, et qui se calculent selon les définitions usuelles de Fréchet ou de Gâteaux, la dérivation par rapport à la présence d'un trou de taille infinitésimale requiert une formulation spécifique de la dérivée, dite dérivée topologique. Elle est définie à partir d'un développement asymptotique de la fonction

coût Φ au point $\tilde{\mathbf{x}}$, comme suit :

$$\Phi(\Psi_\epsilon(\tilde{\mathbf{x}})) = \Phi(\Psi) + f(\epsilon)g(\tilde{\mathbf{x}}) + o(f(\epsilon)) \tag{2.1}$$

Le terme Ψ est la fonction caractérisant la topologie du milieu, ϵ le rayon d'un trou circulaire dans la géométrie, f est une fonction positive du premier ordre et $g(\tilde{\mathbf{x}})$ correspond à la dérivée topologique. On peut l'interpréter physiquement comme la vraisemblance de la présence d'un trou. Cette définition de la dérivée topologique est issue des travaux de Schumacher [24], qui ont été repris par la suite par Solokowski [25].

Pour la problématique de la porosité, l'objectif est alors de pouvoir déterminer la position des trous en fonction du signal ultrasonore $U(\mathbf{x}_i, t)_{obs}$ reçu, lequel a été influencé par leur présence lors de la propagation. La fonction coût associée est de la forme :

$$\begin{cases} \Phi(U(\Omega)) = \dfrac{1}{2}\displaystyle\sum_{i=1}^{N_{elem}} \int_0^T ||U(\mathbf{x}_i, t)_{obs} - U(\mathbf{x}_i, t; \Omega)_{syn}||^2 \mathrm{dt} \\ \displaystyle\int_{S(\Omega)} (\Delta U(\mathbf{x}, t) - \frac{1}{c^2}\frac{\partial^2 U(\mathbf{x}, t)}{\partial t^2} - \sum_{i=1}^{N_{elem}} f(\mathbf{x}_i, t))\mathrm{d}\mathbf{x} = 0 \ \forall t \in [0, T] \end{cases} \tag{2.2}$$

avec U le champ de pression acoustique, N_{elem} le nombre de récepteurs, T la durée de l'enregistrement des données, Ω la topologie du milieu et $f(\mathbf{x}_i, t)$ le terme source du i$^{\text{ème}}$ émetteur.

Le couplage de ces deux équations décrit un problème d'optimisation contraint par une équation aux dérivées partielles. La première équation décrit la fonction coût, qui tend à minimiser l'écart entre entre les données ultrasonores simulées $U(t, \mathbf{x}_i; \Omega)_{syn}$ et observées $U(t, \mathbf{x}_i)_{obs}$. La seconde équation, exprimée sous la forme faible pour incorporer naturellement la topologie Ω, décrit le comportement du champ de pression $U(\mathbf{x}, t)$, qui suit l'équation d'onde à l'intérieur du domaine S de topologie Ω, qui est la variable que l'on souhaite déterminer dans le problème d'optimisation.

Le calcul pratique du gradient topologique peut s'effectuer de plusieurs façons. La méthode de l'adjoint, qui sera détaillée dans la partie 3.1, est utilisée ici. Dans le cas de l'élastodynamique transitoire, Nicolas Dominguez a déterminé les expressions du gradient topologique en fonction du type de défaut [26], qui sont récapitulées dans le tableau 2.1. Parallèlement à ces travaux, Marc Bonnet a également travaillé sur ces problèmes dans les domaines élastique et acoustique [27, 28, 29].

Type de défaut	**Gradient topologique associé**
2D Dirichlet	$\int_0^T \Big(\frac{2\mu(\mu+\eta)}{2\mu+\eta} U(\mathbf{x},t)V(\mathbf{x},T-t)\Big)\mathrm{dt}$
3D Dirichlet	$\int_0^T \Big(\frac{3\mu(\lambda+2\mu)}{2\lambda+5\mu} U(\mathbf{x},t)V(\mathbf{x},T-t)\Big)\mathrm{dt}$
2D Neumann	$\int_0^T (-\frac{(\mu+\eta)}{2\mu\eta}(4\mu\sigma(U(\mathbf{x},t)) :$ $\epsilon(V(\mathbf{x},T-t)) + (\eta-2\mu)tr(\sigma(U(\mathbf{x},t))tr\epsilon(V(\mathbf{x},T-t)) - \rho\partial_t U(\mathbf{x},t)\partial_t V(\mathbf{x},T-t)\Big)\mathrm{dt}$
3D Neumann	$\int_0^T (-\frac{3\pi(\lambda+2\mu)}{9\lambda+14\mu}(20\mu\sigma(U(\mathbf{x},t)) :$ $\epsilon(V(\mathbf{x},T-t)) + (3\lambda-2\mu)tr(\sigma(U(\mathbf{x},t))tr\epsilon(V(\mathbf{x},T-t)) - \rho\partial_t U(\mathbf{x},t)\partial_t V(\mathbf{x},T-t)\Big)\mathrm{dt}$

TABLE 2.1 – Tableau récapitulant les expressions du gradient topologique selon les cas de figure. Plus de détails sont disponibles dans [26].

Le champ $U(\mathbf{x}, t)$ est le même que celui décrit dans l' Eq. 2.2. La méthode de l'adjoint a introduit un deuxième champ $V(\mathbf{x}, t)$, dit champ adjoint, qui se définit comme suit :

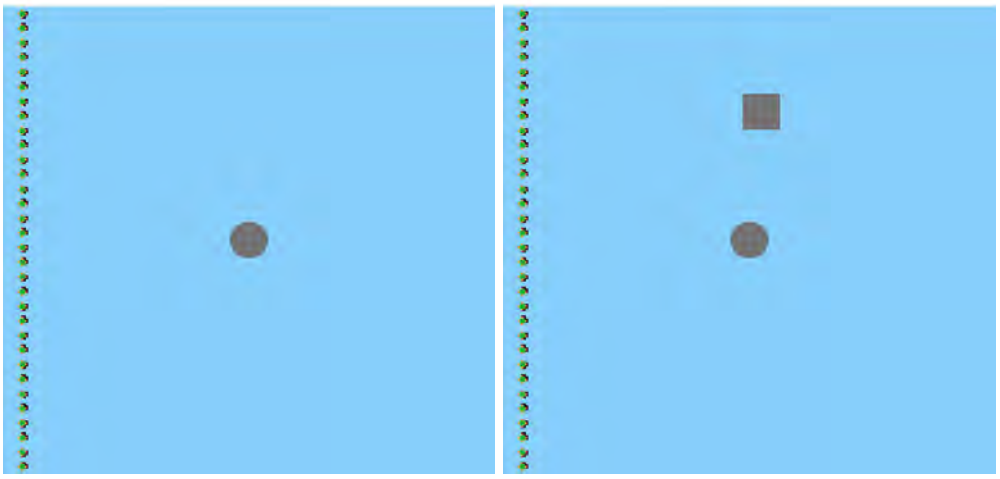
$$\Delta V(\mathbf{x}, t) - \frac{1}{c^2}\frac{\partial^2 V(\mathbf{x}, t)}{\partial t^2} = \sum_{i=1}^{N_{elem}} \left(U_{obs}(\mathbf{x}_i, T-t) - U_{syn}(\mathbf{x}_i, T-t)\right) \ \forall (\mathbf{x}, t) \in S \times [0, T] \tag{2.3}$$

Après plusieurs expérimentations numériques, et le constat que le résultat obtenu n'évoluait que très peu au fil des itérations, une quantité similaire, l'énergie topologique, a été définie :

$$ET(\mathbf{x}) = \int_0^T U^2(\mathbf{x}, t)V^2(\mathbf{x}, T-t)dt \tag{2.4}$$

Cette formule, qui ne fait désormais apparaître plus que les champs direct et adjoint, permet l'imagerie des contrastes d'impédance en une seule itération. Bien que non justifiée mathématiquement, c'est avant tout son interprétation physique qui rend particulièrement pertinente son utilisation pour l'imagerie des hétérogénéités.

Interprétation dans le domaine temporel



(a) Modèle initial (b) Milieu réel

FIGURE 2.1 – Modèles de propagation.

L'énergie topologique peut être vue comme la corrélation temporelle entre deux champs acoustiques numériques, le champ direct et le champ adjoint, dont la signification par rapport à la réalité expérimentale est la suivante :

- Le champ direct, qui a le même terme source que celui utilisé dans la réalité, peut être vu comme la version numérique de la propagation de l'onde émise dans l'expérience réelle, à la seule différence de la non prise en compte des réflexions sur les hétérogénéités dont la présence est inconnue, et qui ne sont donc pas modélisées dans la version numérique illustrée sur la figure 2.1(a) du milieu de propagation.
- Le champ adjoint est quant à lui la rétropropagation de la différence entre données observées et données synthétiques. Les données synthétiques correspondent à la simulation numérique de l'enregistrement des transducteurs, et contiennent donc l'ensemble des échos sur des structures modélisées dans le milieu numérique, mais aussi les ondes transmises d'un transducteur à un autre. Alors, la différence entre les données observées, qui contiennent toute l'information, et les données synthétiques, qui contiennent de l'information sur ce qui est déjà connu, permet d'obtenir la signature des échos qui n'étaient pas prévisibles. Ainsi, le champ adjoint, qui a pour terme source cette différence, peut être vu comme la propagation rétrograde des ondes réfléchies par les hétérogénéités inconnues du milieu.

Pour mieux comprendre l'efficacité de cette méthode, nous pouvons considérer l'expérience numérique suivante : un milieu réel homogène avec deux structures, l'une circulaire et l'autre carrée (figure 2.1(b)). Nous supposons connaître la présence et la position de la structure circulaire, qui sera donc modélisée dans notre milieu numérique (figure 2.1(a)).

Les propagations de l'onde réelle (qui est en fait simulée dans cette expérience numérique), et de l'onde simulée sont illustrées sur les figures 2.3(a),2.3(e),2.3(i),2.3(m) et 2.3(b),2.3(f),2.3(j),2.3(n) respectivement. La construction de la source adjointe est illustrée sur la figure 2.2, on peut observer sur les données synthétiques sur la figure 2.2(a) la signature des fronts d'ondes émis par les autres transducteurs, ainsi que l'écho de la structure circulaire. Par différence, la source adjointe contient deux échos, l'un dû à la réflexion directe de l'onde émise par les transducteurs, l'autre dû à la réflexion de l'onde réfléchie par la structure circulaire.

Les figures 2.3(d),2.3(h),2.3(l)2.3(p) illustrent la construction de l'image, qui résulte de la corrélation entre le champ direct illustré sur les figures 2.3(b),2.3(f),2.3(j),2.3(n) et le champ adjoint illustré sur les figures 2.3(c),2.3(g),2.3(k),2.3(o). Les deux champs se croisent deux fois, au moment du passage de l'onde directe et de l'onde réfléchie par la structure circulaire. A ces instants, la valeur du produit $U(t)^2V(T-t)^2$ est localement très élevée aux points de croisement de ces deux champs, ce qui contribue grandement à la construction de l'image. Notons par ailleurs qu'au moment du croisement des champs, le champ adjoint se trouve être focalisé, ce qui exprime bien qu'il est la conséquence d'une réflexion sur cette structure, et ce qui se traduit donc par une valeur localement très élevée de $V(T-t)$. La combinaison de ces deux propriétés, l'une exprimant la coïncidence spatio-temporelle, l'autre la focalisation, motive l'expression de l'énergie topologique. L'élévation au carré de chacun des deux champs assure une valeur positive aux contributions locales afin que celles-ci n'interfèrent pas, et diminue l'impact des contributions trop faibles et potentiellement bruitées, et permet aussi de gagner en lisibilité sur l'image obtenue.

Lors d'une expérience réelle, certains paramètres sont moins bien maîtrisés. D'une part, au cours de l'émission de l'onde, l'enregistrement simultané est impossible, et les fronts d'ondes reçus des autres transducteurs sont de trop grande amplitude pour pouvoir être enregistrés fidèlement. Une technique consiste alors à mettre à zéro cette portion dans la partie correspondante du signal du champ adjoint. Une autre source d'erreur peut être, dans notre exemple, une différence entre l'écho réel et l'écho simulé de la structure circulaire, tout d'abord de temps de vol à cause d'une potentielle erreur sur la vitesse de fond, mais surtout d'amplitude à cause d'une mauvaise maîtrise des propriétés d'atténuation. Ici encore, la mise à zéro de la portion du signal correspondant résout le problème, et est sans impact sur la qualité de l'image car la structure circulaire est déjà connue et n'a pas besoin d'être imagée.

Ainsi, la version temporelle de la méthode s'interprète comme une corrélation de deux champs, qui révèle les zones échogènes. Cette faculté de réflexion n'est pas seulement valable pour des petites hétérogénéités, mais aussi pour des interfaces ou pour tout contraste d'impédance, ce qui a motivé la thèse de Perrine Sahuguet [30], soutenue en 2012, qui a pu étendre à l'imagerie médicale l'utilisation de l'énergie topologique, en montrant expérimentalement la possibilité d'imager de faibles contrastes d'impédance. De fait, l'efficacité de la méthode n'est, en dépit de son nom, pas liée à son origine topologique. Nous verrons dans la partie 3.1 que la méthode de l'adjoint permet d'obtenir des expressions très similaires en utilisant d'autres grandeurs physiques.

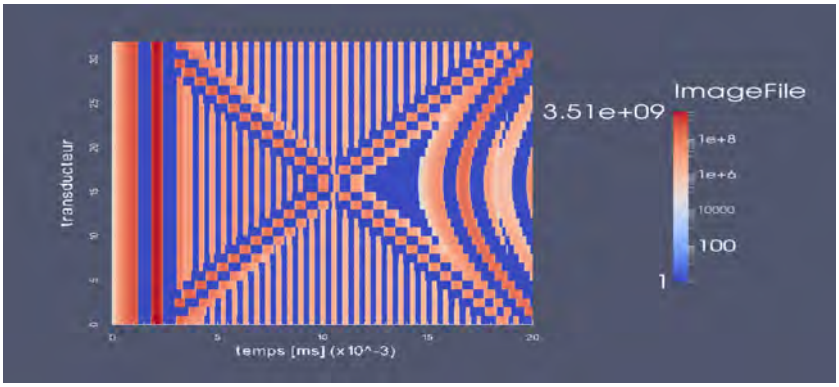
Transposition dans le domaine fréquentiel

Développement théorique

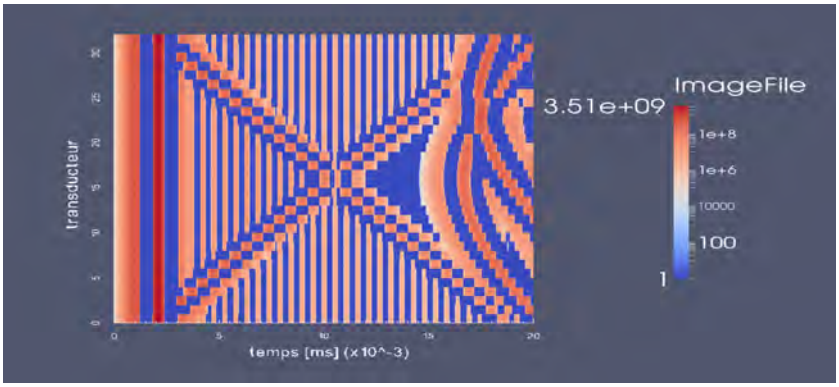
En 2012, Samuel Rodriguez et al [31] ont reformulé le problème dans le domaine fréquentiel. L'expression du gradient topologique est simplifiée d'une façon similaire à TDTE en ne laissant apparaître que l'expression des champs U et V, ce qui permet une interprétation physique identique :

$$G(\mathbf{x}) = \int_0^T U(\mathbf{x},t)V(\mathbf{x},T-t)dt \tag{2.5}$$

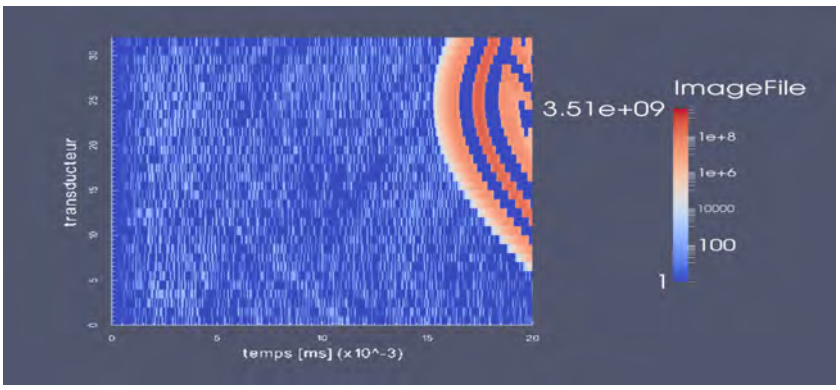
Grâce à une utilisation judicieuse de la propriété d'isométrie de la transformation de Fourier, il



(a) Bscan simulé dans le milieu de référence (modèle initial avec une inclusion)



(b) Bscan simulé dans le milieu "réel" (modèle avec deux inclusions)



(c) Source adjointe

FIGURE 2.2 – (a) et (b) sont les données enregistrées dans les différents domaines de propagation (réel et synthétique). (c) représente la construction de la source adjointe, qui est la différence entre (a) et (b). Les simulations sont faites avec un code de calcul de type éléments spectraux (code Specfem). Les données sont représentées en échelle logarithmique pour garder une dynamique correcte de couleur entre le front émis et les fronts d'onde réfléchis.

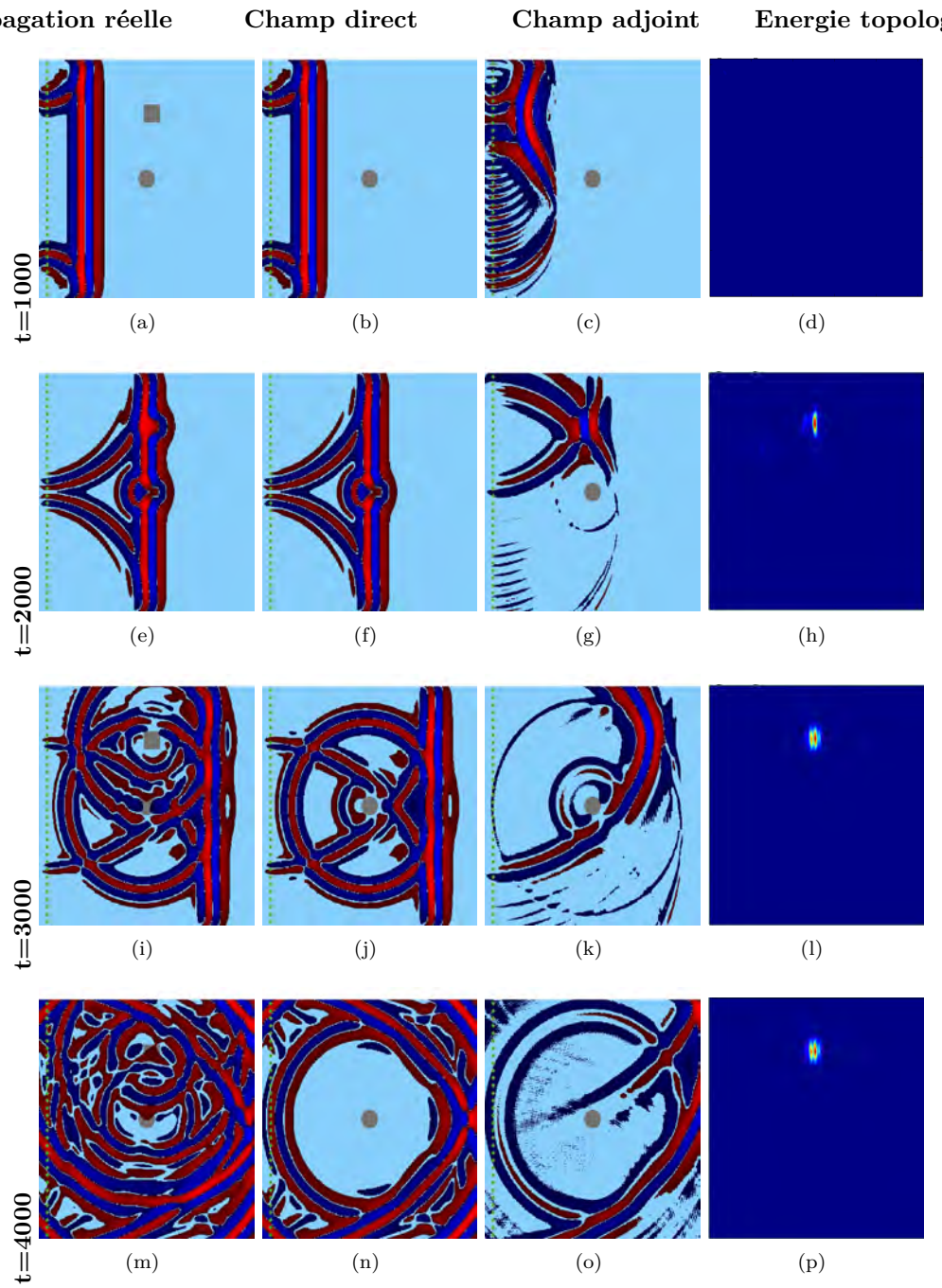
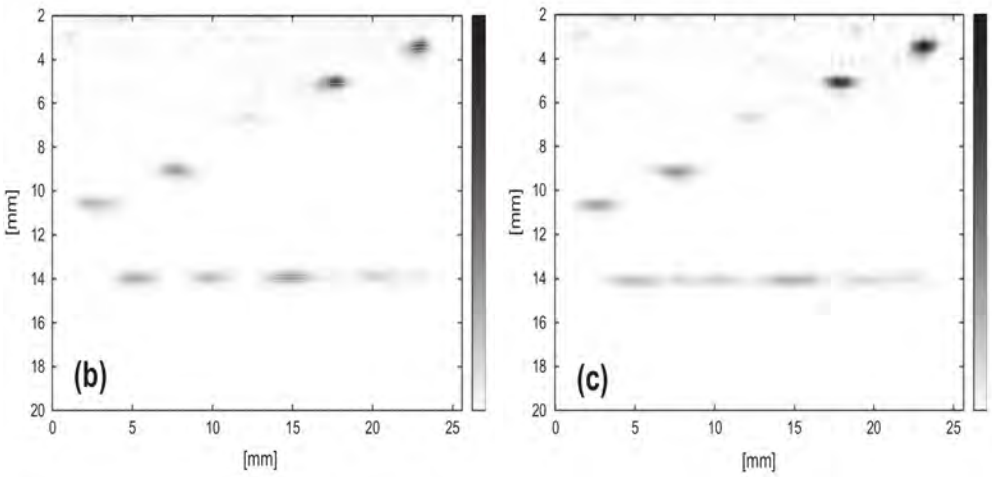


FIGURE 2.3 – Illustration de la construction de l'énergie topologique.



(a) TDTE (b) FTIM

FIGURE 2.4 – En prenant le module de la transformée de Hilbert de l'image, les oscillations présentes dans TDTE disparaissent dans FTIM. Le minimum et maximum du code couleur correspondent au minimum et maximum des images

est possible de définir une quantité de même norme dans le domaine fréquentiel. On a :

$$\begin{aligned} F(\mathbf{x}) &= |\int_{\mathbb{R}} U(\mathbf{x}, f)V(\mathbf{x}, f)df| \\ &= |G(\mathbf{x})| \end{aligned} \tag{2.6}$$

Avec au calcul dans le domaine fréquentiel, on obtient facilement l'enveloppe spatiale de ce gradient, en définissant la quantité :

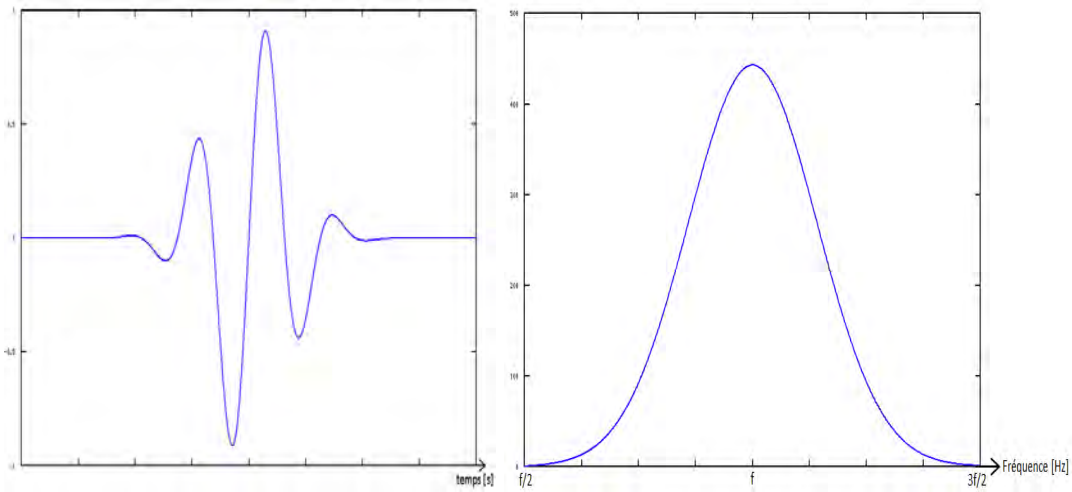
$$\tilde{F}(\mathbf{x}) = |\int_{\mathbb{R}^+} U(\mathbf{x}, f)V(\mathbf{x}, f)df| \tag{2.7}$$

Physiquement, ceci se traduit par l'élimination du phénomène d'oscillations du gradient topologique, inhérent à la propagation d'onde dans le milieu, comme on peut le voir sur l'image 2.4. Cette élimination est semblable à celle effectuée sur les Bscan classiques, ce qui améliore la lisibilité de l'image obtenue.

Le passage dans le domaine fréquentiel permet par ailleurs d'exploiter une particularité des transducteurs piézoélectriques : leur bande passante limitée. Comme nous l'avons vu en introduction, ces transducteurs ont une bande passante équivalente à leur fréquence centrale. Comme l'essentiel de l'énergie est situé dans cette bande de fréquence, et que le signal d'entrée utilisera un spectre similaire comme illustré sur la figure 2.5, il est alors possible de considérer le critère d'imagerie suivant :

$$I_{FTIM}(\mathbf{x}) = |\int_{F_{min}}^{F_{max}} U(\mathbf{x}, f)V(\mathbf{x}, f)df| \tag{2.8}$$

L'intérêt de cette réduction de bande est de limiter le nombre de fréquences discrètes à utiliser pour le calcul de $I_{FTIM}(\mathbf{x})$, ce dont il sera possible de tirer profit par la suite. Idéalement, la réduction à une seule fréquence limiterait au maximum des calculs, mais il ne faut pas oublier l'importance du caractère transitoire du signal émis. En vertu du principe d'incertitude, qui veut qu'une fonction concentrée en temps soit étalée en fréquence et vice-versa, il convient de trouver un compromis entre signal transitoire et signal à bande étroite. En ce sens, l'utilisation d'un signal gaussien qui est l'invariant de la transformée de Fourier, ou d'une de ses variantes, comme l'ondelette de type Ricker, constituent les meilleures options.



(a) Signal temporel émis (b) Tranformée de Fourier du signal

FIGURE 2.5 – Caractéristiques du signal émis, qui présente l'avantage d'être compact à la fois en temps et en fréquence. La compacité fréquentielle permet au signal d'être situé dans la bande passante d'un transducteur piézoélectrique.

Discrétisation

Pour son implémentation numérique, la quantité I_{FTIM} est discrétisée comme suit :

$$\hat{I}_{FTIM}(\mathbf{x}) = |\sum_{i=1}^{N_{freq}} U(\mathbf{x}, f_i)V(\mathbf{x}, f_i)| \tag{2.9}$$

Dans la perspective de limiter le nombre d'opérations sans sacrifier la précision de la méthode, il est alors très important de réfléchir au bon nombre de fréquences nécessaires au calcul de $I_{FTIM}(\mathbf{x})$. Pour cela, deux conditions sont nécessaires :

- Respecter le critère de Shannon, à savoir utiliser une fréquence d'échantillonnage F_e d'au moins $2F_{max}$.
- Utiliser un pas en fréquence inférieur ou égal à F_e/N, N étant le nombre d'échantillons temporels, pour obtenir la meilleure résolution spectrale possible. En pratique, nous choisirons $\delta f = F_e/N$. Remarquons au passage que pour une durée fixe, la résolution spectrale ne dépend pas de la fréquence d'échantillonnage : multiplier par deux la fréquence d'échantillonnage oblige à multiplier par deux le nombre d'échantillons N, et laisse invariant le rapport δf. On peut alors estimer le nombre de fréquences nécessaires :

$$\begin{aligned} N_{freq} &= \frac{F_{max} - F_{min}}{\delta f} + 1 \\ &= N\frac{(F_{max} - F_{min})}{F_e} + 1 \end{aligned} \tag{2.10}$$

Ainsi, le nombre de fréquences nécessaires est directement proportionnel à la durée du signal émis, et à la largeur de la bande passante, que l'on peut résumer à la fréquence centrale. En fonction de la situation, il appartiendra à l'utilisateur de déterminer son besoin en résolution spatiale, modulable par la fréquence centrale, et la profondeur du domaine qu'il souhaite sonder, modulable par la durée d'acquisition.

Le domaine fréquentiel présente le grand avantage de transformer les convolutions en multiplications, ce qui va se révéler être très utile pour calculer les champs acoustiques de façon pratique,

que l'on peut exprimer ainsi :

$$U(\mathbf{x}, f) = \sum_{j=1}^{N_{elem}} H_j(\mathbf{x}, f) S_j(f) \tag{2.11}$$

où $H_j(\mathbf{x}, f)$ est la réponse impulsionnelle du j$^{\text{ème}}$ transducteur dans le milieu et $S_j(f)$ est la transformée de Fourier du signal source émis par ce transducteur.

Le diagramme de rayonnement H de la source représente la réponse impulsionnelle d'une source acoustique dans le milieu de référence. Il peut être calculé de façon semi-analytique pour un milieu simple, ou par une méthode numérique de type différences ou éléments finis pour un milieu complexe. Dans le cadre d'une application où les propriétés acoustiques ou élastiques du milieu sont connues à l'avance, il est possible de précalculer ce diagramme de rayonnement afin de n'avoir à effectuer que la multiplication entre la transformée de Fourier du signal et le diagramme, ce qui représente un nombre très faible d'opérations mathématiques. C'est l'un des avantages de cette méthode, qui favorise aussi son implémentation efficace sur GPU. Par ailleurs, la résolution spatiale du diagramme détermine celle des images calculées. Le nombre d'échantillons temporels N étant dans la pratique très inférieur au nombre de points constituant le diagramme de rayonnement, le coût de la transformée de Fourier, proportionnel à $N \log N$ pour la transformée de Fourier rapide, est négligeable devant le coût de la multiplication signal-diagramme. Remarquons que la taille en mémoire T_{diag} du diagramme de rayonnement est la suivante :

$$T_{diag} = \text{sizeof(complexe)} \times N_{freq} \times N_x \times N_y \ (\times N_z) \tag{2.12}$$

Nous avons vu les conditions à respecter sur le nombre de fréquences. Les quantités N_x, N_y, et N_z dans le cas à 3D représentant les discrétisations axiale et latérale(s) du domaine à calculer. Si rien n'empêche de choisir des valeurs quelconques pour ces constantes vis-à-vis de la stabilité de la méthode, il demeure important de les choisir suffisamment petites afin de pouvoir observer les défauts à imager. Pour un domaine de taille $L_x \times L_y \times L_z$, un signal d'émission de fréquence maximale F_{max}, et un minimum global $c_{min} = \min c(\mathbf{x}),\ \mathbf{x} \in \otimes_{i=1}^{N_{dim}} [0; L_i]$ de la vitesse de compression dans le milieu à imager, le pas spatial selon la dimension i est donc $\Delta x_i = L_i/N_i$, et doit être choisi comme inférieur à la demi longueur d'onde la plus petite dans le milieu, soit $\Delta x_i < c_{min}/2F_{max}$. En combinant ces deux équations, on doit respecter :

$$N_i > \frac{2 L_i F_{max}}{c_{min}} \ \forall i \in [1, N_{dim}] \tag{2.13}$$

Enfin, comme le diagramme est constitué de nombres complexes, sa taille est proportionnelle à la place mémoire d'un nombre complexe sizeof(complexe), qui peut varier selon le type de représentation choisi. On a sizeof(complexe) = 2 ×sizeof(float), où float est la représentation numérique à virgule flottante d'un nombre réel. L'intérêt du représentation en double précision (8 octets) est assez limitée, car sur l'image obtenue, la précision à 15 chiffres significatifs qu'elle offre ne pourra pas être distinguée. La simple précision (4 octets) offre 7 chiffres significatifs, ce qui est est tout a fait suffisant pour que les effets de l'arithmétique finie ne se fassent pas sentir. Enfin, on peut penser qu'une représentation en demi-précision (2 octets) qui fournit 3 chiffres significatifs conviendrait également. La perte de précision de l'information réelle par rapport à la simple précision devrait être quasiment négligeable, car même en absence totale de bruit expérimental, la digitalisation du signal enregistré qui dépend du convertisseur analogique numérique (CAN) n'est souvent pas très élevée pour des fréquences d'échantillonnage aussi importantes que rencontrées pour des barrettes échographiques. A titre d'exemple, le CAN Lecœur utilisé au laboratoire digitalise le signal enregistré sur 12 bits, soit 4096 niveaux. Les signaux enregistrés possèdent donc 3 chiffres significatifs, et seulement pour les valeurs les plus élevées car la représentation est à virgule non flottante, ce qui laisse penser qu'avec des données expérimentales, 3 chiffres significatifs est le maximum de chiffres interprétables dans l'image obtenue. Remarquons que cette considération dépend du matériel et des condition expérimentales (bruit, moyennage), et non pas de la

méthode employée. Au delà du gain en mémoire, la représentation dans une précision inférieure peut permettre aussi de réduire le temps de calcul, et notamment sur les futures générations de GPU Nvidia qui embarqueront des unités de calcul logique en demi-précision.

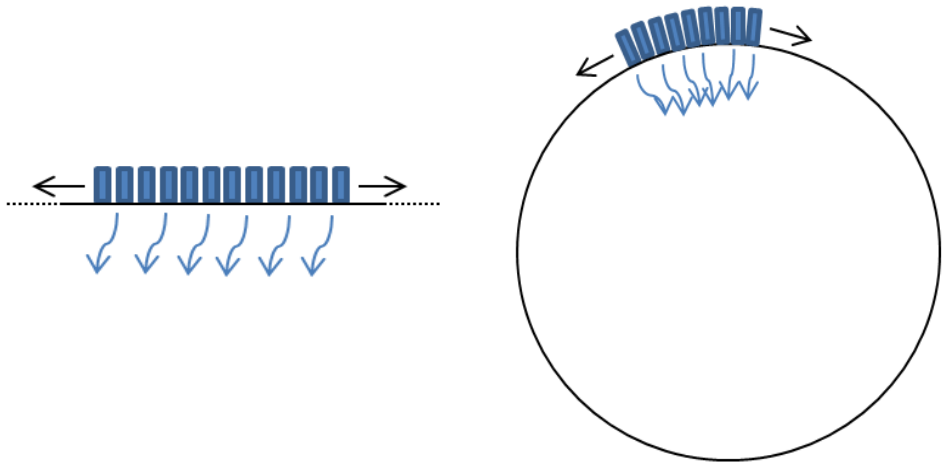
Avantages et limites de la méthode

Comparaison avec TDTE

FTIM et TDTE proposent une image très similaire, l'une étant la racine de l'enveloppe spatiale de l'autre, ce qui offre un léger avantage de lisibilité à FTIM. Mais outre le résultat en question, comme nous l'avons évoqué, la principale raison de la transposition au fréquentiel est de pouvoir réduire drastiquement le nombre d'opérations mathématiques nécessaires, pour obtenir un temps de calcul de l'image beaucoup plus faible.

Cependant, le bénéfice de cette reformulation n'intervient pas dans tous les cas de figure. Il est primordial dans la mesure du possible de pouvoir amortir le coût du calcul du diagramme de rayonnement pour pouvoir diminuer de plusieurs ordres de grandeur le temps de calcul de l'image. Pour cela, il faut idéalement avoir à utiliser un grand nombre de fois ce diagramme, en devant par exemple avoir à calculer plusieurs images d'affilée, comme c'est le cas en imagerie en temps réel. En supposant l'utilisation d'une barrette échographique multi-éléments, en fonction du contexte d'utilisation et des hypothèses sur la géométrie du milieu, il est possible de distinguer les cas suivants :

- Le milieu possède a priori une certaine symétrie, comme indiqué sur la figure 2.6. Ce cas de figure se produit dans la plupart des situations, comme en imagerie médicale, pour l'imagerie des tuyaux cylindriques, et un certain nombre d'autres objets à inspecter en CND. Grâce à cette symétrie, le diagramme de rayonnement à calculer ne dépend pas de la position du transducteur, ce qui permet donc d'utiliser le même diagramme de rayonnement pour chacun des éléments de la barrette. Un tel gain est tout d'abord appréciable pour réduire le temps de calcul de ces diagrammes, mais surtout pour la place mémoire requise par ces diagrammes, qui ne dépend plus du nombre d'éléments de la barrette. De plus, ce diagramme de rayonnement est invariant par translation de la barrette, ce qui autorise la réutilisation du même diagramme de rayonnement pour différentes acquisitions. Notons enfin qu'à 2D, il est possible par symétrie du diagramme de n'en conserver que la moitié, et qu'à 3D seul un quart de diagramme est suffisant.



(a) Milieu homogène semi-infini (b) Milieu homogène à symétrie cylindrique

FIGURE 2.6 – Symétries possibles du milieu.

- Le milieu n'a aucune symétrie. Ces situations peuvent intervenir en CND, pour l'imagerie de pièces hétérogènes lorsque que la zone d'intérêt à imager est située derrière ces hétérogénéités. Le

diagramme de rayonnement dépend alors de la position du transducteur par rapport au milieu, et il faut en calculer autant qu'il y a de transducteurs. Par ailleurs, du fait de cette dépendance il n'est pas possible de bouger la barrette échographique sans devoir tout recalculer. Si ces deux observations sont assez pénalisantes, il est possible de les contourner en sélectionnant une zone d'intérêt à imager plus étroite. Si une approche numérique de type différences ou éléments finis impose de mailler tout le domaine pour obtenir le diagramme de rayonnement, il est possible a posteriori de ne conserver que la portion du diagramme comportant la zone d'intérêt, et de réduire fortement le coût mémoire associé. Par ailleurs, Rodriguez et al [32] ont également montré la possibilité dans un cas à 2D de se passer du calcul de ce diagramme, en le mesurant expérimentalement sur la zone d'intérêt à l'aide de lasers. Ces deux alternatives sont valables également pour les autres cas de figure, bien que celui-ci en fait l'utilisation la plus pertinente. Une fois les diagrammes obtenus, il n'est pas possible de bouger la barrette, mais il est possible d'imager de cette façon une multitude de pièces complexes, en posant la barrette sur une position précise de l'objet. On peut imaginer que cette situation est tout à fait envisageable dans le contexte du CND, où une grande quantité d'objets à la structure complexe mais connue est à inspecter et où le temps de calcul associé doit être très court.

Dans les cas où le temps de calcul du diagramme de rayonnement ne peut être amorti, à savoir quand un seul objet est à imager, l'écart de performance entre FTIM et TDTE est fortement réduit. Dans le cas d'un milieu homogène, la rapidité du calcul semi-analytique dans le domaine fréquentiel peut être bien supérieur à une méthode numérique dans le domaine temporel nécessaire pour TDTE. Cet écart peut être comblé par une version semi-analytique dans le domaine temporel, mais n'aurait cependant pas la fidélité de modélisation de la géométrie du transducteur, dont les méthodes semi-analytiques fréquentielles peuvent rendre compte, comme on le verra dans la partie 2.1. Enfin, aucun gain n'est à espérer si le milieu est hétérogène et qu'une seule image est requise.

Quelques autres avantages pratiques sont par ailleurs à mettre au crédit de la méthode FTIM :

— Les approches numériques concernant la simulation de la propagation d'onde dans le domaine temporel nécessitent un niveau de discrétisation supérieur à la demi-longueur d'onde pour des raisons de convergence, obligeant à l'utilisation de tableaux plus grands que nécessaire pour stocker les champs, ce qui, en plus du coût en temps, coûte également plus de mémoire et peut limiter la taille de la résolution du domaine. De plus, ces méthodes sont parfois exprimées selon un schéma explicite, ce qui rend la simulation conditionnellement stable, et astreint le choix d'un pas de temps suffisamment petit. Le plus souvent, ce pas de temps est inférieur au pas de temps de l'échantillonnage, et force à sur-échantillonner par interpolation, alors que ce n'est pas nécessaire dans le cas fréquentiel.

— La taille des diagrammes de rayonnement peut parfois paraître pénalisante du fait de leur dépendance spatiale et fréquentielle. Remarquons cependant qu'ils possèdent la même taille que les champs acoustiques $U(f)$ et $V(f)$ qui sont calculés. Un solveur temporel calcule, sauf implémentation dédiée, les champs acoustiques un à un, obligeant pour TDTE à stocker l'intégralité du champ acoustique $U(t)$, avant de pouvoir le multiplier de le champ V lors de la simulation adjointe. Or la taille de $U(t)$ dépend à la fois de l'espace et du temps. Grâce à l'astuce de réduction de bande incorporée dans FTIM, le nombre de fréquences nécessaires à la reconstruction fidèle du champ est très inférieur au nombre de pas de temps de la méthode numérique temporelle, qui est quant à lui démultiplié par la contrainte de stabilité conditionnelle. Cela a pu être vérifié avec lors d'une expérience au laboratoire, où pour 8000 échantillons relevés, 50000 pas de temps étaient nécessaires pour représenter $U(t)$, alors que 500 fréquences suffisaient pour $U(f)$. Il en découle un facteur 100 de taille mémoire entre les deux tableaux, qui s'avère critique dès que la dimension du problème augmente. Une alternative possible à la contrainte mémoire imposée dans le domaine temporel est alors de ne sauvegarder qu'un sous-multiple du nombre de frames temporelles, conduisant inévitablement à une perte de précision de la méthode.

Ainsi, en plus d'une qualité d'image légèrement améliorée, FTIM offre un temps de calcul inférieur de plusieurs ordres de grandeur par rapport à TDTE dans la plupart des situations, et autorise la résolution de problèmes de plus grande taille, que ce soit en espace ou en fréquence.

Positionnement parmi les méthodes échographiques

Nous pouvons à présent comparer la méthode FTIM aux autres méthodes de sa catégorie.

La force principale de la méthode FTIM est de pouvoir s'adapter à des milieux fortement hétérogènes dès lors que ces informations sont a priori disponibles. Comme nous l'avons vu, pour les méthodes de type ouverture synthétique ou de retournement temporel, il est impossible de rendre compte des réflexions multiples, alors que ces effets ne sont plus négligeables dans ce type de milieu. Cet avantage doit être nuancé, car il est conditionné par la connaissance du milieu initial, la capacité mémoire, ou le déplacement du capteur.

Pour certains milieux hétérogènes possédant une certaine symétrie, comme illustré figure 2.6, il est possible de bouger le capteur sans contraintes.

Dans le cas des milieux a priori homogènes, on peut relever également plusieurs points en faveur de la méthode FTIM :

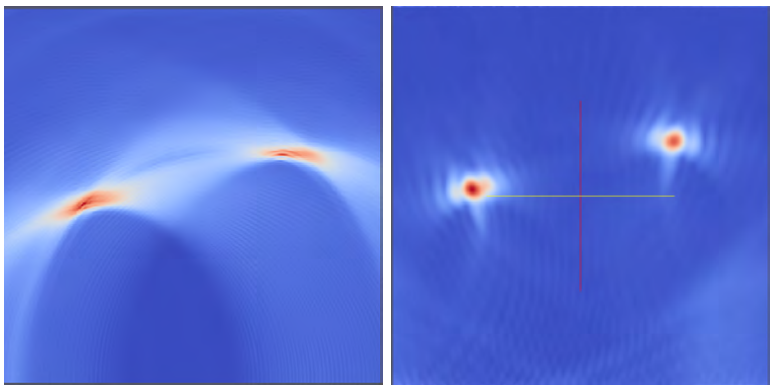
— Le modèle de propagation ne repose pas sur un algorithme de type lancer de rayons, comme c'est le cas pour la plupart des autres méthodes. Cela se traduit notamment par une meilleure prise en compte de la géométrie du transducteur, qui peut être quelconque pour la méthode FTIM. Cela offre un premier avantage pour l'imagerie des hétérogénéités situés en champ proche. Il en résulte aussi une meilleure modélisation de la directivité de la source, qui est toutefois dans les méthodes de type ouverture synthétique compensée par un terme d'apodisation. Les méthodes utilisant la matrice de retournement temporel en revanche ne peuvent pas compenser cet effet qui n'est pas inclus ni compatible avec le formalisme matriciel.

— L'imagerie d'une interface est possible, et a été démontrée expérimentalement, ce qui n'est pas le cas avec une méthode utilisant la matrice de retournement temporel.

— Une seule illumination du milieu est suffisante pour pouvoir obtenir une image, contrairement à toutes les autres méthodes, se basant sur la FMC. Cela permet de bénéficier des deux principaux avantages qui y sont associés, comme un meilleur ratio signal sur bruit et la possibilité d'une cadence d'acquisition plus élevée.

Comparaisons des résultats d'imagerie de TFM et FTIM

Afin d'apprécier davantage la différence entre FTIM et les autres méthodes échographiques, nous présentons quelques expériences numériques supplémentaires. La TFM a été choisie ici pour sa simplicité de mise en œuvre, et ses meilleurs résultats par rapport à la méthode SAFT comme vu dans la partie 1.1. Pour les deux méthodes, les calculs sont effectués en supposant une vitesse de compression égale à 1500 m/s constante dans le milieu.



(a) TFM (b) FTIM

FIGURE 2.7 – Comparaison de TFM et FTIM, avec une acquisition de type FMC. Le minimum et maximum du code couleur correspondent au minimum et maximum des images.

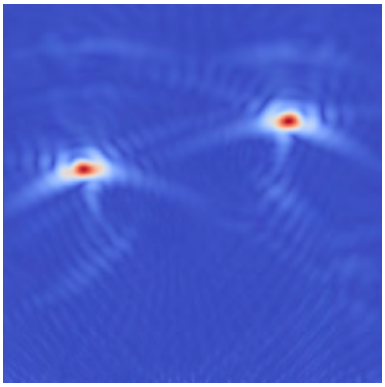
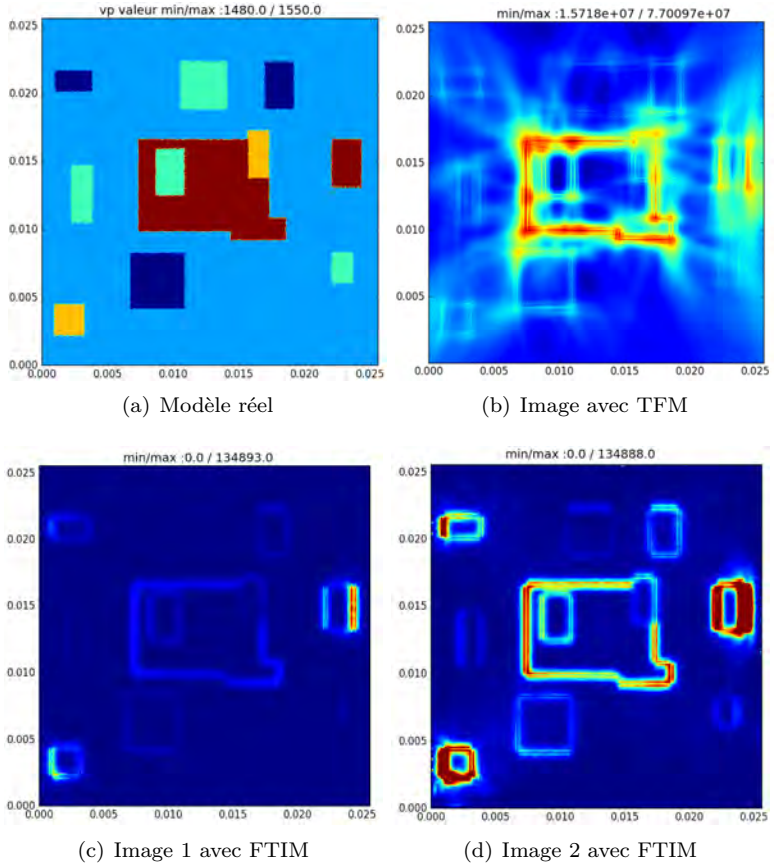


FIGURE 2.8 – Image FTIM obtenue avec une seule illumination.

Résolution spatiale La figure 2.7 présente un comparatif entre FTIM et TFM, où le milieu de référence est celui utilisé dans l'expérience illustrée sur 1.5. On peut voir le bénéfice apporté par la refocalisation avec un modèle de propagation précis : l'effet de "moustache" inhérent à l'approximation géométrique utilisée avec la TFM disparaît sur l'image obtenue avec FTIM. De fait, les hétérogénéités à reconstituer se situent en champ proche, ce qui est pénalisant pour la TFM à cause de son approximation. Par ailleurs, la résolution latérale avec la TFM est assez mauvaise : l'hétérogénéité semble "étalée". Sur l'image obtenue avec FTIM, la reconstruction qui a tiré parti des illuminations multiples en champ proche autorise une excellente résolution latérale, qui apparaît équivalente à la résolution axiale, bien que quelques oscillations persistent. Enfin, remarquons que pour pouvoir appliquer ce comparatif, une acquisition simulée de type FMC a dû être employée pour satisfaire aux exigences de la TFM, alors qu'une seule acquisition suffisait avec FTIM. Sur la figure 2.7(b), l'image est obtenue en faisant la moyenne des 32 images obtenues avec chacune des séries de Bscans. Sur la figure 2.8, on peut voir le résultat obtenu en utilisant une seule illumination du milieu, avec une onde plane. Les oscillations sont légèrement plus visibles que sur l'image 2.7(b), mais restent très acceptables en comparaison avec les résultats de la TFM et de SAFT.

Imagerie des interfaces Les méthodes conventionnelles présentent également quelques difficultés à correctement représenter les interfaces à cause de l'approximation géométrique. La figure 2.9 compare les résultats d'imagerie du milieu [2.9(a)] avec de multiples interfaces internes de contrastes différents avec les méthodes TFM et FTIM. Pour calculer ces images, une simulation de type FMC est réalisée de chaque côté du milieu avec une barrette . Le résultat obtenu avec FTIM est présenté sur deux images avec un code couleur différent : par rapport à l'image 2.9(c), le code couleur de l'image 2.9(d) est déterminé tel que le maximum de couleur (ici rouge) corresponde au maximum d'intensité de l'hétérogénéité centrale. Cet ajustement est nécessaire à cause de la refocalisation du champ adjoint sur les interfaces proches, qui cumulé à des fortes valeurs de la source adjointe génère des valeurs très élevées pour ces points. Avec cette modification, les interfaces des hétérogénéités sont plus clairement identifiables sur l'image obtenue avec FTIM qu'avec TFM, où celles-ci sont plus étalées spatialement. Un constat similaire peut être dressé pour l'imagerie d'un milieu avec de fortes hétérogénéités comme illustré sur 2.10. Sur l'image obtenue avec la TFM, de nombreux artefacts apparaissent, en particulier à cause de la diffraction par les points anguleux du domaine imagé. Ces points anguleux apparaissent également plus lumineux sur l'image FTIM que les autres interfaces à cause de la forte refocalisation, mais sont bien localisés spatialement. En revanche on peut observer que dans les deux cas les interfaces sont décalées par rapport au milieu réel, à cause de l'hypothèse d'une vitesse de compression constante dans le milieu à 1500m/s dans les deux méthodes. On peut observer en particulier la variation de position des interfaces basse et haute de l'hétérogénéité centrale, décalées à cause du passage préalable de



(a) Modèle réel (b) Image avec TFM

(c) Image 1 avec FTIM (d) Image 2 avec FTIM

FIGURE 2.9 – Comparaison de l'imagerie des interfaces avec faible contraste avec TFM et FTIM.

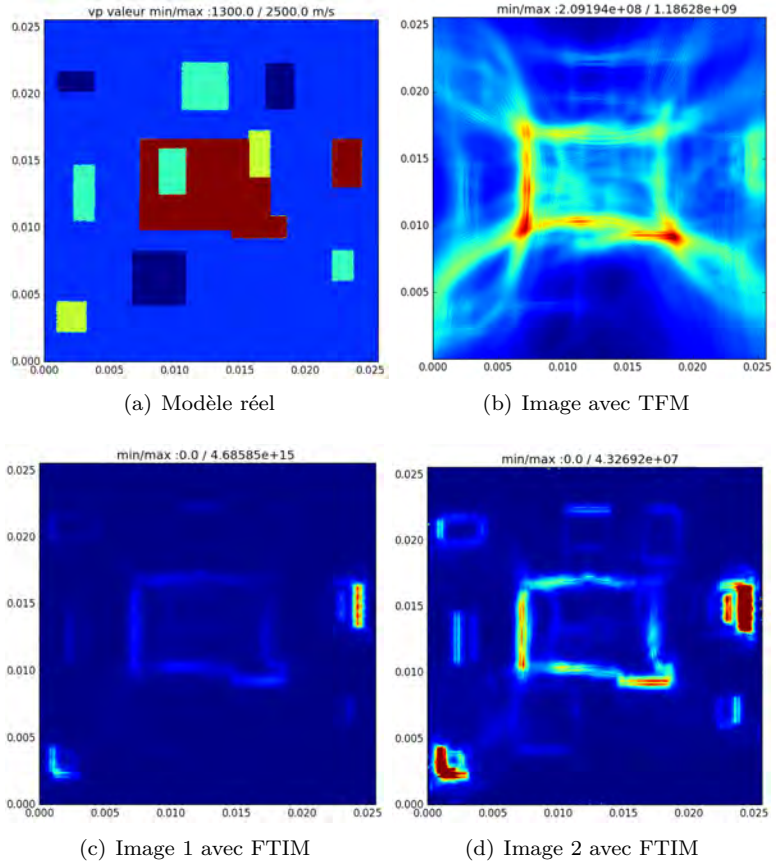
l'onde à travers d'autres hétérogénéités. La correction de ce phénomène n'est pas possible sans connaissance supplémentaire a priori sur la vitesse, ou au moyen d'algorithmes différents et plus coûteux, comme nous le verrons par la suite.

Le dernier critère à vérifier et qui constitue une condition sine qua non dans beaucoup de situations est la compatibilité de la méthode avec le temps réel. La principale difficulté est le nombre d'opérations mathématiques associées au calcul de l'image avec FTIM, qui est supérieur aux méthodes classiques. Cela constitue tout l'enjeu de la transposition sur GPU.

Calcul du diagramme de rayonnement pour un milieu homogène semi-infini

Nous allons montrer comment calculer de façon semi-analytique le diagramme de rayonnement pour un milieu homogène semi-infini, ce qui regroupe la totalité des cas actuellement traités par l'imagerie échographique. Pour ce faire, on considère les expressions suivantes, dénommées intégrales de Rayleigh [33] :

$$\text{A 2D} : \begin{cases} H(x,y,f) = \displaystyle\int_0^L -\frac{1}{2} H_0\left(\frac{2\pi f}{c} r\right) \mathrm{dx'} \\ H_0 \text{ la fonction de Hankel cylindrique d'ordre 0} \\ r = \sqrt{(x-x')^2 + y^2} \\ L \text{ la longueur d'un élément} \end{cases} \tag{2.14}$$



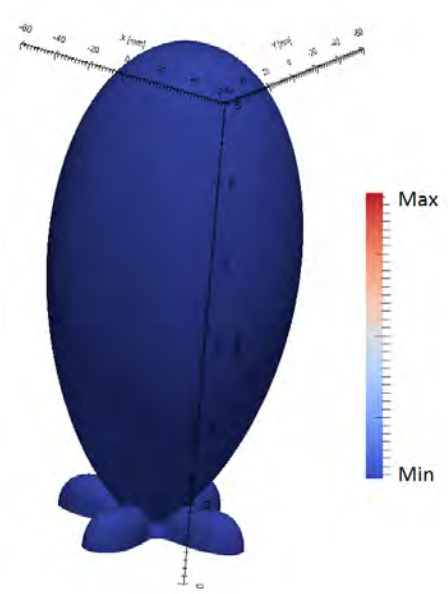
(a) Modèle réel (b) Image avec TFM

(c) Image 1 avec FTIM (d) Image 2 avec FTIM

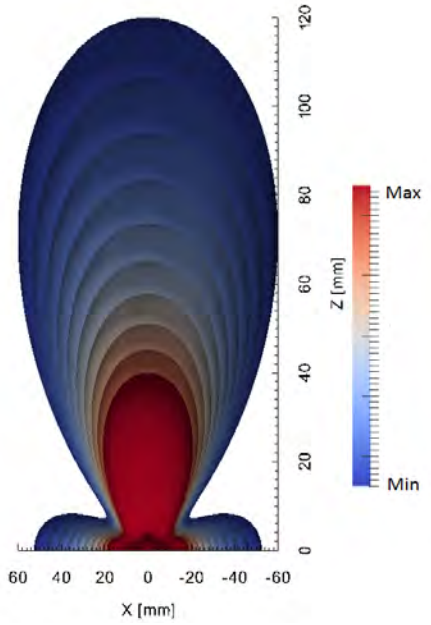
FIGURE 2.10 – Comparaison de l'imagerie des interfaces avec fort contraste avec TFM et FTIM.

$$\text{A 3D} : \begin{cases} H(x,y,z,f) = if\rho \iint_S \frac{e^{\frac{-2i\pi fr}{c}}}{r} dx'dy' \\ r = \sqrt{(x-x')^2 + (y-y')^2 + z^2} \\ S \text{ la surface d'un élément} \end{cases} \tag{2.15}$$

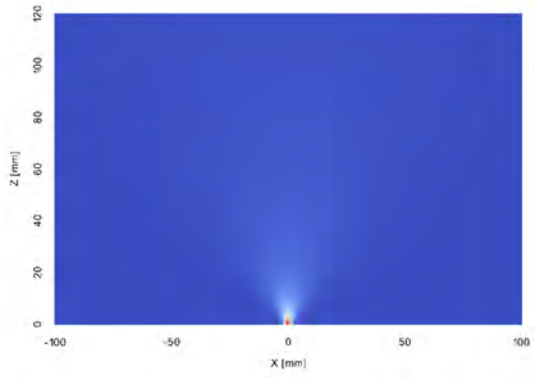
Ces intégrales expriment la façon dont une source, représentée par un segment à 2D, et par une surface à 3D, rayonne dans le milieu homogène semi-infini de masse volumique ρ et une vitesse de compression c. Cette expression présente ainsi l'avantage de pouvoir tenir compte des dimensions réelles du transducteur, afin de modéliser le plus fidèlement possible son rayonnement. Dans le cas à 3D, il est possible de représenter tout type de surface, que le transducteur soit carré, rectangulaire ou cylindrique.



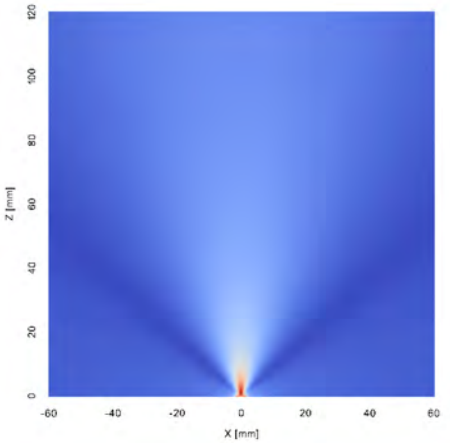
(a) Diagramme de rayonnement 3D, transducteur carré de 4mm de côté, vue en perspective



(b) Diagramme de rayonnement 3D, transducteur carré de 4mm de côté, vue en coupe



(c) Diagramme de rayonnement 3D circulaire de 4mm de diagonale, vue dans le plan de symétrie



(d) Diagramme de rayonnement 2D, transducteur de 4mm de large

FIGURE 2.11 – Module du diagramme de rayonnement de différents transducteurs à 500kHz.

Comme ces intégrales n'ont pas de solutions analytiques exactes, une intégration numérique est inévitable. Par exemple, elle prend la forme suivante dans le cas à 3D pour un élément piézo-

électrique rectangulaire de dimension $L_x \times L_y$:

$$\begin{cases} H(i,j,k,f_l) = if_l\rho \sum_m \sum_n \frac{e^{\frac{-2i\pi f_l r}{c}}}{r}(\Delta x)^2 \\ r = \sqrt{(x_i - x_m)^2 + (y_j - y_n)^2 + z_k^2} \\ x_i = i \times \Delta x \ \forall i \in [1; N_x] \\ y_j = j \times \Delta x \ \forall j \in [1; N_y] \\ z_k = k \times \Delta z \ \forall k \in [1; N_z] \\ x_m = (m-1) \times \frac{\Delta x}{8} \ \forall m \in [1; E(\frac{8 \times L_x}{\Delta x})] \\ x_n = (n-1) \times \frac{\Delta x}{8} \ \forall n \in [1; E(\frac{8 \times L_y}{\Delta x})] \\ f_l = F_{min} + (l-1) \times \delta f \ \forall l \in [1; N_{freq}] \end{cases} \tag{2.16}$$

Le calcul du rayonnement étant indépendant d'un pixel à l'autre, cette intégration se prête bien à une parallélisation sur GPU. Les détails ne sont pas présentés dans la mesure où le critère de rapidité n'est ici pas indispensable. Le facteur "8" qui apparaît a été fixé dans le but d'augmenter le niveau de discrétisation de la source par rapport à celle de l'image, afin d'améliorer la précision de l'intégration. Les résultats numériques obtenus sont présentés figure 2.11.

2.2 Implémentation sur GPU

Dans cette section, nous exposons les détails de l'implémentation de l'algorithme FTIM sur GPU, puis comparons les gains obtenus par rapport à la version CPU. Au delà de la simple valorisation du temps passé à l'écriture du code, l'intérêt de cette partie est de fournir un exemple concret pour le lecteur néophyte en matière de programmation GPU et désireux d'en découvrir les aspects primordiaux. L'accent est mis sur la façon dont l'algorithme est pensé pour s'adapter au GPU, puis repensé afin d'éviter les principaux goulots d'étranglement. Pour la bonne compréhension de cette partie, il est suggéré au lecteur non habitué à la programmation sur GPU de se référer à l'annexe A, où le vocabulaire et les bases du paradigme de programmation sur GPU sont exposées. Pour le calcul des transformées de Fourier, les fonctions prédéfinies dans la librairie cuFFT de Nvidia ont été utilisées. Comme nous l'avons vu, le temps de calcul des transformées de Fourier est négligeable dans ce problème. L'objectif de cette partie est de calculer le plus rapidement possible la quantité I_{FTIM} qui s'exprime, dans le cas à 2D par exemple, ainsi :

$$\hat{I}_{FTIM}(i,j) = \left| \sum_{l=1}^{N_{freq}} \left(\sum_{m=1}^{N_{elem}} H(i - p(m-1), j, f_l) S_m(f_l) \right) \left(\sum_{m=1}^{N_{elem}} H(i - p(m-1), j, f_l) Bscan_m(f_l) \right) \right| \tag{2.17}$$

avec p l'écart en pixels entre deux éléments piézoélectriques, S_m la transformée de Fourier du signal émis par le m$^{\text{ème}}$ élément, et $Bscan_m$ la transformée de Fourier du signal reçu par le m$^{\text{ème}}$ élément.

Cas bidimensionnel

Le cas à deux dimensions est celui qui a été le plus optimisé, car il correspond au type d'imagerie actuellement proposé par les méthodes actuelles, et ne nécessite qu'une barrette linéaire de transducteurs, dont dispose le laboratoire, par opposition à un capteur matriciel indispensable à trois dimensions. Afin de mettre en avant les atouts de la méthode, seul le cas d'un milieu avec symétrie sera considéré, ce qui nous permettra d'augmenter fortement l'intensité arithmétique. La barrette de transducteurs comporte 32 éléments comme dans nos expériences, ce qui correspond de plus à la taille d'un warp.

Mesure de la performance et matériel

Pour mesurer la performance associée à l'implémentation, il est intéressant de dénombrer le nombre d'opérations flottantes (FLOP) à effectuer pour calculer l'image. En divisant ce nombre par la durée réelle mise par la portion du code effectuant l'opération, il est alors possible de déterminer la performance liée à l'implémentation et de la comparer avec la performance maximale théorique du GPU, exprimée en FLOPS. On peut ici estimer le nombre d'opérations flottantes N_{op} nécessaire à FTIM pour obtenir l'image en se basant sur l'équation 2.17 :

$$\begin{aligned} N_{op} &\simeq \underbrace{2}_{\boxed{1}} \times (\underbrace{6}_{\boxed{2}} + \underbrace{2}_{\boxed{3}}) \times N_{elem} \times N_{freq} \times N_x \times N_y \\ &\simeq 16 \times N_{elem} \times N_{freq} \times N_x \times N_y \end{aligned} \tag{2.18}$$

1. $\boxed{1}$: deux champs acoustiques sont calculés
2. $\boxed{2}$: coût de la multiplication complexe entre un signal et le diagramme de rayonnement
3. $\boxed{3}$: coût de l'addition complexe pour la somme sur les éléments

Ce nombre représente en fait le nombre d'opérations nécessaires au calcul des deux champs acoustiques. Le reste des opérations, ne dépendant plus du nombre de transducteurs, est négligeable. Dans notre test numérique, nous utiliserons les valeurs suivantes :

$$\begin{cases} N_x = 480 \\ N_y = 1120 \\ N_{freq} = 256 \\ N_{elem} = 32 \end{cases} \tag{2.19}$$

La taille en mémoire T_{diag} du diagramme de rayonnement associé est ainsi : $T_{diag} = 1.1$Go. On peut également estimer le nombre d'opérations flottantes : $N_{op} = 70.4 \times 10^9$ opérations flottantes. La performance obtenue sera dépendante du matériel utilisé. Nous utilisons dans ce test une Nvidia Titan Black, de génération Kepler, possédant 2880 unités de calcul, 6 Go de mémoire globale, une bande passante de 336 Go/s, et une performance maximale théorique de 5.1 TFLOPS. A titre comparatif, la performance sera comparée au code Matlab implémentant FTIM, qui a été exécuté sur un Intel i5 cadencé à 3.30 GHz. Ces performances théoriques sont reportées dans le tableau 2.2.

Composant informatique	**CPU Intel i5**	**GPU Nvidia Titan Black**
Meilleur temps théorique de calcul	5.29s	0.013s
Performance maximale théorique	13.3 GFLOPS	5.1 TFLOPS
Ratio des performances maximales des deux composants		×383

TABLE 2.2 – Tableau récapitulant les performances possibles sur les CPU et GPU utilisés dans le test.

Comme on peut le voir, l'écart de performance possible entre les deux composants est sans équivoque à l'avantage du GPU, avec une accélération de ×383, alors que les deux composants sont dans la même gamme de prix. Pour ce cas test, le temps réel n'est envisageable qu'avec le GPU.

Première implémentation

La toute première considération est l'acheminement des données vers le GPU. On peut remarquer que le temps de chargement du diagramme de rayonnement du disque dur à la mémoire RAM est bien plus important que le temps de calcul visé. Même avec un SSD affichant une vitesse de transfert de BP = 500 Mo/s, on a $t_{char} = T_{diag}/\text{BP} = 2.2\text{s}$ alors que l'on souhaite $t_{calcul} \simeq 0.04\text{s}$. Le temps de chargement de la RAM vers le GPU est négligeable en comparaison, avec une vitesse de transfert de 12 Go/s entre les deux composants. Lorsque l'objectif de la simulation est de ne calculer qu'une seule image, le temps d'exécution du programme sera alors très proche du temps t_{char}. Dans notre cas cependant, le but est d'obtenir une imagerie temps réel, et comme le diagramme de rayonnement n'a besoin de n'être chargé qu'une seule fois sur la mémoire globale du GPU, il sera réutilisé ensuite, et le temps de chargement t_{char} peut être assimilé à une simple étape d'initialisation. Le temps de chargement des Bscan est quant à lui négligeable, on a en effet $t_{bscan} = N_{elem} \times \text{sizeof(bscan)}/\text{BP} = N_{elem} \times \text{sizeof(float)} \times N_{ech}/\text{BP} =\simeq 2 \times 10^{-3}\text{s}$.

Pour aborder le problème de l'implémentation, il faut se poser la question de la possibilité de parallélisation offerte par le code. Le calcul de la quantité définie à l'équation 2.17 peut-il se faire selon un modèle Single Instruction Multiple Data ? En regardant l'expression, on peut s'apercevoir de la linéarité par rapport aux données d'entrée. Le calcul du pixel $I_{ftim}(i,j)$ ne dépend pas du calcul sur les autres pixels, ce qui est propice à la parallélisation. On peut alors adapter la grille du kernel de calcul à celle de l'image, en déclarant $N_x \times N_y$ blocs et chaque bloc responsable du calcul d'un pixel. Comme les threads d'un même warp ont la possibilité de communiquer, il est possible d'effectuer la somme sur les sources en utilisant un algorithme de réduction, qui est un exemple classique sur GPU. On peut alors définir la taille d'un bloc comme étant égale au nombre de transducteurs, qui dans notre cas est 32, ce qui correspond parfaitement à la taille d'un warp. Ainsi, chaque thread calcule les 2 produits $H(i-p(m-1),j,f)S_m(f_l)$ et $H(i-p(m-1),j,f)Bscan_m(f_l)$ puis communique ses valeurs à un autre thread selon le principe de réduction. Enfin, la somme sur les fréquences peut être gérée par une boucle à l'intérieur du kernel, en incrémentant de façon séquentielle la quantité $I_{ftim}(i,j)$. Le pseudo code associé est le suivant :

Données : Signal, Bscan, Diagramme de rayonnement
Résultat : Image
Image=0;
pour *chaque fréquence* **faire**
  lire Signal, Bscan, Diagramme;
  U = Signal × Diagramme;
  V = Bscan × Diagramme;
  Réduction de U et de V sur tous les éléments;
  Image += U × V;
fin
Image=|Image|;
si *je suis le thread 0* **alors**
  Écrire Image sur mémoire globale du GPU
fin

Algorithme 1 : Première implémentation du code sur GPU

On peut remarquer que la somme sur les fréquences est calculée de façon séquentielle, alors que ce calcul ne dépend pas des fréquences voisines. Il était en effet tout à fait possible de définir une grille plus large, en déclarant $N_x \times N_y \times N_{freq}$ blocs, et en déclarant chaque bloc responsable du calcul de la contribution d'une seule fréquence au pixel (i,j). Cependant, la grille, constituée de $N_x \times N_y$ blocs est déjà suffisamment grande, le nombre de pixels $N_x \times N_y$ dépassant le nombre d'unités de calcul du GPU, ce qui assure une occupation totale. De plus, chaque bloc devrait ajouter sa contribution fréquentielle dans un tableau $I_{ftim}(i,j)$ défini sur la mémoire globale du GPU. Du fait de l'exécution parallèle et concurrente des différents blocs, il est obligatoire de s'assurer qu'entre le moment où un bloc lit la valeur du tableau I_{ftim} puis écrit la valeur

actualisée de sa contribution au même emplacement mémoire de I_{ftim}, aucun autre bloc n'accède à la case $I_{ftim}(i, j)$. Si cela se passait, certaines contributions ne seraient pas prises en compte. Pour pallier à ce problème, CUDA définit des fonctions spéciales, dites opérations atomiques. Même si ces fonctions ont été optimisées, une opération atomique demande naturellement plus de cycles d'horloge qu'une opération standard, et est à éviter dans la mesure du possible, ce qui est notre cas.

Afin d'implémenter l'algorithme de réduction responsable du calcul de la somme des contributions de chacun des transducteurs, il est nécessaire de faire communiquer les threads entre eux. Pour cela, la mémoire partagée, commune à tous les éléments d'un seul warp est toute indiquée. Son utilisation doit obéir à certaines règles, comme l'évitement des conflits de banques, pour que l'accès aux données soit le plus rapide possible. Bien utilisée, le temps d'accès à la mémoire partagée est le même que pour un registre privé du thread, soit 4 cycles d'horloge, ce qui est l'accès le plus rapide possible. L'algorithme de réduction est le suivant :

Données : U(i),V(i)
Résultat : U,V
Chaque thread i détient la valeur sur sa voie de U(i) et V(i);
j=16;
Charger la mémoire partagée : U_{sh}(i)=U(i), V_{sh}(i)=V(i);
tant que *j>0* **faire**
  si *i<j* **alors**
    U_{sh}(i) += U_{sh}(i+16);
    V_{sh}(i) += V_{sh}(i+16);
  fin
  j=j/2;
fin

Algorithme 2 : Réduction des champs acoustiques U et V

Pour ne pas trop alourdir l'exemple, cet algorithme ne sera pas expliqué en détail, et le lecteur désireux d'en savoir plus peut se référer à [34], qui explique de façon progressive et pédagogique la construction de l'algorithme. Remarquons simplement que ce modèle permet de maximiser le nombre d'additions effectuées en parallèle, tout en respectant la dépendance nécessaire de la connaissance des quantités à additionner.

On notera également que notre situation se prête bien à une réduction rapide, car le nombre de transducteurs est égal à la taille d'un warp. Remarquons aussi qu'un multiple de 32 est grandement souhaitable pour le nombre de transducteurs pour cette implémentation, afin que les warps exécutés aient tous leurs threads actifs lors de la multiplication diagramme-signal.

L'algorithme à présent terminé peut être confronté à un test numérique. Sur notre configuration test, le programme s'est exécuté en 1.79s, pour une performance correspondante de 40 GFLOPS. Si cette performance outrepasse largement la performance maximale théorique sur CPU (13.3 GFLOPS) et offre un gain déjà très intéressant, elle est encore très loin de la performance possible (5.1 TFLOPS). Nous allons voir, lors d'une deuxième analyse, les raisons de cette 'contre-performance' et comment y remédier.

Considérations sur le problème

Tout d'abord, on peut remarquer que dans ce premier algorithme, chaque warp accède à $32\times2\times N_{freq}$ octets de la mémoire globale pour obtenir les valeurs du diagramme de rayonnement, dans le but de calculer un pixel de l'image. Ainsi, pour la construction de l'image entière, au moins $64\times N_{freq}\times N_{blocs} = 64\times N_{freq}\times N_x\times N_y$ octets sont accédés, soit 32 fois la taille du diagramme de rayonnement. Même en considérant qu'il est possible d'utiliser la totalité de la bande passante du GPU, le temps d'accès au diagramme de rayonnement est $T = 32\times T_{diag}$/bande passante interne = 0.10s, qui est largement supérieur au meilleur temps théorique possible (0.013s), et prouve que

cet algorithme est limité par la bande passante. De fait, l'intensité arithmétique associée à la construction des deux champs est extrêmement faible, avec 4 opérations (deux multiplications complexes) pour 6 octets accédés, soit 0.66FLOP/octet. De plus, l'accès à la mémoire n'est pas optimal dans le sens où chaque thread du warp accède à une portion différente et non adjacente du diagramme de rayonnement, ce qui diminue considérablement la bande passante effective. Enfin, on peut remarquer que lors de la réduction, tous les threads ne sont pas utilisés : au fur et a mesure de la boucle, seul la moitié, puis le quart, etc.. effectuent une addition pour sommer les contributions de chaque voie. Cette inoccupation est également pénalisante du point de vue des performances, assurant que même en présence de toutes les données nécessaires dans les registres du warp, la plupart des threads seront à l'arrêt lors de nombreux cycles d'horloge.

On peut également penser à une réécriture du problème avec les fonctions de la librairie CUDA, en remarquant que la quantité à calculer n'est autre que la composition de deux produits matrice vecteur suivis d'une contraction, pour former les champs U et V, et d'un produit scalaire entre ces deux champs. L'intérêt de cette approche est alors d'utiliser ces fonctions optimisées par les développeurs Nvidia pour obtenir une meilleure performance. Cependant, il est inévitable de fournir en entrée de ces fonctions un diagramme de rayonnement reconstruit pour chaque transducteur par rapport à sa position sur l'image et fournis simultanément. Cela implique d'abord un coût mémoire inabordable vis à vis de la mémoire globale du GPU, nous avons en effet $32 \times T_{diag} = 35.2$ Go > 6 Go, mais cette approche souffre surtout du même défaut que notre première implémentation, à savoir l'obligation d'accéder à au moins 35.2 Go de données, ce qui laisse le problème toujours limité par la bande passante.

Ainsi, le plus gros défaut de ces deux approches est le nombre d'accès au diagramme de rayonnement, qu'il faut réduire pour espérer une nouvelle accélération. Pour cela, la seule alternative est d'augmenter le taux d'utilisation d'une donnée du diagramme en profitant de la possibilité de communication des données à l'intérieur d'un bloc, et de repenser dans la mesure du possible la fonction associée à ce dernier. La question à résoudre est alors : lorsqu'une donnée du diagramme est accédée pour la construction du champ en un pixel donné, quels sont les autres pixels qui utilisent cette même donnée ? La réponse à cette question et ses implications sont à la base de la seconde implémentation.

Seconde Implémentation

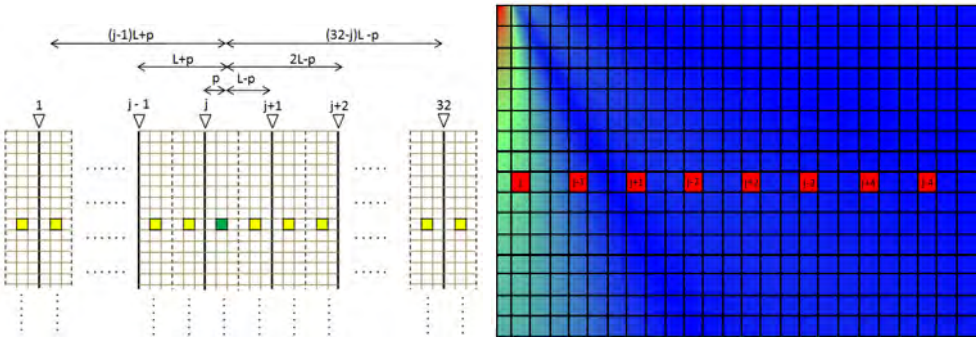
Dans le cas le plus général, il n'y a pas de loi permettant d'établir un lien entre donnée du diagramme et pixel concerné. On suppose la barrette de transducteurs posée à l'horizontale. La donnée utilisée est choisie en fonction de la distance entre la position du transducteur T_i situé en $(x_i, 0)$ et celle du pixel P_j en (x_j, z_j). La même donnée sera utilisée par le transducteur T_{i+1} situé en $(x_i + k, 0)$ pour le calcul du pixel en $(x_j + k, z_j)$. On comprend alors qu'il est indispensable de supposer la constance du pas k entre deux transducteurs adjacents pour espérer faire apparaître une certaine symétrie. Cette hypothèse est largement justifiée par le fait qu'elle est vérifiée sur la quasi totalité des barrettes échographiques utilisées en pratique. Grâce à cette hypothèse, il est alors possible d'affirmer que la donnée utilisée par le transducteur T_i situé en $(x_i, 0)$ pour le pixel P_j en (x_j, z_j) est la même que pour le transducteur T_{i+l} en $(x_i + kl, 0)$ et le pixel en $(x_j + kl, z_j)$ dès lors que $i + l < 32$. Mathématiquement, on peut écrire la construction du champ acoustique ainsi :

$$U(x, y, f) = \underbrace{\sum_{j=0}^{k-1} H(jL + p, y, f) S_{k-j}(f)}_{\text{contribution de gauche}} + \underbrace{\sum_{j=1}^{N_{elem}-k} H(jL - p, y, f) S_{k+j}(f)}_{\text{contribution de droite}} \tag{2.20}$$

avec

$$\begin{cases} x = kL + p, k \in [\![0,\, N_{elem} - 1\,]\!], p \in [\![-L/2,\, L/2\,]\!] \\ L \text{ la distance, en mm ou en pixels, entre deux éléments} \end{cases}$$

On peut déduire de l'équation 2.20 qu'à z et à p fixé, cette formule est valable pour 32 pixels à



(a) (b)

FIGURE 2.12 – Représentation des symétries du problème
(a) représente sur l'image discrétisée la distance horizontale entre le point vert choisi et chaque transducteur de la barrette. Tous les points jaunes peuvent être calculés en utilisant la même partie du diagramme. (b) représente sur le diagramme discrétisé les morceaux correspondants du diagramme qui doivent être accédés pour reconstruire le champ acoustique pour le point vert et les points jaunes.

reconstruire, et qu'elle implique 64 points du diagramme de rayonnement. De plus, pour $p' = -p$, on a :

$$\begin{aligned} U(k'L+p',y,f) &= \sum_{j=0}^{k'-1} H(jL-p,y,f)S_{k'-j}(f) + \sum_{j=1}^{N_{elem}-k'} H(jL+p,y,f)S_{k'+j}(f) \\ &= \sum_{j=0}^{N_{elem}-k} H(jL-p,y,f)S_{k'-j}(f) + \sum_{j=1}^{k-1} H(jL+p,y,f)S_{k'+j}(f) \\ &= \sum_{j=0}^{k-1} H(jL+p,y,f)S_{k'+j}(f) + \sum_{j=1}^{N_{elem}-k} H(jL-p,y,f)S_{k'-j}(f) \end{aligned} \tag{2.21}$$

L'équation 2.21 exprime, par rapport à l'équation précédente 2.20, le fait que le point $kl-p$, qui est le symétrique du point $kl+p$ par rapport à l'axe du transducteur T_k, utilise exactement les mêmes points du diagramme. Cette considération, valable $\forall k \in [0;31]$, élargit à 64 le nombre de pixels utilisant les mêmes données du diagramme. Enfin, on pourra remarquer que, comme illustré sur la figure 2.12, les données du diagramme utilisées ici ne le sont pas ailleurs, et qu'ainsi l'utilisation faite des données est optimale. Comme il est essentiel que ces 64 points du diagramme soient accédés une seule fois et qu'ils puissent être transmis entre threads, on peut alors définir que le warp est responsable du calcul des 64 pixels correspondants. Et pour mettre totalement à profit la symétrie, le thread k sera responsable du calcul des deux pixels d'abscisse $kl+p$ et $kl-p$ situés à égale distance du transducteur k. Le nombre de warp nécessaires est quant à lui égal à $k/2 \times N_z$. Les pixels situés sur l'axe de symétrie, ainsi que ceux situés en $p = E(k/2)+1$, si k est impair, jouissent d'une condition de symétrie particulière, on a en effet, sur l'axe de symétrie $H(jL+p,f) = H(jL-p,f)$ car $p=0$, et si k est impair : $H(jL+p,f) = H((j+1)L-p,f)$ $\forall j \in [0;31]$. Le fait que k soit impair dépend du niveau de discrétisation souhaité par l'utilisateur. Cette symétrie a été exploitée et deux kernels dédiés à ces points, où un warp ne calcule que 32 pixels, ont été écrits. Cependant dans la suite, nous ne présentons que le kernel dédié au cas $p \neq 0$ et $p < k$ lorsque k est impair, ce qui constitue le cas le plus complexe, et qui est aussi le plus important car la grille du kernel est $2E(k/2)$ fois plus grande, pour couvrir tous les pixels. Son optimisation est donc cruciale pour un gain substantiel en performance.

Le warp doit donc accéder, pour chaque fréquence (lesquelles seront traitées comme pour la première implémentation avec une boucle for), à 64 valeurs du diagramme de rayonnement et les communiquer entre threads du warp. Pour la communication entre threads, une nouvelle technique

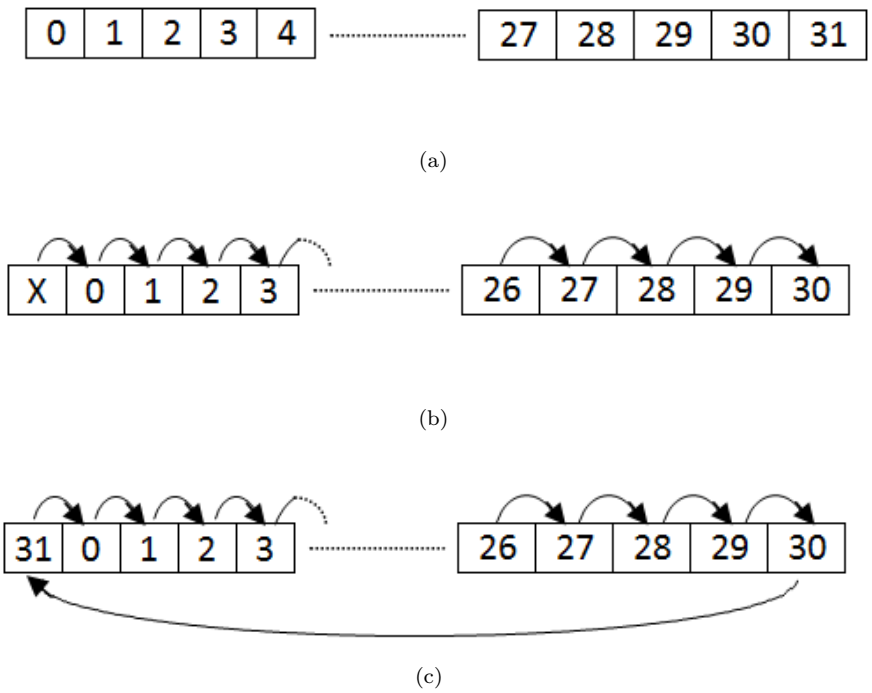


FIGURE 2.13 – Schématisation du passage des données à l'intérieur d'un warp.
(a) Chaque case représente la donnée originale contenue par les threads.
(b) représente le passage de données en faisant appel à la fonction CUDA shfl_rot.
(c) représente la version personnalisée ptx_rot de la rotation des données qui a été utilisée.

que nous nommerons ici fonctions shuffle, introduite en 2014 avec l'architecture Kepler de Nvidia, permet à un thread de lire directement dans les registres d'un autre thread du même warp sans passer par la mémoire partagée. L'intérêt de cette approche est d'éviter les phénomènes de conflit de banque qui peuvent arriver avec la mémoire partagée et la possibilité de retirer la fonction de synchronisation entre thread qui doit toujours être présente après une écriture sur la mémoire partagée. Ces fonctions shuffle permettent ainsi une lecture dans tous les cas aussi rapide que la mémoire partagée. Il en existe plusieurs, et nous utiliserons ici deux d'entre elles, le shuffle broadcast (d = shfl(D,j)) et le shuffle rotation (d = shfl_rot(D,j)). Le shuffle broadcast permet au thread numéro j d'envoyer la valeur de sa donnée D à tous les autres threads, qui pourront la recevoir dans leur registre personnel d, ou directement utiliser le retour de la fonction shuffle comme opérande pour d'autres calculs si possible. Le shuffle rotation consiste en une translation de la valeur de la donnée D du thread i vers le thread $i+j$. La fonction intrinsèque définie par CUDA n'implique pas tous les threads, de sorte que pour j>0, les j premiers threads ne reçoivent aucune donnée, comme illustré figure 2.13.

Pour utiliser et communiquer proprement ces données, il faut définir un lien entre numéro de thread et signification de la donnée. Nous choisirons, afin de permettre une implémentation de la formule 2.17, d'attribuer au thread i la responsabilité du chargement en mémoire des points du diagramme de rayonnement $D(il+p)$ et $D((i+1)l-p)$. Le signal émis et le Bscan seront chargés de la même façon que lors de la première implémentation, à savoir que le thread i chargera en mémoire le coefficient de Fourier du transducteur T_i. L'idée du nouvel algorithme est la suivante : tous les threads possèdent à ce moment là le coefficient de Fourier de la voie la plus proche de leurs pixels. Ils ont donc besoin de la donnée du diagramme de rayonnement $D(0 \times l+p)$ appartenant au thread 0, qui contient la façon dont rayonne un transducteur à p pixels d'écart. On utilise donc la fonction shuffle broadcast pour la transmettre à tous les threads. Après estimation de la contribution de la voie la plus proche, chaque thread n'a plus besoin de la donnée coefficient de

Fourier B qu'il vient d'utiliser, et peut la transmettre aux threads voisins, à droite par exemple, en utilisant la fonction shuffle rotation shfl_rot(B,1). Alors, le thread i possède la valeur de B du transducteur $i-1$. Cette formule ne s'applique bien entendu pas au thread 0. L'axe du transducteur $i-1$ est situé à une distance de $k-p$ et $k+p$ pixels des pixels qui doivent être calculés par le thread i. On déduit alors la nécessité de broadcaster les valeurs du diagramme de rayonnement détenues par le thread 0 ($D((0+1)l-p)$) et par le thread 1 ($D(1\times l+p)$). Ce processus peut être répété jusqu'à l'itération 31, pendant laquelle seul le thread 0 transmet sa valeur au thread 31. A la fin, l'ensemble des contributions des voies situées à la gauche du pixel considéré sont prises en compte, ce qui correspond au premier membre de droite de l'équation 2.20. Le second membre de droite peut être obtenu de façon similaire en appelant la fonction shuffle rotation shfl_rot(B,-1), puis shfl_rot(B,-2) etc... Le pseudo code est présenté dans l'algorithme 3.

Dans cette implémentation, on peut voir que l'opération de réduction s'écrit naturellement en sommant au fur et à mesure de la boucle. On peut également voir qu'une variable 'zero' a été introduite, et qu'elle est utilisée lorsque le test logique est faux. Cela peut paraître surprenant, mais pour assurer que l'ensemble des threads d'un warp exécutent le même nombre d'instructions, cette approche est obligatoire pour assurer la synchronisation des threads, dans la perspective d'assurer une performance optimum. Il est en général déconseillé d'effectuer un test conditionnel dépendant du numéro de thread, qui désynchronise l'exécution à l'intérieur du warp et qui conduit à une chute significative en performance. Ici, quelle que soit l'issue du test, un thread donné devra effectuer le même nombre d'opérations que les autres. Idéalement, on souhaiterait éviter de recourir au stratagème de l'addition d'un zéro, et à la place ajouter la contribution d'un autre transducteur. Remarquons que pour l'ajout des contributions des transducteurs à gauche, le thread 0, puis les threads 0 et 1, et ainsi de suite, deviennent inactifs au fil de la boucle for. En revanche si on regarde la boucle à l'envers, seul le thread 31, puis les threads 30 et 31, etc, sont actifs. Pour l'ajout des contributions de droite, c'est le contraire, le thread 31, puis les threads 30 et 31, etc, deviennent inactifs au fil de la boucle. On peut arriver à réduire cette innocupation en s'arrangeant pour que, par exemple, lorsque le thread 31 est le seul thread inactif lors de l'ajout des contributions de droite, il redevienne actif en ajoutant la dernière contribution de la gauche (celle du transducteur 0), qui est normalement effectuée à l'itération 31 de la deuxième boucle. Pour cela, nous avons implémenté en PTX (Parallel Thread eXecution), le langage sur lequel CUDA repose, une version plus bas niveau permettant aux j derniers threads d'envoyer leur donnée aux j premiers threads, et vice versa si j est négatif. Il s'agit donc d'une version modifiée de la fonction shuffle rotation, illustrée en 2.13(c). Grâce à elle, l'opération de rotation permet réellement la transmission d'une donnée nouvelle à la totalité des threads. Cela va nous permettre ici de diviser par deux la taille de la boucle sur les voies, et de rendre l'ensemble des opérations utiles, en supprimant l'utilisation du zero. Le nouveau pseudo code, qui est le dernier, est présenté sur l'algorithme 4.

La dernière condition à régler est l'adjacence des accès à la mémoire globale pour obtenir le maximum de la bande passante. Nous devons donc nous arranger pour que le diagramme de rayonnement soit arrangé tel que nous l'avons défini, à savoir que le thread i charge la donnée $D(il+p)$ puis $D((i+1)l-p)$. En mémoire, il faudra que $D(0\times l+p)$, $D(1\times l+p)$, $D(2\times l+p)$ etc soient adjacent en mémoire, suivis de $D((0+1)l-p)$, $D((1+1)l-p)$, $D((2+1)l-p)$... Cet arrangement est effectué en amont, lors de la création du diagramme de rayonnement, comme illustré figure 2.2.

Le nouvel algorithme est à présent terminé, et sa confrontation à l'expérience réelle donne un temps de 0.046s, soit 1.53TFLOPs. Par rapport à la première implémentation sur GPU, un facteur 40 est obtenu, soit le même gain que du passage du CPU vers la première implémentation GPU comme montré dans le tableau 2.3, qui recense le résultat de chacune des expériences. Même si le code Matlab proposé sur CPU n'avait pas été pensé pour obtenir un gain maximal, l'écart considérable entre les deux composants justifie amplement la transcription sur GPU. Le cas test utilisé, qui correspond à une situation réaliste avec une résolution d'affichage très confortable, peut être utilisé en temps réel, avec environ 23 frames par seconde.

Données : Signal, Bscan, Diagramme de rayonnement
Résultat : 64 points de l'image résultat.
Image1=0 ;Image2=0 ; zero=0;
pour *chaque fréquence* **faire**
    `// mon_thread contient le numéro de thread.`
    Lire S=Signal[mon_thread], B=Bscan[mon_thread],
     D1=Diagramme1[mon_thread],D2=Diagramme[mon_thread+32];
    `// Broadcast de la valeur de D1 du thread 0, qui est le point du diagramme situé à` p `pixels de l'axe du transducteur`
    d1=shfl(D1,0);
    `// U1 et U2 seront les champs directs des deux points situés à` p `pixels de la voie numéro mon_thread.`
    U1 = S × d1 ; V1 = B × d1 ; U2 = S × d1 ; V2 = B × d1;
    d2=shfl(D2,0);
    pour i = *1 ....nombre de transducteurs-1* **faire**
        `// Décalage de` i `threads vers le bas de la valeur du Bscan (le signal est supposé être le même sur chaque voie)`
        b = shfl_rot(B,-i);
        `// Broadcast de la valeur de D1 du thread` i`, qui est le point du diagramme situé à` $p + L \times i$ `pixels de l'axe du transducteur.`
        d1=shfl(D1,i);
        `// Ajout des contributions des voies de gauche.`
        si *mon_thread > i-1* **alors**
            U1+= S × d2 ; V1+= b × d2 ; U2+= S × d1 ; V2+= b × d1;
            sinon
                U1+= zero × d2 ; V1+= zero × d2 ; U2+= zero × d1 ; V2+= zero × d1;
            fin
        fin
        b = shfl_rot(B,i);
        `// Ajout des contributions des voies de droite.`
        si *mon_thread < 32 - (i+1)* **alors**
            U1+= S × d1 ; V1+= b × d1 ; U2+= S × d2 ; V2+= b × d2;
            sinon
                U1+= zero × d1 ; V1+= zero × d1 ; U2+= zero × d2 ; V2+= zero × d2;
            fin
        fin
        d2=shfl(D2,i);
    fin
    Image1 += U1 × V1 ; Image2 += U2 × V2;
fin
Image1=|Image1| ;Image2=|Image2|;
Écrire Image1 et Image2 sur mémoire globale du GPU;

Algorithme 3 : Première réécriture du code pour un warp. L'utilisation de la variable zero évite la divergence d'instructions au sein du warp.

Données : Signal, Bscan, Diagramme de rayonnement
Résultat : 64 points de l'image résultat.
Image1=0 ;Image2=0;
pour *chaque fréquence* **faire**
  `// mon_thread contient le numéro de thread.`
  Lire S=Signal[mon_thread], B=Bscan[mon_thread],
   D1=Diagramme1[mon_thread],D2=Diagramme[mon_thread+32];
  `// Broadcast de la valeur de D1 du thread 0, qui est le point du diagramme situé à` p `pixels de l'axe du transducteur`
  d1=shfl(D1,0);
  U1 = S × d1 ; V1 = B × d1 ; U2 = S × d1 ; V2 = B × d1;
  d2=shfl(D2,0);
  `// Boucle ajoutant les contributions des voies de gauche.`
  pour *i = 1 ....(nombre de transducteurs-1)/2* **faire**
    b1 = ptx_rot(B,-i);
    b2 = ptx_rot(B,i);
    `// Broadcast de la valeur de D1 du thread` i`, qui est le point du diagramme situé à` $p + L \times i$ `pixels de l'axe du transducteur.`
    d1=shfl(D1,i);
    d1p=shfl(D1,32-i);
    d2p=shfl(D2,32-i);
    si *mon_thread > i-1* **alors**
      U1+= S × d2 ; V1+= b1 × d2 ; U2+= S × d1 ; V2+= b1 × d1;
      sinon
        U1+= S × d2p ; V1+= b1 × d2p ; U2+= S × d1p ; V2+= b1 × d1p;
      fin
    fin
    d2=shfl(D2,i);
    si *mon_thread < 32 - (i+1)* **alors**
      U1+= S × d1 ; V1+= b2 × d1 ; U2+= S × d2 ; V2+= b2 × d2;
      sinon
        U1+= S × d1p ; V1+= b2 × d1p ; U2+= S × d2p ; V2+= b2 × d2p;
      fin
    fin
    d2=shfl(D2,i);
  fin
  Image1 += U1 × V1 ; Image2 += U2 × V2;
fin
Image1=|Image1| ;Image2=|Image2|;
Écrire Image1 et Image2 sur mémoire globale du GPU;

Algorithme 4 : Seconde et dernière réécriture du code pour un warp. Une boucle for avec seulement 16 itérations est suffisante pour calculer les 4 champs.

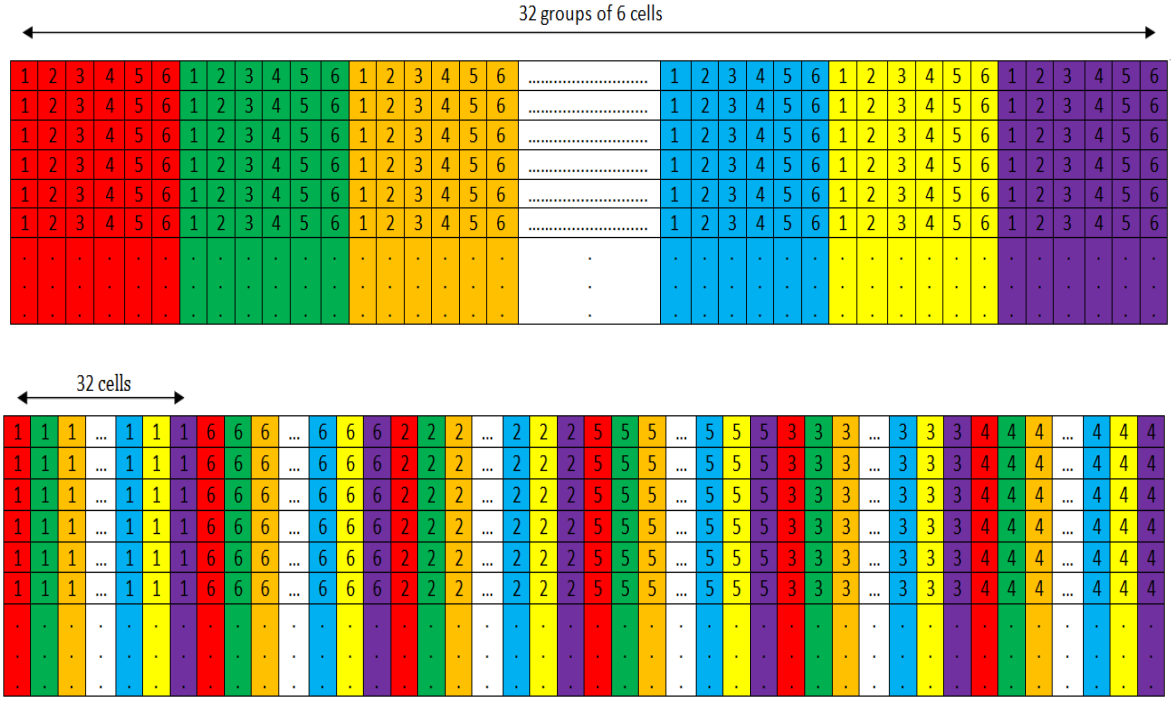


FIGURE 2.14 – Réarrangement du diagramme de rayonnement pour assurer l'adjacence des accès mémoire au sein d'un warp.

Composant informatique/Software utilisé	**CPU/Matlab**	**Première version GPU/CUDA**	**Seconde version GPU/CUDA**
Meilleur temps possible	5.29s	0.013s	0.013s
Temps de calcul mesuré	71s	1.79s	0.046s
Accélération par rapport à la version CPU		40	1580

TABLE 2.3 – Temps de calcul requis par la méthode FTIM sur CPU, et sur GPU avec deux implémentations différentes. Dans notre configuration test, environ 70.4 milliards d'opérations flottantes sont nécessaires. La version CPU version a été testée sur un Intel i5 cadencé à 3.30 GHz et les versions GPU versions sur une Nvidia Titan Black avec 2880 cœurs cadencés à 889 MHz.

Visualisation en temps réel

Le dernier défi associé au calcul en temps réel est à présent de pouvoir afficher à l'écran l'ensemble des images obtenues en temps réel. A priori sans difficulté, cette tache n'est pas aussi simple à réaliser concrètement. Par exemple, la transformation de l'image obtenue, qui est pour l'instant un tableau de float en binaire, en image JPEG prend environ une seconde, et l'affichage à l'écran du JPEG n'est pas instantané non plus. Pour résoudre le problème, nous avons mis à profit l'interopérabilité entre CUDA et OpenGL, qui est une API capable d'afficher des structures de données complexes. La seule transformation numérique nécessaire consiste en la transposition du tableau de float, en un tableau de triplets d'entiers courts non signés, qui est la convention habituellement utilisée pour définir une couleur en RGB. Le choix de la convention couleur est libre de choix pour l'utilisateur, nous avons ici choisi de représenter la valeur la plus faible du tableau en bleu, la plus forte en rouge, et en utilisant une échelle définie sur 256 niveaux pour les valeurs intermédiaires, en utilisant la formule : Niveau de couleur local $= \frac{\text{Valeur locale de } I_{FTIM}}{\max I_{FTIM} - \min I_{FTIM}}$.

Le passage à OpenGL s'effectue en déclarant un pointeur 'Pixel Buffer Object' (PBO), qui est une structure prédéfinie pour l'affichage à l'écran, et en pointant le PBO sur l'adresse du tableau

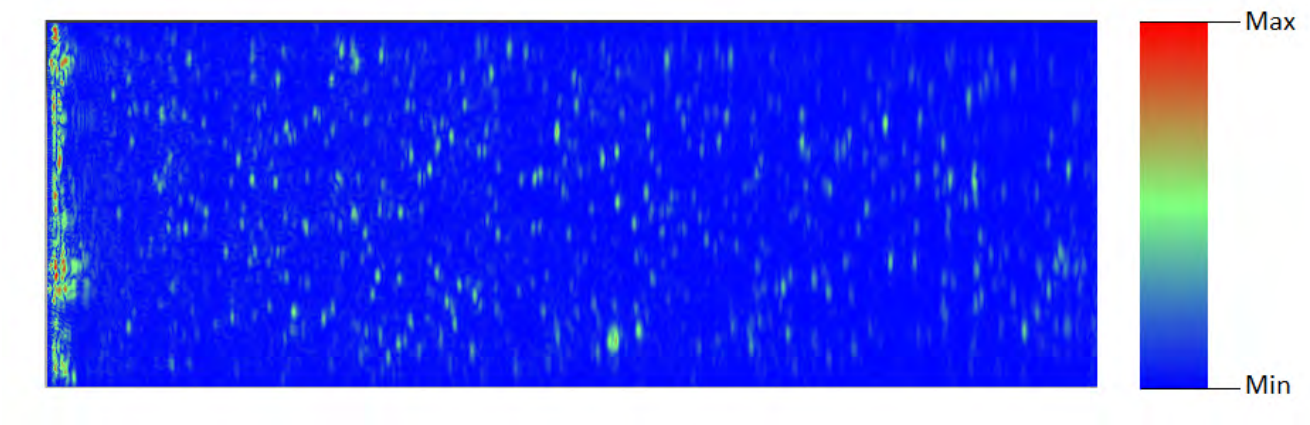


FIGURE 2.15 – Rendu à l'écran avec OpenGL.

d'entiers courts non signés, défini en mémoire globale avec CUDA. Alors, l'image une fois calculée est affichée instantanément à l'écran, comme illustré image 2.15. L'intérêt de l'interopérabilité CUDA-OpenGL réside dans le fait qu'aucun transfert CPU-GPU n'est nécessaire, OpenGL utilise directement le tableau défini par CUDA. Le cas échéant, il aurait fallu faire deux transferts GPU-CPU puis CPU-GPU pour obtenir le même résultat. Enfin, l'image calculée peut être enregistrée sur le disque dur si nécessaire, mais peut aussi être simplement effacée en étant remplacée par une nouvelle image, selon les désirs de l'utilisateur.

Cas tridimensionnel

Le cas à trois dimensions a été implémenté dans le but de démontrer sa faisabilité, mais n'a pas été optimisé comme le cas à deux dimensions. A trois dimensions, on peut remarquer que la forme du transducteur confère certaines propriétés au diagramme de rayonnement qu'il peut être intéressant d'exploiter : dans le cas de transducteurs circulaires, le diagramme de rayonnement observe lui aussi une symétrie de révolution autour de l'axe principal du transducteur. Ainsi, l'information à trois dimensions représentant le diagramme de rayonnement peut être contenue dans un demi-plan, ce qui peut permettre un gain subtantiel en terme de coût mémoire. Comme la taille du diagramme de rayonnement peut déjà être un facteur critique dans le cas à 2D, nous avons décidé de nous concentrer sur le cas des transducteurs circulaires, afin de limiter ce coût mémoire.

On peut nuancer l'affirmation précédente en remarquant que les barrettes matricielles utilisent souvent une fréquence centrale moins élevée que les barrettes linéaires, ce qui peut compenser la dimension spatiale supplémentaire, car une fréquence inférieure autorise une discrétisation spatiale moins fine. De plus, pour une durée d'enregistrement égale, un nombre inférieur de fréquences est nécessaire pour reconstruire parfaitement le signal car la bande passante est moins large et le pas fréquentiel optimal ne dépend pas de la fréquence.

La structure de l'algorithme utilisé est similaire à la première implémentation du cas à 2D, avec un warp dédié au calcul d'un pixel. Dans ce warp, chaque thread utilise la distance horizontale entre le transducteur qu'il représente et le pixel considéré pour déduire quelle donnée du diagramme de rayonnement utiliser. Cette distance, calculée dans un plan à deux dimensions, est à valeur réelle. Elle est alors convertie en entier en fonction du pas de discrétisation spatiale du diagramme. Nous avons pu utiliser cet algorithme avec des données expérimentales, fournies par l'Institut de Mécanique et d'Ingénierie de Bordeaux, qui ont été obtenues avec l'utilisation d'un capteur matriciel de 11 X 11 transducteurs avec une fréquence centrale de 500 kHz. Lors de cette expérience, une bille en métal était immergée dans de l'eau et maintenue par un fil en nylon. Le résultat obtenu, représenté figure 2.16, identifie clairement la bille, même si ses contours restent

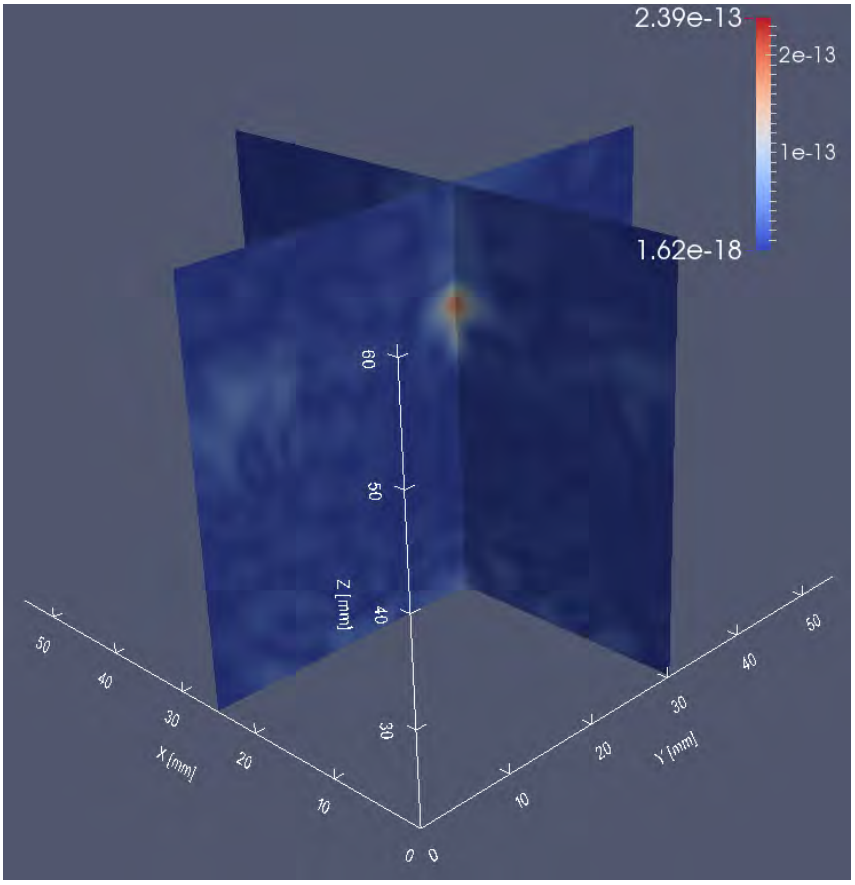


FIGURE 2.16 – Image 3D d'une bille de plomb obtenue avec un capteur matriciel.

approximatifs. Ceci s'explique notamment par l'approximation de la géométrie des transducteurs, modélisés comme étant circulaires afin de pouvoir utiliser un diagramme de rayonnement à 2D, alors qu'ils sont carrés dans la réalité. Si le diagramme de rayonnement des deux géométries est assez similaire, les lobes secondaires diffèrent et cette hypothèse est trop forte alors que la bille était placée dans le champ proche des capteurs.

Un travail futur concernera l'implémentation du code FTIM pour des configurations de transducteurs rectangulaires utilisant un véritable diagramme de rayonnement à 3 dimensions spatiales. Une implémentation performante basée sur la deuxième implémentation du code à 2D est possible, sans être immédiate.

2.3 Application à la PIV

Afin de mettre en avant les atouts de rapidité de la méthode, le laboratoire PHASE expérimente depuis quelques années l'application à la vélocimétrie par imagerie des particules, à laquelle nous référerons par son acronyme anglais, PIV (Particle Image Velocimetry). Elle consiste à déterminer le champ de vitesse d'un fluide à partir du suivi de la position de différentes particules immergées dans ce fluide. Habituellement, la position des particules est déterminée de façon optique, ce qui implique souvent un coût matériel assez élevé à cause de l'utilisation de caméras CCD dédiées. Par ailleurs, cette méthode restreint à une utilisation à des milieux non opaques à la lumière. L'unique inconvénient inhérent aux ultrasons est une cadence d'acquisition moins élevée à cause de la vitesse du son, et dépend de la profondeur du milieu. En milieu turbulent, l'imagerie ultrason reste difficile à cause d'une cadence d'acquisition très élevée et de l'apparition de phénomènes non linéaires. Ainsi, la PIV par ultrasons se positionne comme une alternative économique par rapport à son analogue optique. Les hypothèses de la méthode FTIM s'adaptent particulièrement

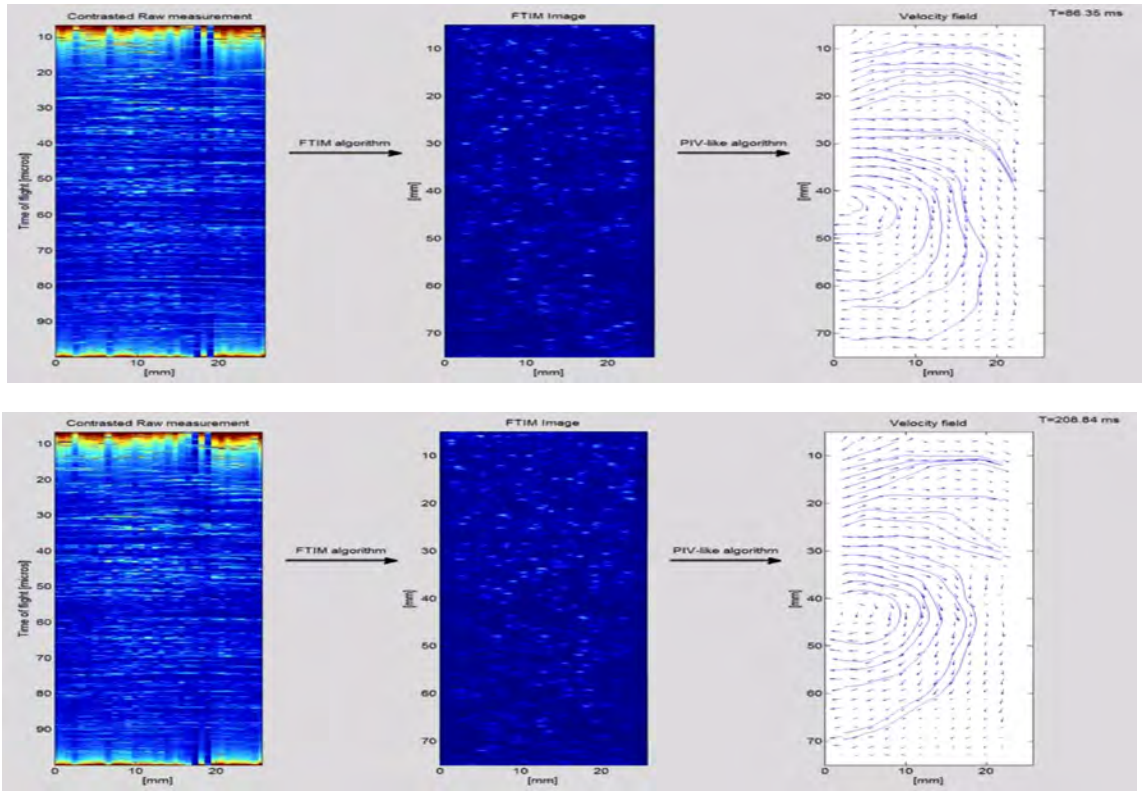


FIGURE 2.17 – Détermination du champ de vitesse à partir de la position des particules. Cette étape, codée par S. Rodriguez sur Matlab, devrait prochainement être transposée sur GPU.

bien à ce problème, car la vitesse du son dans le fluide est bien connue et les particules sont suffisamment peu denses et petites pour que l'approximation de Born soit acceptable. Pour les acquisitions, nous avons utilisé une barrette échographique Imasonic de 128 transducteurs avec une fréquence centrale de 5MHz. Le convertisseur analogique-numérique Lecœur Electronique permet le traitement simultané de 32 canaux, ce qui restreint à utiliser 32 transducteurs. Avec une fréquence d'échantillonnage maximale de 80MHz, le signal numérisé respecte les conditions imposées par le théorème de Shannon. Plusieurs expérimentations ont été réalisées, tout d'abord avec de la béatite, puis avec des particules dédiées de PIV optique. Les résultats sont présentés sur la figure 2.3. A cause de la carte d'acquisition, les données enregistrées n'ont pas pu être envoyées en temps réel vers l'ordinateur et donc de réaliser concrètement l'imagerie en temps réel. Cet inconvénient devrait être contourné à l'avenir avec l'utilisation d'une carte d'acquisition compatible avec le temps réel.

Enfin, la partie calcul du champ de vitesse à partir de deux images n'a pas encore été implémentée sur GPU, et interfacée avec le code FTIM. La façon la plus conventionnelle de procéder est de découper en plusieurs petites fenêtres ces deux images, et de corréler les différentes fenêtres. A 2D, cette corrélation, effectuée dans le domaine de Fourier, a un coût de calcul proportionnel à $N^2 log(N)$ en utilisant un algorithme de type FFT. Pour la configuration utilisée lors des expériences, cela revient à devoir faire entre 5 et 50 millions d'opérations pour calculer ce champ de vitesse, ce qui est très inférieur au nombre d'opérations nécessaires pour le calcul des images. Ainsi, la perspective du temps réel pour la PIV à 2 voire 3 dimensions est totalement envisageable, et devrait être l'objet d'un futur travail.

Chapitre 3

L'inversion de la forme d'onde complète

L'inversion de la forme d'onde complète, à laquelle nous référerons par son acronyme anglais, FWI (Full Waveform Inversion), est une technique d'imagerie quantitative issue de la géophysique. En toute généralité, la FWI peut être vue comme une méthode d'assimilation de données, où les données observées sont incorporées avec un traitement mathématique rigoureux à un modèle numérique décrivant la réalité. Mathématiquement, elle repose sur la minimisation itérative de l'écart entre données réelles et données simulées numériquement, en modifiant localement les propriétés physiques du milieu qui influencent la propagation d'onde. Cette méthode a été initiée par les travaux de Claerbout en 1971 [35], qui visaient à reconstruire à partir de données en réflexion les couches des réservoirs de pétrole situés dans le sous-sol. En 1984, Tarantola [36] a exprimé à nouveau le problème en n'impliquant que le calcul de deux champs acoustiques par source et par itération. Cependant, en dépit de cette reformulation, le coût numérique de ce calcul est resté très longtemps prohibitif même à deux dimensions. De plus, l'utilisation de données essentiellement en réflexion n'a pas apporté les résultats escomptés : au lieu de reconstruire une carte de densité et de module de compression, ces méthodes ont montré en pratique qu'elles permettaient de retrouver les interfaces internes du milieu. Ainsi, on réfère à ces deux travaux fondamentaux comme principe d'imagerie, ou technique de migration. A cause du temps de calcul, des modèles de propagation basés sur l'équation eikonale ont d'abord été utilisés et des résultats pertinents sur la structure globale de la Terre à une dimension ont été obtenus, grâce à l'utilisation d'ondes en transmission. Puis, avec l'évolution progressive de la puissance de calcul, l'utilisation de modèles plus réalistes ont permis d'incorporer les effets 2D et 3D, mais aussi de prendre en compte l'aspect fini des fréquences des ondes sismiques. La compréhension de l'intégralité du contenu des sismogrammes est devenue possible et interprétable, et c'est en ce sens que l'on emploie le terme inversion de la forme d'onde complète.

L'objectif de ce chapitre est de présenter la FWI. La formulation mathématique du problème d'optimisation ainsi que les différentes techniques amenant à sa résolution seront exprimées. Ensuite, nous verrons les éléments clés de l'inversion, qui influent d'une part sur la capacité à converger vers la solution la plus proche de la réalité en dépit de la forte non linéarité du problème et de la non-unicité possible de la solution, et d'autre part sur la vitesse à laquelle il sera possible de converger vers la solution, qui s'avère en pratique être aussi un facteur critique étant donné la taille du problème et le coût numérique associé. Ces facteurs peuvent être liés à la fois au problème mathématique d'optimisation en lui-même, mais aussi être spécifiques à l'implication de l'équation d'onde. Enfin, comme cette méthode fait intervenir la simulation numérique de la propagation d'onde et que la précision avec laquelle elle est simulée joue également un facteur déterminant, nous décrirons la méthode numérique employée.

3.1 Théorie

Mise en équation du problème

Tout comme pour la formulation du problème d'optimisation topologique décrit au début du chapitre précédent, la FWI repose sur la minimisation d'une fonction $f(m)$ qui évalue l'écart entre données réelles $p_{obs}(\mathbf{x}_i, t)$ et données simulées $p_{syn}(\mathbf{x}_i, t, m)$, ce qui s'exprime :

$$\min f(m) = \frac{1}{2} \sum_{i=1}^{N_{rec}} \int_0^T \left| p_{obs}(\mathbf{x}_i, t) - p_{syn}(\mathbf{x}_i, t, m) \right|^2 dt \tag{3.1}$$

De même, la modification locale et itérative d'un ou de plusieurs paramètres physiques m du modèle du milieu à imager est le moteur de cette minimisation, et la combinaison finale de chaque paramètre, qui s'apparente à une carte spatiale des paramètres en question est l'image recherchée. Dans le cas de l'optimisation topologique, le paramètre utilisé est la topologie Ω du milieu, et l'image obtenue définit la forme des bords et des trous tels qu'ils sont dans la réalité. Pour la FWI appliquée en géophysique, les paramètres physiques sont des grandeurs continues telles que la densité $\rho(\mathbf{x})$, le tenseur d'anisotropie $c_{ijkl}(\mathbf{x})$ ou l'atténuation $\alpha(\mathbf{x})$ du milieu. Dans les deux cas, la propagation d'onde influence les données, ce qui se traduit mathématiquement par un problème d'optimisation contraint par une équation aux dérivées partielles. Dans le cas de l'acoustique linéaire, avec la vitesse de compression c et la masse volumique ρ comme fonctions de la position $\mathbf{x}$, en introduisant le module de compressibilité isotrope $\kappa = \rho c^2$, et en définissant un potentiel scalaire Φ à partir du déplacement s :

$$s = \frac{\nabla \Phi}{\rho} \tag{3.2}$$

Cette définition inhabituelle du potentiel à partir du déplacement est héritée de l'origine géophysique de cette méthode, qui permet une écriture simplifiée des phénomènes de couplage entre milieux fluide et solide.

On définit alors l'équation d'onde suivante, dont la pression est la dérivée seconde :

$$\begin{cases} p_{syn}(\mathbf{x}_r, t) = -\partial_t^2 \, \Phi(\mathbf{x}_r, t) \\ \kappa(\mathbf{x})^{-1} \partial_t^2 \, \Phi - \nabla \cdot (\rho(\mathbf{x})^{-1} \nabla \, \Phi) = \sum_{s=1}^{N_s} \kappa(\mathbf{x}_s)^{-1} f_s(t) \end{cases} \tag{3.3}$$

Les termes sources $f_s(t)$ sont des sources de pression, ce qui correspond par ailleurs à la nature physique des données enregistrées, et montre la pertinence de cette formulation en potentiel. Le cas échéant, une double dérivation numérique des signaux enregistrés aurait été nécessaire, ce qui aurait augmenté le bruit numérique.

On pourra remarquer que l'implication de l'équation d'onde avec des cartes de paramètres $c(\mathbf{x})$, $\rho(\mathbf{x})$ ou $\kappa(\mathbf{x})$ rend le problème d'optimisation non convexe. De plus, la relative liberté quant au nombre de sources, leur position spatiale et la durée des signaux enregistrés influence largement la possibilité à obtenir la vraie solution. Si la réalisation pratique de l'expérience permet d'affirmer l'existence de la solution en supposant l'absence de bruit expérimental, l'unicité n'est pas garantie non plus, à cause de la non convexité du problème.

Résolution du problème inverse

Choix de la méthode de résolution

De façon générale, le coût numérique d'évaluation de la fonction coût joue un rôle primordial dans le choix de l'algorithme à utiliser. Normalement, pour un problème non convexe comme ici, les algorithmes d'optimisation globale, tels que des méthodes stochastiques ou heuristiques, sont en pratique les plus efficaces quant à la convergence vers l'un des optima globaux. Cependant,

ces derniers impliquent un très grand nombre d'évaluations de la fonction coût et/ou de son gradient, ce qui les rend inutilisables de façon pratique dès lors que cette évaluation se chiffre en minutes ou plus. Ici, l'évaluation de la fonction coût est très coûteuse car elle n'est pas obtenue de façon analytique ou semi-analytique, comme cela aurait pu être le cas si le milieu de départ était homogène, mais que seul les méthodes numériques permettent d'inclure proprement dans le calcul de la fonction coût l'ensemble des conditions aux limites, ainsi qu'une carte spatiale potentiellement variable de l'ensemble des paramètres impliqués dans l'équation d'onde. L'utilisation de ces méthodes numériques pour évaluer la fonction coût, qui dans notre cas est un simulateur de propagation d'onde, est systématiquement coûteuse en temps de calcul, et inévitable ici. Pour limiter le nombre d'estimations de la fonction coût, le seul moyen est de passer par des méthodes d'optimisation locale, de type descente. Le principal inconvénient est lié au caractère local de l'optimisation, qui dans le cas d'un problème non linéaire et non convexe se traduit par le risque fort de converger vers un minimum local et non global. Étant donné la forte non convexité du problème, on peut se représenter la fonction coût comme étant une planète entièrement remplie de montagnes, et l'optimisation locale consiste à descendre vers le lac ou la vallée la plus proche, en fonction de la position initiale choisie, alors que l'on souhaite trouver le point de plus basse altitude existant. Ainsi, sans reformuler plusieurs fois au cours de l'algorithme le problème d'optimisation afin de diminuer le niveau de non-linéarité, la convergence est impossible, et nous verrons notamment que la connaissance du problème physique concerné est d'une importance cruciale pour formuler les bonnes hypothèses.

A présent, nous allons voir les principales méthodes de descente, qui permettent la résolution du problème d'optimisation locale. Elles s'expriment autour de la formulation suivante :

$$\begin{cases} \mathbf{x}_0 \text{ est donné} \\ \mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k d_k \end{cases} \tag{3.4}$$

où $\mathbf{x}_k$ est la suite d'itérés qui doit converger vers le minimum, α_k est le pas de descente et d_k la direction de descente.

La différence entre les méthodes que nous allons présenter réside dans le choix de la direction de descente et du pas.

Méthodes de type gradient

La première famille de méthodes consiste à utiliser le gradient $\nabla f(\mathbf{x})$ comme direction de descente. En effet, son action sur la fonctionnelle est optimum à l'ordre 1 dans le sens où :

$$(-\nabla f(\mathbf{x}))^T \nabla f(\mathbf{x}) \leq d^T \nabla f(\mathbf{x}) \ \forall d \in \mathbb{R}^n \text{ tel que } ||d|| = ||\nabla f(\mathbf{x})|| \tag{3.5}$$

Différentes stratégies sont possibles quant au choix du pas α : avec une certaine connaissance du problème, on peut simplement fixer un pas constant pour toutes les itérations. La recherche du pas peut elle même être vue comme un problème d'optimisation visant à résoudre :

$$\min_{\alpha_k > 0}(\mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k))) \tag{3.6}$$

La méthode correspondante, dite de pas optimal, ou de plus grande pente (steepest descent) , peut de prime abord paraître la plus plus intéressante dans la mesure où elle minimise le plus la fonction coût. En pratique, cette méthode est considérée comme mauvaise car elle ne converge que très lentement : par construction, deux directions de descente successives sont orthogonales, et il résulte une trajectoire de descente qui peut s'apparenter à de multiples 'zig-zag'. Pour la recherche du pas de descente, les conditions de Wolfe [37] sont reconnues comme étant les plus efficaces et sont utilisées par la plupart des algorithmes d'optimisation. Elles sont applicables pour tout type d'algorithmes de descente et s'articulent autour des deux conditions suivantes :

— La condition d'Armijo, qui permet d'éviter de choisir un pas trop grand, reformule la condition de décroissance de la fonction coût d'une itération à l'autre ainsi :

$$f(\mathbf{x}_k + \alpha_k d_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k d_k \nabla f(\mathbf{x}_k) \text{ avec } 0 < c_1 < 1 \tag{3.7}$$

— La condition de courbure évite de choisir un pas trop petit :

$$d_k^T \nabla f(\mathbf{x}_k + \alpha_k d_k) \geq c_2 d_k^T \nabla f(\mathbf{x}_k) \text{ avec } 0 < c_2 < 1 \tag{3.8}$$

Lorsque ces conditions sont utilisées ensemble, on veille aussi à respecter la condition $c_1 < c_2$ Dans la pratique, on choisit souvent $c_1 = 10^{-4}$ et $c_2 = 0.99$.

L'utilisation du gradient comme direction de descente assure la convergence locale vers l'optimum, mais la vitesse de convergence associée reste souvent assez faible. Dans la pratique, cet algorithme est apprécié pour sa simplicité de mise en œuvre, mais son utilisation est plutôt destinée aux problèmes de petite dimension. Pour des problèmes impliquant une fonction coût quadratique, de type $f(x) = x^T Ax - b^T x$, une version modifiée de la descente de gradient existe : la méthode du gradient conjugué. La direction de descente choisie utilise le gradient de l'itération courante, mais aussi ceux des itérations passées, et en utilisant le procédé d'orthogonalisation de Gram-Schmit :

$$\forall k \geq 1, d_k = -\nabla f(\mathbf{x}_k) + \sum_{j=1}^{k-1} \alpha_{k,j} d_j \text{ , avec } \alpha_{k,j} = \frac{\langle \nabla f(\mathbf{x}_k), d_i \rangle_A}{\langle d_i, d_i \rangle_A} \text{ .} \tag{3.9}$$

La force de la méthode est de pouvoir converger vers l'optimum en un nombre d'itérations inférieur à la dimension du problème. Cependant, pour des problèmes de très grande dimension, et où le calcul du gradient est plus long, cette méthode souffre toujours d'une vitesse de convergence trop faible, qui est linéaire.

Méthodes de type Newton

Les méthodes de type Newton présentent l'avantage de converger bien plus rapidement vers l'optimum. Elles utilisent la condition d'optimalité :

$$\nabla f(\mathbf{x}_{opt}) = 0 \tag{3.10}$$

La recherche des points où la dérivée s'annule est mise en œuvre avec la méthode de Newton, qui vise à résoudre l'équation $g(\mathbf{x}) = 0$, à partir de la relation de récurrence :

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \big(Dg(\mathbf{x}_k)\big)^{-1} g(\mathbf{x}_k) \tag{3.11}$$

où Dg est la matrice jacobienne de g.
La combinaison de la condition d'optimalité 3.10 et de la méthode de Newton 3.11 mène à :

$$\mathbf{x}_{k+1} = \mathbf{x}_k - H[f](\mathbf{x}_k)^{-1} \nabla f(\mathbf{x}_k) \tag{3.12}$$

Dans le cas général, la condition d'optimalité 3.10 qui est nécessaire ne devient suffisante que si le problème est convexe, ce qui n'est pas notre cas. Cela implique que l'algorithme ne différencie pas les maxima, minima ou points stationnaires, et qu'ainsi la convergence même locale n'est pas assurée comme avec les méthodes de type gradient. Par ailleurs, la construction et le stockage mémoire de la hessienne s'avère être impossible pour des problèmes de grande dimension. Pour pallier ce problème, des approximations de la hessienne sont utilisées, et on parle alors de méthodes de quasi-Newton dans ce cas.

On peut par exemple exprimer le produit hessienne-direction de descente à partir d'un développement limité :

$$\nabla f(\mathbf{x}_{k+1}) = \nabla f(\mathbf{x}_k) + H[f](\mathbf{x}_{k+1})(\mathbf{x}_k - \mathbf{x}_{k+1}) + o(\mathbf{x}_k - \mathbf{x}_{k+1}) \tag{3.13}$$

On peut alors déduire une approximation H_{k+1} de la hessienne :

$$H_{k+1}(\mathbf{x}_k - \mathbf{x}_{k+1}) = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k) \tag{3.14}$$

Pour un problème de dimension N, seules N équations sont posées par le produit matrice vecteur alors que la hessienne contient N^2 termes, et le problème est sous-déterminé. En 1965, Broyden [38] eut l'idée de choisir la hessienne H_{k+1} en fonction de la hessienne H_k calculée à l'itération précédente, en résolvant le problème des moindres carrés :

$$\min_{H_{k+1}} ||H_{k+1} - H_k||^2 \text{ sous les contraintes : 3.14 et } H_{k+1}^T = H_{k+1} \tag{3.15}$$

La contrainte de symétrie a été rajoutée car la hessienne réelle l'est aussi, ce qui réduit aussi la dimension du problème. En fonction de la norme employée pour ce problème de moindres carrés, plusieurs méthodes peuvent être déduites. La plus utilisée, nommée BFGS (qui vient du nom de ses créateurs Broyden-Fletcher-Golfarb-Shannon), utilise la norme matricielle :

$$||X||_W = ||W^{\frac{1}{2}} X W^{\frac{1}{2}}||_F \tag{3.16}$$

avec W une matrice symétrique inversible vérifiant : $\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k) = W(\mathbf{x}_{k+1} - \mathbf{x}_k)$ et où F est la norme de Frobenius qui se définit ainsi :

$$||X|| = tr(X^T X)^{\frac{1}{2}} \tag{3.17}$$

Après quelques manipulations algébriques, on peut exprimer la solution ainsi :

$$\begin{cases} H_{k+1}^{-1} = (I - \dfrac{\sigma_k y_k^T}{y_k^T \sigma_k}) H_k^{-1} (I - \dfrac{y_k \sigma_k^T}{y_k^T \sigma_k}) + \dfrac{\sigma_k \sigma_k^T}{y_k^T \sigma_k} \\ \sigma_k = \mathbf{x}_{k+1} - \mathbf{x}_k \\ y_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k) \end{cases} \tag{3.18}$$

Avec une écriture de cette forme, la construction de la hessienne est toujours nécessaire, dont le coup reste prohibitif. Dans la pratique, une version dénommée L-BFGS (pour Limited memory BFGS) est utilisée, car elle n'utilise pas directement la matrice hessienne ou son inverse, mais simplement les gradients d'un certain nombre d'itérations précédentes afin de calculer le produit entre l'inverse de la hessienne et le gradient.

On pourra remarquer que cette formulation effectue plusieurs approximations : l'action de la hessienne est construite uniquement à partir des gradients, donc de l'information d'ordre un, et le problème 3.15 est posé comme si les coefficients de la hessienne étaient identiques d'une itération à l'autre, alors qu'ils dépendent du point $\mathbf{x}_k$. Néanmoins, dans la pratique, l'utilisation de l'algorithme L-BFGS montre des résultats de convergence significativement plus rapides que pour les méthodes de type gradient, et la convergence quadratique semble vérifiée [39]. C'est cette méthode qui sera utilisée par la suite.

Calcul et considérations sur le gradient

Comme nous venons de le voir, la résolution du problème inverse passe par les algorithmes d'optimisation locale qui utilisent le gradient. Comme nous l'avons évoqué, l'utilisation d'une carte de paramètres physiques a priori hétérogène empêche tout calcul semi-analytique, et l'utilisation d'une méthode numérique de résolution est inévitable. Une solution naturelle peut être une approche de type différences finies, en considérant :

$$\nabla f(\mathbf{x}).\delta\mathbf{x} \approx \frac{f(\mathbf{x} + \epsilon\delta\mathbf{x}) - f(\mathbf{x})}{\epsilon} \tag{3.19}$$

En procédant ainsi, la fonction coût $f(\mathbf{x})$ doit être évaluée autant de fois qu'il n'y a de paramètres physiques, ce qui implique autant de simulations numériques, et rend le coût de calcul prohibitif dès lors que ces paramètres physiques sont définis dans l'espace. La méthode type différences finies a toutefois pu être mise en pratique en FWI pour obtenir le tenseur des moments de sources sismiques [40], grâce au faible nombre de paramètres à retrouver.

La méthode de l'adjoint, introduite en 1974 par Chavent [41] permet de calcul le gradient de la fonction coût en s'affranchissant du calcul de l'ensemble des dérivées de Fréchet des variables d'état, qui sont ici les données simulées, par rapport aux paramètres du modèle. Sans ces dérivées de Fréchet des variables d'état, le nombre de simulations nécessaires n'est plus proportionnel au nombre de paramètres du milieu. A la place, seules deux simulations numériques sont nécessaires. Nous allons aborder les détails de cette méthode à partir du point de vue Lagrangien, mais son interprétation est également possible à partir de la théorie de la perturbation [42].

La méthode lagrangienne part du problème d'optimisation de la fonctionnelle $J(m;y)$ sous contrainte générique $F(m;y)$, exprimé à partir des variables m, y et λ, qui représentent respectivement les paramètres du modèles, les données simulées et l'état adjoint qui sera caractérisé par la suite. On peut alors formuler le problème ainsi :

$$\mathcal{L}(m;y,\lambda) = J(m;y) + \langle F(m;y)|\lambda\rangle \tag{3.20}$$

Les conditions d'optimalité s'expriment alors :

$$\begin{cases} \partial_m \mathcal{L}(m;y,\lambda).\delta m = 0 \\ \partial_y \mathcal{L}(m;y,\lambda).\delta y = 0 \\ \partial_\lambda \mathcal{L}(m;y,\lambda).\delta \lambda = 0 \end{cases} \tag{3.21}$$

On note $M = \big(\kappa(\mathbf{x}), \rho(\mathbf{x})\big)$ les cartes des paramètres recherchés et $\{P_{syn} \in \big(\mathbb{R}^{N_{dim}} \times [0;T] \times \mathcal{F}(\Omega)^2\big)^{N_{rec}} | P_{syn}(i) = p_{syn}(\mathbf{x}_i,t,M)\}$ le vecteur rassemblant les N_{rec} observations effectuées de 0 à T secondes, $\mathbf{x}_i$ la position du i$^{\text{ème}}$ récepteur, et $\mathcal{F}(\Omega)^2$ l'espace fonctionnel (espace de fonctions a priori non continu) auquel appartient M. Nous allons exprimer le problème en fonction des paramètres $\kappa(\mathbf{x})$ et $\rho(\mathbf{x})$ mais pas de $c(\mathbf{x})$, afin que l'expression de la perturbation δM ne soit pas trop laborieuse à écrire algébriquement. Comme nous nous intéresserons par la suite à l'obtention d'une carte de vitesse $c(\mathbf{x})$, on déduira l'expression algébrique de la perturbation δc grâce à la relation $c = \sqrt{\frac{\kappa}{\rho}}$. Il est important de comprendre que le choix des paramètres est ici motivé par l'objectif de la reconstruction de $c(\mathbf{x})$, et que d'autres paramètres peuvent être choisis en fonction du problème considéré.

Pour la FWI en acoustique linéaire, la formulation du problème d'optimisation sous contrainte de la forme 3.20, obtenu à partir de 3.1 et 3.3 est alors :

$$\begin{aligned} &\mathcal{L}_{FWI}(M;P_{syn},\lambda(\mathbf{x},t)) = \frac{1}{2}\sum_{i=1}^{N_{rec}} \int_0^T \big|p_{obs}(\mathbf{x}_i,t) - p_{syn}(\mathbf{x}_i,t,M)\big|^2 \mathrm{dt} + \\ &\int_0^T \int_\Omega \lambda.\Big(\kappa(\mathbf{x})^{-1} p_{syn} - \nabla\ (\rho(\mathbf{x})^{-1}\nabla\ (\int_0^T \int_0^T p_{syn}\mathrm{dtdt})) - \sum_{s=1}^{N_s} \kappa(\mathbf{x}_s)^{-1} f_s(t)\Big)\mathrm{d\mathbf{x}'dt} \end{aligned} \tag{3.22}$$

En appliquant les conditions d'optimalité 3.21 à la FWI en acoustique linéaire dont le Lagrangien est à présent déterminé (Eq. 3.22), on peut alors déduire l'expression des gradients. Il convient d'exprimer tout d'abord le coefficient de Lagrange $\lambda(\mathbf{x},t)$, en combinant 3.22 avec la deuxième équation de 3.21, et en notant $\Delta p_i = \big|p_{obs}(\mathbf{x}_i,t) - p_{syn}(\mathbf{x}_i,t,M)\big|$:

$$\begin{aligned} \partial_{P_{syn}} \mathcal{L}_{FWI}(M;P_{syn},\lambda).\delta P_{syn} &= \sum_{i=1}^{N_{rec}} \int_0^T \Delta p_i \delta p_{syn}(\mathbf{x}_i,t,M)\mathrm{dt} \\ - \int_0^T \int_\Omega \lambda.\Big(\kappa(\mathbf{x})^{-1}\delta p_{syn} - \nabla\ (\rho(\mathbf{x})^{-1}\nabla\ (\int_0^T \int_0^T \delta p_{syn}\mathrm{dtdt}))\Big)\mathrm{d\mathbf{x}dt} \end{aligned} \tag{3.23}$$

En considérant la relation : $\delta p_{syn} = -\partial_t^2 \delta\Phi$, on peut déduire après plusieurs manipulations algébriques (plus de détails sont fournis dans [43]) :

$$\int_0^T \int_\Omega \Big(\kappa(\mathbf{x})^{-1}\partial_t^2\lambda - \nabla.(\rho(\mathbf{x})^{-1}\nabla\lambda)\Big)\delta\Phi \mathrm{d\mathbf{x}'dt} = -\int_0^T \int_\Omega \sum_{i=1}^{N_{rec}} \partial_t^2 \Delta p_i \delta(\mathbf{x}' - \mathbf{x}_i)\delta\Phi \mathrm{d\mathbf{x}'dt} \tag{3.24}$$

Cette expression suppose que Δp_i et $\partial_t \Delta p_i$ soient nuls à l'instant initial et à l'instant final, ce qui peut être obtenu en fenêtrant le signal.

Finalement, l'équation adjointe 3.24 qui détermine entièrement le coefficient de Lagrange $\lambda(\mathbf{x}, t)$ n'est autre que la rétropropagation de la dérivée seconde de la différence entre données simulées et données réelles. Par la suite, nous noterons :

$$\Phi^\dagger(\mathbf{x}, t) = \lambda(\mathbf{x}, T - t) \tag{3.25}$$

En utilisant l'équation 3 de 3.21 et 3.22, on retombe sur l'équation d'état 3.3.

En absence d'atténuation, en vertu du caractère auto-adjoint de l'opérateur décrivant la propagation d'onde, les équations 3.3 et 3.24 sont les mêmes à la seule différence du terme source. En ce sens, seules deux simulations numériques de propagation d'onde sont nécessaires pour calculer le gradient par événement enregistré.

A présent, on peut s'intéresser à l'expression des gradients, qui est obtenue à partir de l'équation 1 de 3.21 appliquée à 3.22, qui nous conduit à :

$$\partial_M \mathcal{L}_{FWI}(M; p_{syn}, \lambda).\delta M = \int_\Omega \Big(\big(\ln \delta\kappa(\mathbf{x})^{-1} K_\kappa(\mathbf{x}) + \ln \rho(\mathbf{x})^{-1} K_\rho(\mathbf{x}) \big) \mathrm{d}\mathbf{x} \tag{3.26}$$

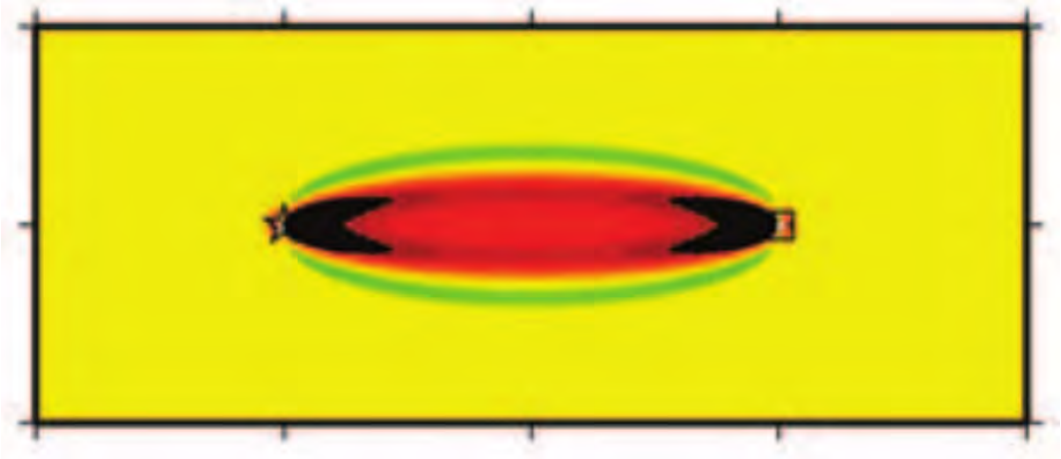
où $\ln \delta\mathbf{x} = \frac{\delta \mathbf{x}}{\mathbf{x}}$ et :

$$\begin{cases} K_\kappa(\mathbf{x}) = -\sum_{s=1}^{N_s} \int_0^T \kappa(\mathbf{x})^{-1} \partial_t^2 \Phi_s^\dagger(\mathbf{x}, t) \partial_t^2 \Phi_s(\mathbf{x}, t) \mathrm{dt} \\ K_\rho(\mathbf{x}) = -\sum_{s=1}^{N_s} \int_0^T \rho(\mathbf{x})^{-1} \nabla \Phi_s^\dagger(\mathbf{x}, t) . \nabla \Phi_s(\mathbf{x}, t) \mathrm{dt} \end{cases} \tag{3.27}$$

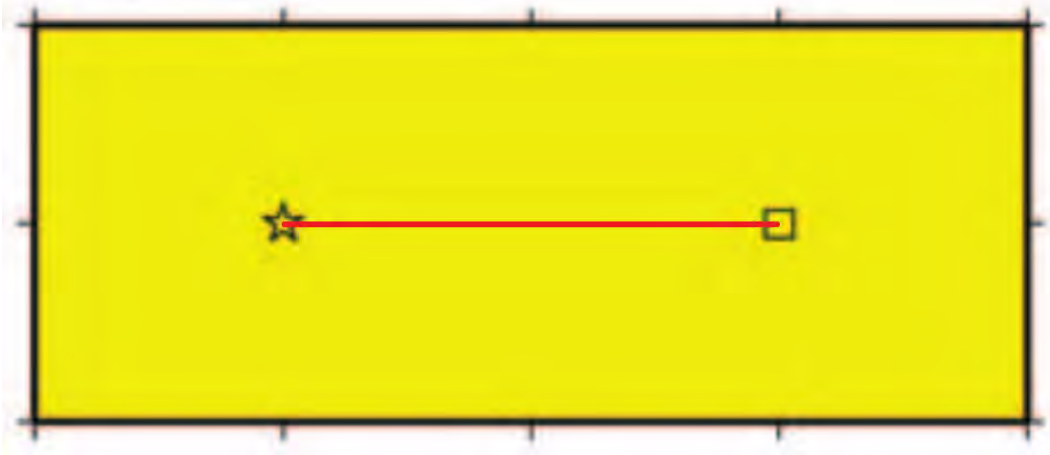
Ainsi, les gradients exprimés dans 3.27 peuvent être vus comme la convolution des deux champs direct et adjoint, ce qui peut rappeler l'expression de l'énergie topologique. On peut s'apercevoir que la refocalisation, due à l'utilisation de l'opérateur de propagation, et la corrélation des champs proviennent du déroulement mathématique de la méthode de l'adjoint, et sont indépendantes du paramètre utilisé. De fait, les propriétés de l'énergie topologique découlent de l'utilisation de la méthode de l'adjoint, et non de l'utilisation spécifique des variables d'optimisation topologique. Dans ces expressions, l'expression de la perturbation au niveau des termes sources n'a pas été prise en compte, pour des raisons de sensibilité de la méthode en ces points.

Remarquons par ailleurs que si le champ direct est exactement le même, à savoir si un même événement est enregistré par deux géophones ou deux transducteurs, alors par linéarité par rapport au champ adjoint la somme des gradients est le gradient de la somme. Ceci nous autorise à utiliser une seule simulation adjointe pour rétropropager l'information de ces deux géophones ou transducteurs, et le calcul du gradient est alors indépendant du nombre de voies utilisé en réception, ce qui assure l'utilisation de seulement deux simulations numériques, une directe et une adjointe. En revanche, si l'on souhaite utiliser les données provenant de deux sources distinctes, les deux champs directs sont différents, et ainsi quatre simulations, deux directes et deux adjointes, sont nécessaires pour obtenir la somme des gradients. Alors, le nombre de simulations numériques est égal au double du nombre d'événements N_s enregistrés. Dans la pratique, nous verrons que la multiplication du nombre de sources est intéressant dans la mesure où la propagation d'onde n'est pas la même et que l'information reçue d'un point donné du domaine dépend de la configuration source-récepteurs. De plus, cela permet aussi de compenser l'approximation de Born, en moyennant en un point différentes contributions toutes affectées par cette hypothèse.

Dans la littérature géophysique, le terme noyau, ou kernel, est employé pour désigner le gradient tel que calculé par la simulation numérique. Le gradient désigne quant à lui la direction de descente utilisée dans l'algorithme d'optimisation locale, et est égal à la somme des kernels de chaque événement, qui peut être pondérée selon la compréhension du problème, et modifié pour linéariser le problème comme nous le verrons. Nous utiliserons la même convention de nommage par la suite. Enfin, comme évoqué précédemment, on peut considérer une paramétrisation différente de $\big(\rho(\mathbf{x}), \kappa(\mathbf{x})\big)$, comme $\big(\rho(\mathbf{x}), c(\mathbf{x})\big)$, ce qui nous conduit alors aux expressions suivantes :



(a) Kernel à fréquence finie, inspiré de [44]



(b) Kernel avec l'approximation géométrique

FIGURE 3.1 – Forme des kernels en milieu homogène infini selon le mode de calcul. En lancer de rayons, seuls les points situés sur la ligne reliant l'émetteur et le récepteur influencent les données.

$$\begin{cases} K_c(\mathbf{x}) = 2K_\kappa(\mathbf{x}) \\ K_{\rho'}(\mathbf{x}) = K_\rho(\mathbf{x}) + K_\kappa(\mathbf{x}) \end{cases} \tag{3.28}$$

Pour mieux comprendre cette notion de noyau, il peut être intéressant de voir à quoi ressemble un noyau source-récepteur dans un milieu homogène. Un noyau illustre en particulier l'influence d'une modification locale d'un paramètre sur les données, en fonction du signal source utilisé, et vice-versa. En fonction de la fidélité de l'opérateur de propagation utilisé, cette influence peut fortement varier. La figure 3.1 représente les noyaux de sensibilité pour une méthode de type lancer de rayons, et la méthode de type éléments finis que nous avons utilisé. Comme on peut le voir, pour le modèle de type lancer de rayons, qui revient à faire l'hypothèse d'un signal émis de fréquence infini, seuls les paramètres situés sur la ligne reliant la source au récepteur influencent les données enregistrées. A contrario, une surface ou un volume entier du domaine peut être responsable d'une perturbation dans les données, si un modèle qui incorpore les effets de fréquence finie est utilisé. Ces noyaux, communément appelés 'banana-donut kernels', et particulièrement mis en avant dans les travaux de Tromp en 2005 [44], motivent l'utilisation de modèles très précis de propagation pour le calcul du gradient. L'aspect 'banana' est typique de l'effet de fréquence finie, et on peut montrer que la largeur du noyau est proportionnel à $\sqrt{\lambda L}$, où L est la distance source-récepteur. Pour des basses fréquences, cela signifie qu'une petite perturbation du modèle est aussi importante dans une très large zone du domaine. Pour des fréquences tendant vers l'infini, la zone critique

d'influence d'une perturbation se cantonne à une toute petite zone qui tend à se confondre avec la ligne source-récepteur, qui était la solution du modèle de propagation de type lancer de rayons. L'aspect 'donut', un peu plus surprenant et qui découle de l'approximation de Born, montre la faible influence de la zone située sur la ligne source-récepteur. Enfin, comme on peut le voir avec une seule paire source-récepteur, de nombreuses parties du domaine n'influencent pas la propagation d'onde, et justifient l'utilisation d'un plus grand nombre de sources et de récepteurs, de façon à augmenter d'une part la couverture spatiale du domaine, et d'autre part la redondance de l'information en un point donné du domaine.

3.2 Principaux facteurs influant sur la convergence et la rapidité de la méthode

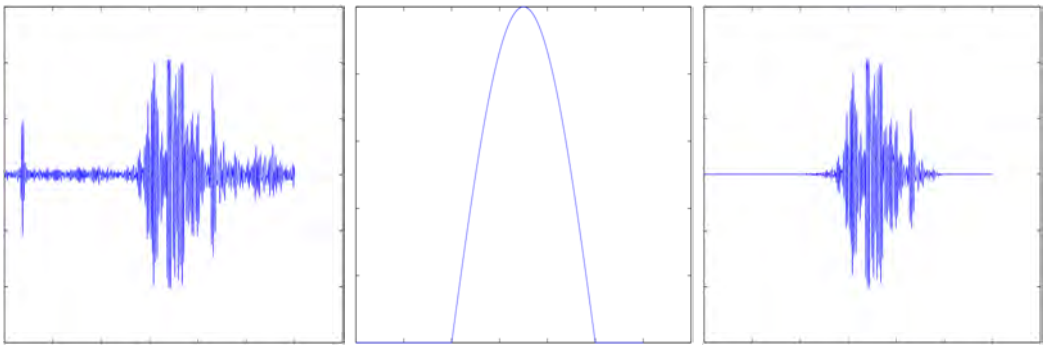
Comme nous l'avons évoqué, le problème d'optimisation posé par la FWI est fortement non convexe, et ne peut pas être résolu sans un certain traitement des données ou une reformulation du problème. En remarquant les points qui influencent la méthode, nous pourrons alors déduire des stratégies pour forcer d'abord sa convergence, puis sa vitesse de convergence. Bien que nous ayons présenté le problème posé par la FWI comme étant la minimisation directe de l'écart entre données réelles et données simulées (appelée forme d'onde ou waveform), la plupart des stratégies employées simplifient le problème en considérant une version filtrée de ces données, où en transformant leur nature même. En toute rigueur, le terme Full Waveform Inversion réfère à l'inversion de la totalité du contenu des données enregistrées. Dans la littérature, on utilise le terme de tomographie adjointe quand il s'agit de transformer les données avant de les traiter, où d'utiliser une fonction coût autre que la forme d'onde. Dans la pratique et dans la mesure où voir dès le début le problème de façon 'FWI' n'est pas possible pour la convergence de la méthode, nous utiliserons ici le terme Full Waveform Inversion dans le sens où au moins à la fin du processus de minimisation, la totalité des données aura effectivement été inversé.

Filtrage des données

Le filtrage des données s'oppose ici à l'aspect 'Full' de la méthode, car un filtrage quel qu'il soit engendre une altération et une perte d'information, qui n'aura pas été inversée à proprement parler. Le filtrage le plus classique consiste à appliquer un fenêtrage temporel. Dans un premier temps, il peut être intéressant de restreindre l'inversion aux premiers fronts d'onde reçus, qui sans considérations sur la configuration de la géométrie d'acquisition devraient probablement davantage respecter l'approximation de Born. En effet, des arrivées plus tardives sont plus à même d'être les signatures de fronts d'onde réfléchis ou réfractés plusieurs fois. Une stratégie classique consiste à ne garder les premiers fronts d'ondes, qui une fois inversés engendrent des variations dans le modèle qui peuvent permettre d'avoir une meilleure simulation des fronts d'onde suivants que celle qui aurait été faite sans avoir d'abord traité les premiers fronts d'onde. De proche en proche, le signal peut être progressivement entièrement inversé. Par ailleurs, des expérimentations pratiques ont montré qu'un fenêtrage coupant brusquement le signal pouvait engendrer des oscillations non désirées. Pour éviter cela, la fenêtre temporelle choisie est apodisée en la multipliant par un signal nul aux bornes de la fenêtre et maximum en son centre, comme un sinus sur $[0, \pi]$, montré figure 3.2.

Lors de la minimisation de l'écart entre données réelles et données simulées, un phénomène important est le cycle skipping, ou saut de cycle. On peut le voir illustré figure 3.3. Concrètement, il correspond à une mauvaise identification d'une oscillation entre les données réelles et simulées, et le processus d'inversion va tenter de faire correspondre l'oscillation numéro n des données simulées à l'oscillation m des données réelles. Alors, l'algorithme converge vers un minimum local qui n'est pas celui recherché. Dans l'espace du modèle, la carte de paramètres correspondant peut être très différente de la carte recherchée, comme on peut le voir sur la figure 3.4.

Mathématiquement, on peut écrire la condition de saut de cycle qui équivaut au fait que le



(a) Signal RF original (b) Masque appliqué (c) Signal réellement traité

FIGURE 3.2 – Transformation du signal original, en vue d'une inversion. Dans cet exemple la zone d'intérêt est sélectionnée, manuellement ou de façon automatisée, et un masque en est déduit et appliqué aux données.

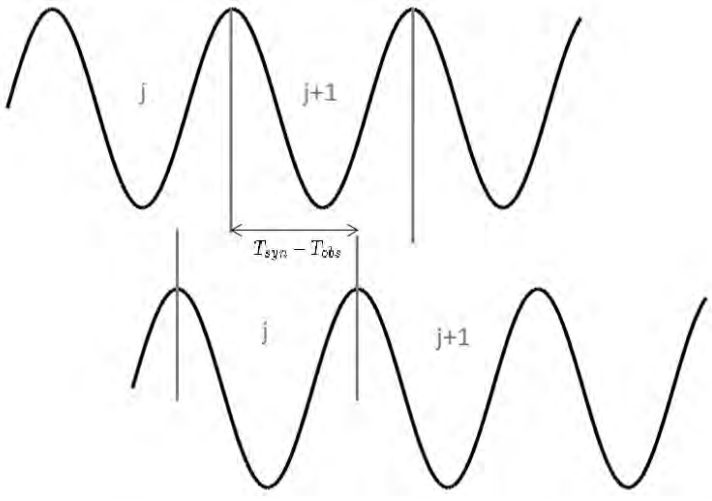
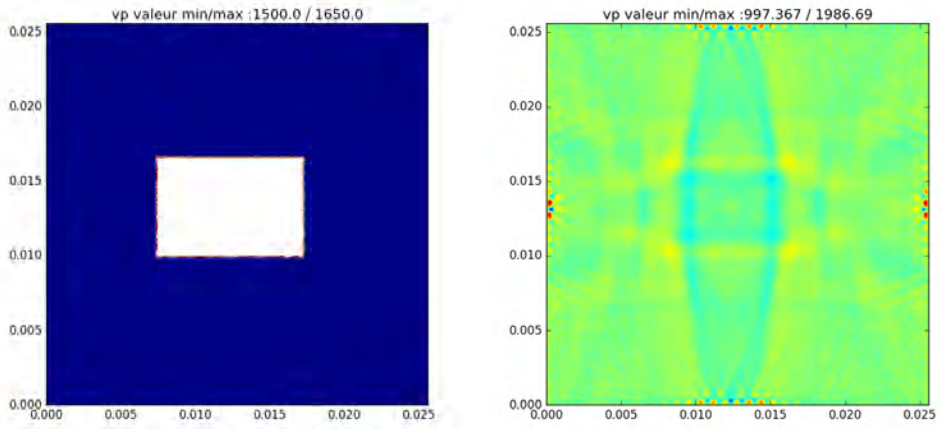
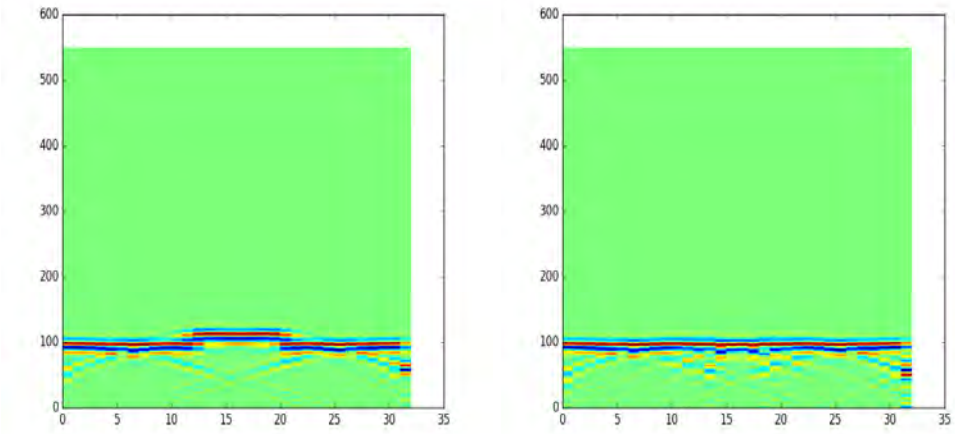


FIGURE 3.3 – Schéma du cycle skipping. Lors du processus d'inversion, l'oscillation j des données réelles est identifiée comme l'oscillation $j + 1$ des données simulées, conduisant à une divergence de la solution.



(a) Modèle réel

(b) Modèle inversé



(c) Données réelles

(d) Données simulées après convergence

FIGURE 3.4 – Exemple d'une inversion divergente à cause du cycle skipping. Ici, l'onde est émise du côté gauche, et enregistrée du côté droit.

front d'onde simulé ne doit pas avoir un retard de plus d'une demi-période $\frac{T}{2}$ sur le front d'onde réel, soit :

$$|T_{obs} - T_{syn}| < \frac{T}{2} \tag{3.29}$$

Dans la pratique, il n'est pas toujours possible de savoir si cette condition est respectée. Alors, il convient de considérer les a priori que l'on peut avoir sur le milieu réel : quelles sont les vitesses maximale et minimale possibles dans la réalité ? La réponse à cette question permet, en connaissance des dimensions du modèle, de se donner une borne supérieure quant à l'incertitude sur l'écart un modèle numérique et modèle réel, et ainsi de deviner quel sera l'écart $|T_{obs} - T_{syn}|$ maximal qu'il est possible d'avoir. En fonction de cette quantité, s'il est possible de choisir le signal émis, il convient alors de choisir un signal dont la bande passante implique des périodes qui respectent l'inégalité 3.29. On veillera également à ne pas choisir des fréquences trop basses, ce qui se traduirait par une résolution spatiale très faible dans le modèle inversé. Si le contrôle du signal n'est pas possible, un filtrage passe bas peut permettre de respecter cette condition. Une fois les basses fréquences inversées, une injection progressive des hautes fréquences permet d'augmenter la résolution spatiale, sans mettre en péril la convergence. Ici aussi, le fenêtrage temporel peut être utilisé pour réduire les risques de saut de cycle, dans la mesure où les fronts d'onde secondaires ont traversé le milieu pendant plus longtemps, et donc l'écart de temps entre les deux types de données est susceptible d'être plus élevé.

Enfin, le filtrage des données peut être utilisé pour sélectionner des fronts d'onde dont l'arrivée était prévue et estimée dans les données simulées. En géophysique, avec une connaissance a priori sur la position de la source, on sait que la première arrivée correspond à l'onde de compression, une autre à l'onde de cisaillement, et d'autres fronts tels que les ondes réfléchies sur la croûte terrestre, ainsi que les conversions de mode. Pour ces fronts d'onde, le modèle initial est généralement construit de sorte qu'ils soient déjà en partie correctement simulés, pour que la fonction compare réellement deux fronts d'onde et tire vraiment parti de l'information contenue dans les données réelles. A l'inverse, si un front d'onde totalement inconnu apparaît seulement dans les données réelles, il peut être judicieux de ne pas le prendre en compte dans un premier temps, et regarder si l'inversion des autres fronts d'onde fera apparaître par la suite la signature d'un nouveau front dans les données simulées, que l'on est alors en mesure de vraiment comprendre et d'interpréter pour l'inversion.

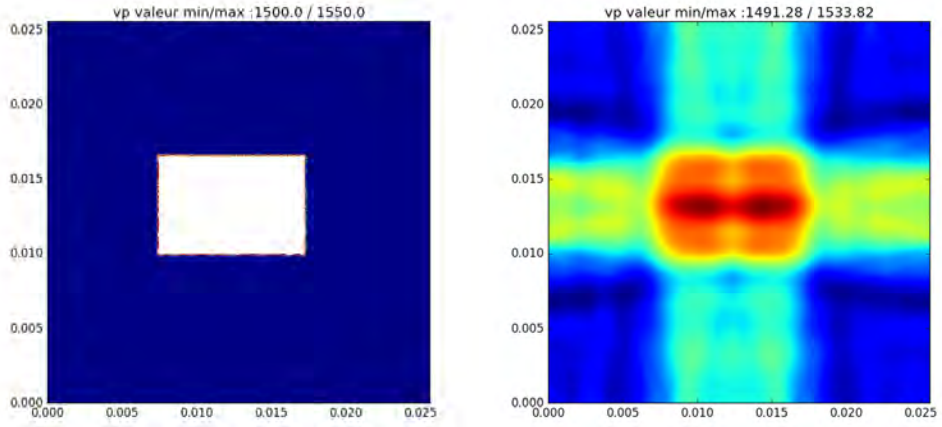
Fonction coût

Au delà du simple filtrage des données, il est aussi possible de redéfinir le problème d'optimisation à résoudre, et ne pas nécessairement chercher à inverser les données elles-mêmes, mais une version transformée des données avec une interprétation physique différente. Dans ce cas, on ne parle plus vraiment d'inversion de type 'waveform' ou forme d'onde. Dans le contexte de la géophysique, l'amplitude de l'onde est une information bien souvent difficile à exploiter : les propriétés d'atténuation du sous-sol peuvent être mal connues, voire non modélisées dans le modèle numérique de propagation, ou l'écart entre modèle numérique et réel peut être trop important pour que la convergence soit possible. Quand l'exploitation de cette information d'amplitude n'est pas possible, une fonction coût de type temps de vol, ou traveltime, peut être préférée et s'écrit ainsi :

$$\min f(m) = \frac{1}{2} \sum_{i=1}^{N_{rec}} \left| T_{syn}(\mathbf{x}_i; m) - T_{obs}(\mathbf{x}_i) \right|^2 \tag{3.30}$$

Un écart de temps de vol entre deux fronts d'onde peut être obtenu en utilisant la fonction de corrélation de deux signaux S_1 et S_2 :

$$R(t) = \int_{t_1}^{t_2} S_1(\tau) S_2(\tau - t) \mathrm{d}\tau \tag{3.31}$$



(a) Modèle objectif (b) Modèle inversé

FIGURE 3.5 – Résultat d'inversion en traveltime avec convergence à l'un des optima globaux. Le modèle obtenu, différent de modèle recherché, illustre la non-unicité de la solution.

L'utilisation de cette fonction implique cependant une sélection, parfois manuelle, des fronts d'onde à corréler. En revanche, une fois cette opération menée, le procédé n'est pas soumis à la condition de saut de cycle, autorisant l'utilisation de fréquences plus élevées. Comme le choix de la fonction coût ne modifie pas l'équation d'onde, et si l'on se base sur la dérivation de l'équation 3.22, l'expression mathématique du noyau par rapport à un paramètre donné est la même qu'avec une fonction coût de type waveform. En dérivant le premier terme de l'équation 3.22 qui n'est plus le même pour respecter la condition d'optimalité, le terme source du champ adjoint se trouve quant à lui modifié. En acoustique linéaire, on a :

$$f^{\dagger}(t) = \sum_{i=1}^{N_{rec}} \left(T_{syn}(\mathbf{x}_i) - T_{obs}(\mathbf{x}_i)\right)\partial_t^3 p_{syn}(\mathbf{x}_i, t) \tag{3.32}$$

A condition d'être capable d'opérer la sélection des différents fronts d'onde, la fonction coût de type traveltime converge plus facilement qu'une fonction coût de type waveform. En revanche et à cause de l'information retirée, l'unicité de la solution est encore moins probable, comme on peut le voir sur l'exemple 3.5. Dans ce cas ci, on peut voir que la fonction coût décroît jusqu'au zéro absolu, ce qui signifie que l'on a convergence globale pour le problème ainsi posé. Toutefois, on peut voir que le modèle inversé ne correspond pas au modèle réel, à cause de la faible quantité d'information exploitée. De plus, la différence avec fonction coût de type waveform n'est pas nulle comme on peut le constater, à cause d'écarts d'amplitude entre les deux jeux de données. En augmentant le nombre de sources, il est possible d'augmenter la quantité d'information disponible et d'atteindre un meilleur résultat.

Il est aussi possible, comme démontré par Luo [45], d'utiliser des fonctions coût hybrides qui font intervenir à la fois les noyaux en waveform et en traveltime, en pondérant au début fortement la contribution du traveltime, afin de bénéficier de sa facilité de convergence, puis d'augmenter progressivement la contribution du terme de waveform, afin d'avoir une meilleure résolution spatiale grâce au supplément d'informations contenues. Une autre fonction coût, la fonction enveloppe, est aussi assez populaire. Sans nécessité de sélectionner les fronts d'onde, il est possible d'alléger la condition de saut de cycle en comparant l'enveloppe des signaux réels et observés. Dans l'équation 3.29, au lieu de considérer la demi période de la fréquence associée, on utilise la demi durée du front d'onde, qui est très souvent une dizaine de fois plus grand. Arrivé à convergence avec la fonction coût de type enveloppe, il peut être judicieux de passer à nouveau à une fonction waveform, car la condition de saut de cycle associée devrait être respectée grâce

à la mise à jour du modèle. D'autres fonctions coût peuvent être définies, ce qui constitue un champ de recherche à part entière dans la communauté géophysique. La fonction coût de type double différences [46] a montré des résultats intéressants, et plus récemment le calcul de phase instantanée [47] est un sujet actif.

Préconditionnement

Le préconditionnement, dont le concept provient de l'optimisation, consiste à trouver un changement de variables pour lequel le problème d'optimisation est mieux conditionné, c'est-à-dire pour lequel une perturbation des paramètres du modèle affecte avec la même amplitude les données simulées, sans que certains paramètres soient plus critiques que d'autres. Pour le problème $Ax = b$, le préconditionnement résout un problème équivalent : $P^{-1}Ax = P^{-1}b$. Il existe des préconditionneurs génériques considérant le problème de façon purement mathématique, comme la SSOR (Symetric Successive OverRelaxation) ou la méthode de Cholesky incomplète. Néanmoins, leur efficacité reste faible par rapport à un préconditionneur spécifique à l'équation du problème. Comme nous l'avons vu avec les noyaux de sensibilité, la configuration source-récepteur rend l'influence de certains paramètres plus importante que pour d'autres, ce qui se traduit physiquement par le fait que certaines zones sont plus insonifiées que d'autres. Ainsi, des préconditionnements quantifiant l'illumination de la source ont été définis :

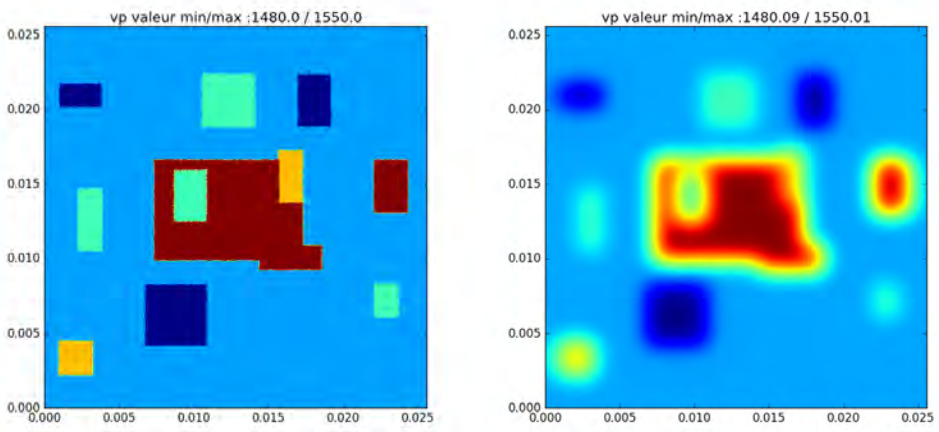
$$P_1(\mathbf{x}) = \int_0^T \partial_t^2 \Phi(\mathbf{x}, t) \partial_t^2 \Phi(\mathbf{x}, t) \mathrm{dt} \tag{3.33}$$

$$P_2(\mathbf{x}) = \int_0^T \partial_t^2 \Phi(\mathbf{x}, t) \partial_t^2 \Phi^\dagger(\mathbf{x}, t) \mathrm{dt} \tag{3.34}$$

Le premier permet de simuler une distribution davantage homogène de l'énergie du champ direct dans le milieu, et le second prend aussi cet effet en compte pour le champ adjoint. Comme il est possible que certaines zones ne soient absolument pas insonifiées, il convient de retravailler ces formules en fonction de la configuration choisie. De même, on peut penser que des zones très faiblement insonifiées n'impacteront pas fortement les données, et qu'une application brute de ces préconditionneurs risque d'augmenter significativement le niveau de bruit. Ici encore, en fonction de la compréhension a priori du problème, il est possible de privilégier certaines portions du domaine, ou de fixer un certain niveau de seuil qui peut varier d'une expérimentation à l'autre. Remarquons par ailleurs que les algorithmes de descente de type Newton peuvent être vus comme des algorithmes de descente de gradient, préconditionnés par la matrice hessienne.

Régularisation du problème non linéaire

Afin de minimiser les risques d'être piégé dans un minimum local loin de la solution globale, plusieurs procédés de régularisation existent. Il se formulent souvent en rajoutant un terme à la fonction coût, souvent appelé terme de pénalisation. Contrairement aux préconditionneurs, qui servent à augmenter la vitesse de convergence mais ne changent pas le problème, les termes de pénalisation modifient le minimum vers lequel l'algorithme va converger. Leur utilité est donc d'ajouter des contraintes au problème, afin de diminuer la possible non-unicité de la solution. La méthode de régularisation la plus populaire, dite de Tikhonov, utilise un terme de pénalisation du type $||\Gamma(m - m_0)||^2$, que l'on peut comprendre comme un terme d'inertie qui s'oppose à des variations trop brusques des paramètres m, lesquelles conduisent souvent vers un minimum local. En fonction de la compréhension du problème il est ici aussi possible de pondérer différemment les données, en ajustant les composantes de la matrice Γ. Dans [48], un terme de pénalité utilisant la norme du laplacien $\frac{\eta}{2}||\Delta m||^2$ a été introduit pour un problème de FWI. Le gradient associé à ce terme est $\eta\Delta^2 m$, ce qui donne une version lissée du gradient original. Avec la pondération de l'importance de ce terme laplacien en modifiant la constante η, on peut contrôler le niveau de lissage associé au gradient, et par exemple le diminuer au fil des itérations, une fois que le modèle courant est plus proche du modèle réel.



(a) Image originale

(b) Image après lissage Gaussien

FIGURE 3.6 – Application d'un lissage Gaussien à une carte de vitesse à 2D.

Un autre terme de régularisation possible consiste en la minimisation des variations totales (total variations), qui s'écrit dans le cas à 2D ainsi :

$$R(\mathbf{x}) = \sum_{i=1}^{N_x} \sum_{j=1}^{N_z} \sqrt{|x_{i+1,j} - x_{i,j}|^2 + |x_{i,j+1} - x_{i,j}|^2} \tag{3.35}$$

Cette formulation est intéressante car elle permet de minimiser le caractère oscillatoire du gradient, qui est souvent à l'origine de nombreuses non linéarités. De bons résultats ont pu être obtenus [49] dans le contexte de la géophysique avec la reconstruction d'interfaces marquant un assez fort contraste de propriétés physiques, en utilisant notamment la méthode de Split-Bregman [50]. Cette méthode, non exploitée dans cette thèse, pourrait être utile à l'échelle ultrasonore, et en particulier en imagerie médicale pour la reconstruction de la forme des os du corps humain.

Comme nous l'avons évoqué, nous avons pu observer dans la pratique que les kernels calculés possèdent un caractère oscillatoire. En effet, si on se base sur l'expression générique du gradient, on retrouve systématiquement une corrélation temporelle entre champ direct et champ adjoint. Comme ces deux champs oscillent, cette propriété se retrouve dans leur convolution. Du point de vue du problème d'optimisation, ces oscillations augmentent le risque de tomber dans un minimum local, et accentuent la complexité de la trajectoire des fronts d'onde. Dans la réalité, le modèle réel n'a a priori aucune raison d'osciller, et ces oscillations sont particulièrement difficiles à retirer une fois transmises par le gradient. Pour éviter ces oscillations, nous appliquons un lissage gaussien illustré sur la figure 3.6 à la somme des kernels obtenus, comme suit :

$$\hat{K}(\mathbf{x}) = \frac{\int_\Omega K(\mathbf{x}') \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{(\mathbf{x}-\mathbf{x}')^2}{2\sigma^2}} \mathrm{d}\mathbf{x}'}{\int_\Omega \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{(\mathbf{x}-\mathbf{x}')^2}{2\sigma^2}} \mathrm{d}\mathbf{x}'} \tag{3.36}$$

où Ω est le domaine sur lequel est calculé l'inversion. Dans la pratique, cette normalisation évite d'une part de sous-évaluer les points situés près des coins du domaine, car la valeur de $\int_\Omega \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{(\mathbf{x}-\mathbf{x}')^2}{2\sigma^2}} \mathrm{d}\mathbf{x}'$ y est minimale, et d'autre part de s'adapter à de très petites valeurs σ, sans craindre une mauvaise intégration numérique.p

Il est possible d'ajuster la valeur de σ pour contrôler le niveau de lissage. Pour les premières itérations, qui ont une influence importante sur la convergence vers l'optimum, un lissage fort peut s'avérer indispensable. Puis, au fil des itérations, la décroissance du rayon de lissage permettra de révéler plus de détails. Il peut être utile de ne pas descendre en dessous de $\frac{\lambda}{2}$ pour éviter de

retrouver des oscillations dans le modèle, même si la solution semble se diriger vers l'optimum global.

3.3 Simulation numérique de la propagation d'onde

En vue de la modélisation de la propagation d'onde, nécessaire à l'évaluation de la fonction coût et du gradient, un modèle numérique est indispensable, dans la mesure où les paramètres physiques peuvent varier localement. Comme évoqué, nous avons choisi un modèle précis de type éléments finis, afin de tenir compte des effets de fréquence finie sur les kernels, lesquels se révèlent parfois cruciaux pour la convergence de la méthode. Ce choix se paie par un nombre d'opérations arithmétiques plus important, et donc un temps de calcul plus élevé. Cette partie présente quelques caractéristiques du code de calcul que nous avons utilisé, le code Specfem [51, 52]. Nous allons présenter la méthode des éléments finis spectraux que ce dernier utilise, puis nous verrons dans les grandes lignes l'implémentation GPU du code, qui est ici aussi cruciale pour pouvoir espérer obtenir un résultat dans un délai acceptable. Pour des soucis de concision, seuls les détails concernant la propagation acoustique en milieu fluide seront abordés. Cependant, le code Specfem est aussi capable de simuler la propagation en milieu élastique, anisotrope [53], avec atténuation ou encore dans les milieux poreux [54] et toute combinaison de ces milieux. Il est également capable de simuler des milieux infinis avec l'implémentation des couches absorbantes parfaitement adaptées par convolution (ou C-PML) [55] ou encore de prendre en compte l'influence de la gravité [56].

La méthode des éléments finis spectraux

Très semblable aux éléments finis conventionnels, la méthode des éléments finis spectraux possède quelques propriétés supplémentaires, utiles à la fois pour la précision numérique et pour une implémentation efficace. La principale différence entre les deux réside sur le choix des fonctions de base, de type Lagrange et d'ordre plus élevé pour la version spectrale, ainsi qu'une intégration numérique utilisant la règle de quadrature de Gauss-Lobatto-Legendre. D'abord utilisés en mécanique des fluides [57, 58], les éléments spectraux se sont progressivement étendus au calcul de propagation d'onde, acoustique ou élastique. Dans la suite, nous considérerons la résolution de l'équation générique linéaire, dépendant de l'espace et du temps :

$$F(t, \mathbf{x}, U(\mathbf{x}, t), \frac{\partial U}{\partial \mathbf{x}}, \frac{\partial U}{\partial t}, \frac{\partial^2 U}{\partial \mathbf{x}^2}, \frac{\partial^2 U}{\partial t^2}) = f \tag{3.37}$$

Pour résoudre le problème, celui-ci va être discrétisé spatialement, puis temporellement.

Discrétisation spatiale

Tout d'abord, le problème numérique est exprimé à partir de la forme faible de l'équation à résoudre :

$$\int_\Omega F(t, \mathbf{x}, U(\mathbf{x}, t), \frac{\partial U}{\partial \mathbf{x}}, \frac{\partial U}{\partial t}, \frac{\partial^2 U}{\partial \mathbf{x}^2}, \frac{\partial^2 U}{\partial t^2}) w \mathrm{d}\mathbf{x} = \int_\Omega f w \mathrm{d}\mathbf{x} \tag{3.38}$$

Cette formulation intégrale, très utilisée par les méthodes numériques, permet d'inclure naturellement la géométrie du domaine grâce à l'intégrale et au vecteur test w. Comme pour toute méthode de type éléments finis, le milieu réel est représenté comme étant l'union d'un certain nombre de surfaces ou volumes disjoints, que l'on nomme éléments. S'il existe une grande variété d'éléments pour des méthodes conventionnelles, celle des éléments finis spectraux privilégie les quadrilatères à 2D, et les hexaèdres à 3D. Chaque élément est lui-même défini comme étant la donnée de N_{geom} nœuds géométriques $\mathbf{x}_i$ et de plusieurs fonctions de forme $N_i(\boldsymbol{\xi})$. Un système de coordonnées locales $\boldsymbol{\xi} = (\xi, \eta)$ à 2D ou $\boldsymbol{\xi} = (\xi, \eta, \zeta)$ à 3D est défini pour l'ensemble des points

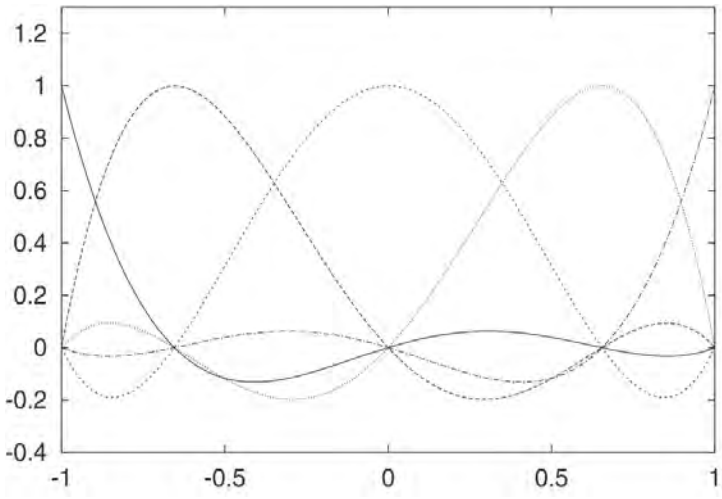


FIGURE 3.7 – Représentation des polynômes de Lagrange choisis comme fonctions de base, pour l'intégration sur un segment à une dimension. Ils ont été choisis pour valoir soit 0 soit 1 au niveau des points d'intégration, dans le but d'obtenir une matrice de masse diagonale. Image issue de [59].

situés à l'intérieur de l'élément, et le passage du local au global s'exprime comme suit :

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_{i=1}^{N_{geom}} \mathbf{x}_i N_i(\boldsymbol{\xi}) \tag{3.39}$$

Dans notre implémentation, on utilise 4 ou 9 nœuds géométriques à 2D, et 8 ou 27 à 3D. Les fonctions de forme sont quant à elles le produit de polynômes de Lagrange d'ordre 2, exprimées sur chacune des 2 ou 3 dimensions. Leur expression analytique exacte est explicitée dans [60]. Pour que le problème soit bien posé, il est important de s'assurer que la transformation des coordonnées globales vers les coordonnées locales et son inverse soient bien définies en tout point. Pour cela, on s'assure que le jacobien associé à la transformation soit non nul partout. Le calcul du jacobien se fait de façon analytique.

A ce stade, on suppose que le maillage est correctement défini en tant qu'union de N_{spec} éléments, eux-mêmes définis comme la donnée de nœuds géométriques et de fonctions de formes. Afin d'exploiter la formulation faible 3.38, une méthode de Galerkin continue est utilisée. Si l'équivalence entre formulation forte et faible n'est vérifiée que si le vecteur ω est quelconque, l'approximation de Galerkin consiste à considérer qu'il y a toujours cette équivalence en choisissant un petit nombre de vecteurs formant sur chaque élément une base polynomiale, dont les vecteurs sont nommés fonctions de base. Contrairement aux méthodes standards éléments finis qui choisissent ces fonctions de bases égales à leurs fonctions de forme, la méthode des éléments spectraux fait un choix différent, en utilisant le produit de polynômes de Lagrange de plus haut degré (typiquement entre 4 et 10) .

Comme cette méthode recourt à l'intégration numérique, il est nécessaire de définir une règle de quadrature, que l'on appliquera sur chacun des éléments du maillage. Pour des polynômes de degré n, la méthode de quadrature de Gauss utilise $n+1$ points d'intégration pour l'évaluation de leur intégrale sur un domaine régulier. Pour un élément déformé, cette évaluation n'est plus exacte, c'est pourquoi la règle de quadrature de Gauss-Lobatto-Legendre, qui fait intervenir les zéros de la dérivée du polynôme de Legendre d'ordre n, a été choisie pour limiter la perte de précision. A une dimension, elle fait intervenir les deux extrémités, -1 et 1 du segment d'intégration, ce qui permettra d'assurer la continuité d'un élément à un autre, grâce à ces points communs. Pour simplifier le calcul de l'intégrale, nous utilisons la liberté que l'on a dans le choix de ces fonctions de base, qui sont les polynômes de Lagrange $h_\alpha(\xi)$ tels que

$$h_\alpha(\xi) = \delta_{\alpha\beta} \tag{3.40}$$

où α et β désignent deux points d'intégration distincts. Leur allure est représentée sur la figure 3.7. Ainsi, seule une valeur est nécessaire pour évaluer l'intégrale, ce qui nous conduit à une matrice de masse diagonale. Cette propriété particulière sera un atout primordial de la méthode, et notamment sur deux points : le coût mémoire de la matrice passe de N_{glob}^2 à N_{glob}, et l'inversion de la matrice est directe. N_{glob} représente ici le nombre total de points distincts d'intégration de l'ensemble des éléments spectraux. Pour des méthodes classiques d'éléments finis, la taille maximale du problème à résoudre est souvent limitée par la dimension de cette matrice, d'abord difficile à stocker, mais aussi à inverser. Dans ces méthodes, l'alternative à l'inversion de la matrice est l'utilisation d'un schéma temporel implicite, inconditionnellement stable, mais qui oblige cependant à la construction entière de la matrice, dont le coût mémoire reste élevé.

Discrétisation temporelle

Le choix de la méthode d'intégration temporelle n'est pas unique. Le code Specfem en implémente deux : une méthode de Runge-Kutta d'ordre 6, et une méthode de Newmark. Dans la pratique, c'est souvent la méthode de Newmark qui offre le meilleur compromis entre vitesse et stabilité, dont l'analyse de stabilité a été menée dans [60]. Ses différentes étapes sont données dans l'algorithme 5.

Données : $U_k, \partial_t U_k, \partial_t^2 U_k, \Delta t, F^{ext}(t, i_{spec}), F^{int}(U, \partial_t U, i_{spec}), M$
% Les données $U_k, \partial_t U_k$ et $\partial_t^2 U_k$ sont définies à chaque point GLL.
%Le calcul des forces intérieures $F^{int}(U, \partial_t U, i_{spec})$ et extérieures $F^{ext}(t, i_{spec})$ dépendent de l'élément spectral i_{spec} considéré, et un point GLL peut appartenir à plusieurs éléments.
Résultat : $U_{k+1}, \partial_t U_{k+1}, \partial_t^2 U_{k+1}$
Étape 1 : Phase de prédiction

$$\begin{aligned}
\hat{U}_{k+1} &= U_k + \Delta t \partial_t U_k + \frac{\Delta t^2}{2} \partial_t^2 U_k \\
\partial_t \hat{U}_{k+1} &= \partial_t U_k + \frac{\Delta t}{2} \partial_t^2 U_k \\
\partial_t^2 \hat{U}_{k+1} &= 0
\end{aligned}$$

Étape 2 : Phase de résolution
$\Delta a = 0$
pour *chaque point GLL* i_{GLL} **faire**
  pour *chaque élément spectral* i_{spec} *contenant* i_{GLL} **faire**
    $\Delta a = \Delta a + \left(F^{ext}(t_{k+1}, i_{spec}) - F^{int}(\hat{U}_{k+1}, \partial_t \hat{U}_{k+1}, i_{spec})\right)/M$
  fin
fin
Étape 3 : Phase de correction

$$\begin{aligned}
\partial_t^2 U_{k+1} &= \Delta a \\
\partial_t U_{k+1} &= \partial_t \hat{U}_{k+1} + \frac{\Delta t}{2} \partial_t^2 U_{k+1} \\
U_{k+1} &= \hat{U}_{k+1}
\end{aligned}$$

Algorithme 5 : Méthode de Newmark explicite appliquée sur un maillage à éléments spectraux implémentée par le code Specfem.

Dans cet algorithme, le terme F^{ext} est égal au second membre de l'équation 3.37. Le calcul des forces intérieures F^{int} et de la matrice de masse M dépendent du problème considéré, et sera détaillé dans la partie suivante concernant l'équation d'onde acoustique. Comme évoqué, ce schéma explicite est conditionnellement stable, ce qui entraîne ici une diminution du pas de temps à utiliser dès que le maillage spatial est raffiné. Pour le problème de la FWI qui comme nous l'avons vu conduit à un calcul de noyaux impliquant l'intégration des champs acoustiques, cette

contrainte n'en est cependant pas vraiment une car la précision de l'intégration numérique de ces champs implique aussi un pas de temps assez faible et variant proportionnellement à la finesse du maillage.

Application à l'équation d'onde

Pour ne pas multiplier les formules, les équations qui suivent décrivent la discrétisation du problème uniquement dans le cas 3D, c'est-à-dire $\Omega \subset \mathbb{R}^3$ et $\mathbf{x} \in \mathbb{R}^3$.

La méthode des éléments spectraux va maintenant être appliquée à l'équation d'onde acoustique 3.3 dont la variable principale, le potentiel $\Phi(\mathbf{x}, t)$, est liée au déplacement par l'équation 3.2. La formulation faible associée peut s'exprimer de cette façon, après multiplication par le vecteur test w et une intégration par partie :

$$\int_\Omega (\kappa^{-1}\partial_t^2\,\Phi) w \mathrm{d}\mathbf{x} = -\int_\Omega \rho^{-1}\nabla\Phi\nabla w \mathrm{d}\mathbf{x} + \int_\Omega (\sum_{s=1}^{N_s} \kappa(\mathbf{x}_s)^{-1} f_s(t)) w \mathrm{d}\mathbf{x} \tag{3.41}$$

Le calcul de l'intégrale se fait sur chacun des éléments spectraux. On note Ω_e l'un de ces éléments. L'intégration d'une fonction f définie en coordonnées globales sur cet élément s'exprime avec la règle de quadrature de Gauss-Lobatto-Legendre :

$$\begin{aligned}\int_{\Omega_e} f(\mathbf{x})\mathrm{d}\mathbf{x} &= \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(\mathbf{x}(\boldsymbol{\xi}))J(\boldsymbol{\xi})\mathrm{d}\boldsymbol{\xi} \\ &= \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=0}^{n} \omega_i\omega_j\omega_k f^{ijk} J^{ijk}\end{aligned} \tag{3.42}$$

avec ω_i,ω_j et ω_k les poids de la règle de quadrature dans chaque dimension, J^{ijk} l'évaluation du jacobien au point de quadrature (ξ_i, η_j, ζ_k) (par la suite appelé points GLL), f^{ijk} l'évaluation de la fonction en ce même point et n le degré du polynôme de Lagrange. On remarquera que l'on a choisi ici le même degré pour chacune des trois dimensions, afin de ne privilégier aucune direction en terme de précision de la méthode.

Il est alors possible de déterminer l'expression discrète de chacun des termes de l'équation 3.41, en les combinant avec l'équation 3.42.

Obtention de la matrice de masse

La matrice de masse caractérise dans la formulation faible et matricielle du problème le comportement de la dérivée seconde temporelle de la variable principale, par analogie avec l'équation de l'oscillateur harmonique. Elle correspond dans l'équation 3.41 au membre de gauche : $\int_\Omega (\kappa^{-1}\partial_t^2\,(.)) w \mathrm{d}\mathbf{x}$.

En notant $\ddot{\Phi}^{ijk}$ la valeur de $\partial_t^2\,\Phi(\mathbf{x}(\xi_i, \eta_j, \zeta_k), t)$, on peut écrire :

$$\begin{aligned}\int_{\Omega_e} (\kappa^{-1}\partial_t^2\,\Phi) w \mathrm{d}\mathbf{x} &= \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \kappa^{-1}(\mathbf{x}(\boldsymbol{\xi}))w(\mathbf{x}(\boldsymbol{\xi}))\partial_t^2\,\Phi(\mathbf{x}(\boldsymbol{\xi}), t)J(\boldsymbol{\xi})\mathrm{d}\boldsymbol{\xi} \\ &= \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=0}^{n} \omega_i\omega_j\omega_k (\kappa^{-1})^{ijk} J^{ijk} w^{ijk} \ddot{\Phi}^{ijk}\end{aligned} \tag{3.43}$$

Comme on peut le voir, en chaque point GLL (ξ_i, η_j, ζ_k) de l'élément Ω_e, la prise en compte de la valeur $\ddot{\Phi}^{ijk}$ pour l'estimation de l'intégrale est simplement obtenue par la multiplication par le terme $\omega_i\omega_j\omega_k(\kappa^{-1})^{ijk}J^{ijk}w^{ijk}$, ce qui nous conduit à une matrice de masse diagonale. Comme précédemment évoqué, cette propriété, héritée de la règle de quadrature et du choix des fonctions de base, combinée à un schéma temporel explicite, nous permettra d'écrire naturellement la parallélisation de l'algorithme.

Obtention de la matrice de rigidité

Tout comme la matrice de masse, la matrice de rigidité fait référence au problème de l'oscillateur harmonique, plus précisément à la constante k de rigidité située devant la dérivée seconde spatiale de la variable principale. Il s'agit ici du premier membre de droite de l'équation 3.41. Sa discrétisation est moins immédiate que la précédente.

On réécrit le premier terme du second membre, en exprimant le produit des gradients qui est un produit scalaire entre les trois composantes de ces deux gradients, avec la convention $x_1 = x$, $x_2 = y$ et $x_3 = z$:

$$-\int_\Omega \rho^{-1}\nabla\Phi\nabla w \mathrm{d}\mathbf{x} = -\int_\Omega \rho^{-1}(\sum_{\alpha=1}^{3}\partial_{x_\alpha}\Phi\partial_{x_\alpha}w)\mathrm{d}\mathbf{x} \tag{3.44}$$

On exprime ensuite l'expression discrète de la dérivée spatiale de Φ dans la direction α en coordonnées locales :

$$\begin{aligned}\partial_{x_\alpha}\Phi(\mathbf{x}(\xi_i,\eta_j,\zeta_k),t) =& \Big[\sum_{\sigma=0}^{n}\Phi^{\sigma jk}(t)h'_\sigma(\xi_i)\Big]\partial_{x_\alpha}\xi(\xi_i,\eta_j,\zeta_k) + \Big[\sum_{\sigma=0}^{n}\Phi^{i\sigma k}(t)h'_\sigma(\eta_j)\Big]\partial_{x_\alpha}\eta(\xi_i,\eta_j,\zeta_k)\\ &+ \Big[\sum_{\sigma=0}^{n}\Phi^{ij\sigma}(t)h'_\sigma(\zeta_k)\Big]\partial_{x_\alpha}\zeta(\xi_i,\eta_j,\zeta_k)\end{aligned} \tag{3.45}$$

Comme on peut le voir, cette expression fait intervenir l'inverse de la matrice jacobienne du changement de variable, dont on avait préalablement calculé le déterminant en chaque point pour s'assurer du caractère bien posé de ce changement de variables. Elle fait aussi intervenir les dérivées des fonctions de base aux points d'intégration, que l'on aura pris soin de calculer analytiquement lors de l'initialisation du programme.

D'une façon similaire, on peut exprimer les dérivées spatiales du vecteur w, qui ne dépendent pas du temps et sont dans la pratique calculées à l'initialisation du programme. En combinant leurs expressions et avec la règle d'intégration 3.42, on obtient :

$$\begin{aligned}-\int_{\Omega_e}\rho^{-1}\nabla\Phi\nabla w\mathrm{d}\mathbf{x} =& \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=0}^{n} w^{ijk}\Big[\omega_j\omega_k\sum_{i'=0}^{n}\omega_{i'}\rho^{-1^{i'jk}}J^{i'jk}\partial_x\Phi^{i'jk}(t)h'_{i'}(\xi_i)\\ &+ \omega_i\omega_k\sum_{j'=0}^{n}\omega_{j'}\rho^{-1^{ij'k}}J^{ij'k}\partial_y\Phi^{ij'k}(t)h'_{j'}(\eta_j)\\ &+ \omega_i\omega_j\sum_{k'=0}^{n}\omega_{k'}\rho^{-1^{ijk'}}J^{ijk'}\partial_z\Phi^{ijk'}(t)h'_{k'}(\zeta_k)\Big]\end{aligned} \tag{3.46}$$

Cette expression, significativement plus complexe que celle de la matrice de masse, fait intervenir trois doubles sommations en chaque point GLL (ξ_i,η_j,ζ_k) pour obtenir ce terme en fonction de $\Phi^{ijk}(t)$. Ainsi, il n'est pas possible d'exprimer avec une matrice simplement diagonale le produit à obtenir. Même si cette expression, exprimée dans le second membre de la formulation faible, ne nécessite pas d'inversion de matrice, la simple construction d'une telle matrice aurait un coût prohibitif, de dimension $N_{glob}\times N_{glob}$. A titre d'exemple, les simulations à 2D concrètement réalisées sur un ordinateur personnel utilisent actuellement quelques millions de points d'intégration. Si une telle matrice était construite, le coût mémoire associé serait de plusieurs dizaines de téraoctets. Même avec un mode de stockage de type parcimonieux, le grand nombre de points d'intégration à l'intérieur d'un seul élément spectral entrainerait un coût mémoire de l'ordre de $N_{GLL}\times N_{GLL}\times N_{GLL}\times N_{spec}$, qui est trop élevé. Pour éviter cet assemblage, un tableau permettant de récupérer la liste de tous les points GLL associés à un élément spectral donné est créé et utilisé. Son coût mémoire ne dépasse pas la centaine de mégaoctets pour une simulation équivalente, ce qui est beaucoup plus raisonnable et peut être intégralement chargé en mémoire RAM ou en mémoire principale d'un GPU par exemple. Enfin, cette façon de résoudre le problème est facilement parallélisable, puisqu'il suffira de diviser ce tableau selon les portions du maillage.

Cette portion du code, appelée opération d'assemblage en éléments finis, qui fait intervenir un très grand nombre d'opérations mathématiques successives, doit être appelée à chaque itération temporelle. De ce fait, elle est la partie la plus coûteuse en terme de temps de calcul pour des larges simulations, et a été l'objet d'une attention particulière lors de l'optimisation du code.

Prise en compte du terme source

Enfin, il nous faut discrétiser le second terme du membre de droite de la formulation faible 3.41, qui modélise le terme source de l'équation d'onde. L'obtention de son expression est assez directe, en utilisant à nouveau la règle de quadrature :

$$\begin{aligned}\int_\Omega \Big(\sum_{s=1}^{N_s} \kappa(\mathbf{x}_s)^{-1} f_s(t)\Big) w \mathrm{d}\mathbf{x} &= \sum_{s=1}^{N_s} \int_{\Omega_{es}} \kappa(\mathbf{x}_s)^{-1} f_s(t) w \mathrm{d}\mathbf{x} \\ &= \sum_{s=1}^{N_s} \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=0}^{n} \omega_i \omega_j \omega_k \kappa^{-1^{ijk}} f_s^{ijk} J^{ijk} w^{ijk}\end{aligned} \tag{3.47}$$

Considérations numériques et parallélisation

Nous venons de voir que le code Specfem implémente la méthode des éléments spectraux appliquée à l'équation d'onde acoustique, avec une règle de quadrature et des fonctions de base particulières permettant de limiter le coût mémoire et le coût numérique associé habituellement à ce type de méthode. En plus de la méthode choisie, il est important, afin de pouvoir obtenir concrètement et de la façon la plus rapide possible l'expression des kernels nécessaires au processus d'optimisation locale, d'identifier quels sont les facteurs limitant la performance. Nous allons à présent les caractériser, et voir quels sont les principaux obstacles à la parallélisation. Si historiquement le code a été parallélisé dans le but d'une exécution sur CPU, nous détaillerons ici les caractéristiques de la version dédiée à une exécution sur GPU [21, 61].

Calcul pratique des noyaux

L'expression des noyaux 3.27 exprimant la convolution de deux champs peut laisser penser que seules deux simulations de propagation acoustique sont nécessaires pour son calcul. Cependant, cela impliquerait de stocker la totalité du champ en mémoire globale, ou à l'aide de disques durs. Ici aussi, le coût mémoire associé serait prohibitif, avec une taille de $4 \times N_{glob} \times N_{step}$, N_{step} étant le nombre d'itérations temporelles. Il serait possible de ne stocker qu'un nombre limité de frames, mais cela conduirait à une perte d'information lors de l'intégration temporelle. Enfin, en utilisant cette approche avec un cluster, la lecture à chaque itération d'une grande quantité de données conduirait à de gros problèmes d'accès disque, et au mieux à un ralentissement conséquent de l'exécution du programme. Pour pallier ce problème, le code Specfem fait le choix de recalculer intégralement le champ direct lors du calcul du champ adjoint. Comme ces deux champs doivent être corrélés à des instants différents (t et $T-t$), le champ direct est en fait reconstruit "à l'envers" : lors de la simulation du champ direct, la dernière frame temporelle est sauvegardée, et est utilisée comme premier instant du champ direct pour la reconstruction. Lors des itérations temporelles, les termes sources $f_s(t)$ sont aussi lus à l'envers. A l'itération i, les deux champs direct et adjoints ne sont connus qu'aux instants i et $N_{step} - i$, ce qui permet de faire l'intégration temporelle inhérente au calcul des noyaux au fil de la propagation, pour un coût mémoire qui n'est que le double de celui de la propagation directe, sans perte d'information, et pour un coût numérique total seulement 30% supérieur à l'approche "intuitive".

Parallélisation sur GPU

Pour augmenter la vitesse d'exécution du code séquentiel, le code Specfem a été implémenté sur GPU [21, 61] à 3D, puis a été implémenté à 2D dans le cadre de cette thèse. La principale

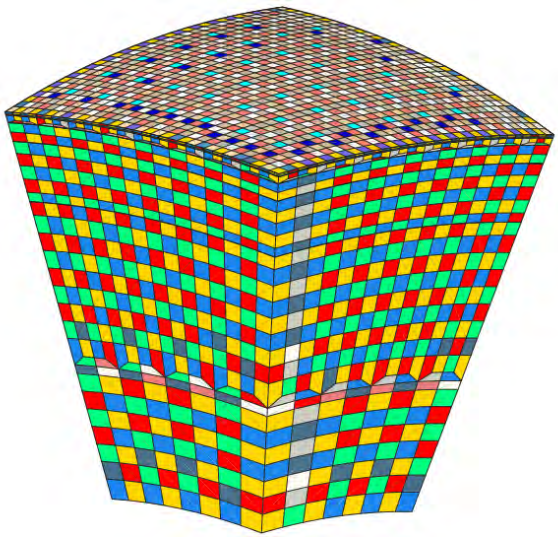


FIGURE 3.8 – Découpage du maillage en plusieurs sous-parties pour éviter le recours aux fonctions atomiques. Chaque zone, représentée par une couleur différente, ne possède que des éléments non-adjacents. Image issue de [61].

motivation de cette implémentation à 2D était alors de s'adapter au type de données qu'il est possible d'acquérir au laboratoire PHASE, qui dispose d'une barrette linéaire à 128 éléments. En effet, étant donné les fréquences impliquées et les dimensions spatiales des problèmes à traiter, le problème n'était pas modélisable à 3D actuellement pour des raisons de coût mémoire, et la version à 2D sur CPU était alors trop lente pour pouvoir mener à bien une inversion sans l'aide d'un cluster.

De nombreuses portions du code, et en particulier à l'intérieur de la boucle temporelle ont été portées sur GPU. De façon générale, le maillage avec des éléments spectraux a été adapté à la structure d'un kernel GPU en définissant un bloc responsable d'un élément spectral, et un thread responsable d'un point d'intégration GLL. Ensuite, en fonction de la situation, la taille de la grille a été adaptée. Par exemple, le calcul des forces acoustiques nécessite une grille de taille égale à celle du nombre total d'éléments spectraux du maillage, puisqu'ils sont tous concernés par cette étape, alors que la grille est définie comme étant de taille égale au nombre de sources pour la prise en compte des termes sources. La division par la matrice de masse s'effectue quant à elle en appelant un nombre de thread total égal à N_{glob}, puisque celle-ci ne dépend plus des éléments spectraux lors de cette opération.

Comme évoqué, le calcul des forces intérieures, qui implémente les équations 3.45 et 3.46, nécessite un grand nombre d'opérations mathématiques. Comme chaque point GLL d'un élément spectral donné a besoin d'utiliser les coefficients (poids GLL, dérivée seconde du potentiel...) de chacun des autre points GLL appartenant au même élément, une utilisation pertinente de la mémoire partagée a permis de limiter les accès à la mémoire globale et d'obtenir une bonne intensité arithmétique. Par rapport au cas à 3D, ce calcul est d'une intensité arithmétique moindre car chaque donnée est réutilisée par chaque point GLL, soit 25 fois à 2D contre 125 fois à 3D. Pour réhausser le niveau d'utilisation de ces données, lors du calcul des noyaux la reconstruction du champ direct a été fusionnée à la construction du champ adjoint dans le kernel qui calcule les forces intérieures, car ces deux champs utilisent les mêmes constantes. Sur une configuration de 160×160 éléments, le calcul des forces intérieures est effectué en 0.30ms lors du calcul d'un seul champ, alors que le calcul des forces intérieures pour les deux champs prend 0.38ms, ce qui est bien moins que le double du temps de calcul pour un champ. Ainsi, la reconstruction du champ direct lors du calcul des noyaux ne coûte que 12% de plus en terme de temps de calcul, et non pas 30% comme sur la version CPU ou sur la version GPU à 3D.

Par ailleurs, l'ajout de la contribution de la force calculée sur un élément spectral donné se fait

de façon particulière, car certains points GLL sont communs à plusieurs éléments. Pour ces points, il est possible que l'ajout de la contribution de deux éléments différents se fasse en simultané, et que le cycle de lecture-écriture en mémoire globale soit corrompu. Pour éviter ce problème, le recours aux fonctions atomiques est la solution la plus appropriée. Dans [61], ce problème est contourné en coloriant le maillage en plusieurs sous-groupes ne possédant pas d'éléments adjacents, comme illustré sur la figure 3.8. Alors, au lieu de déclarer une grille de taille égale au nombre d'éléments spectraux dans le maillage, plusieurs appels aux kernels sont effectués en lançant séquentiellement chacun des groupes. Cette idée, particulièrement efficace il y a quelques années, a moins d'impact positif sur la performance à présent car les fonctions atomiques proposées par CUDA ont été grandement optimisées depuis.

Grâce à cet effort de parallélisation, un facteur d'accélération d'environ 10 a été obtenu par rapport à la version parallélisée sur CPU avec 8 cœurs, en utilisant un double processeur Intel Xeon E5-2609 v2 cadencé à 2.5 GHz et un GPU Nvidia Titan Black avec 2880 cœurs cadencés à 889 MHz. Comme on peut le voir sur la figure 3.9, cet écart est obtenu pour les simulations acoustiques et élastiques, et ne semble pas dépendre de la taille du problème dès lors que celle-ci est assez grande (plusieurs centaines de milliers de points). Cette amélioration a permis notamment de faire passer le coût d'une inversion à 2D, qui utilise plusieurs centaines de fois la simulation de la propagation d'onde, d'une douzaine d'heures à environ une demi-heure, ce qui a été grandement appréciable pour la suite de ces travaux de recherche.

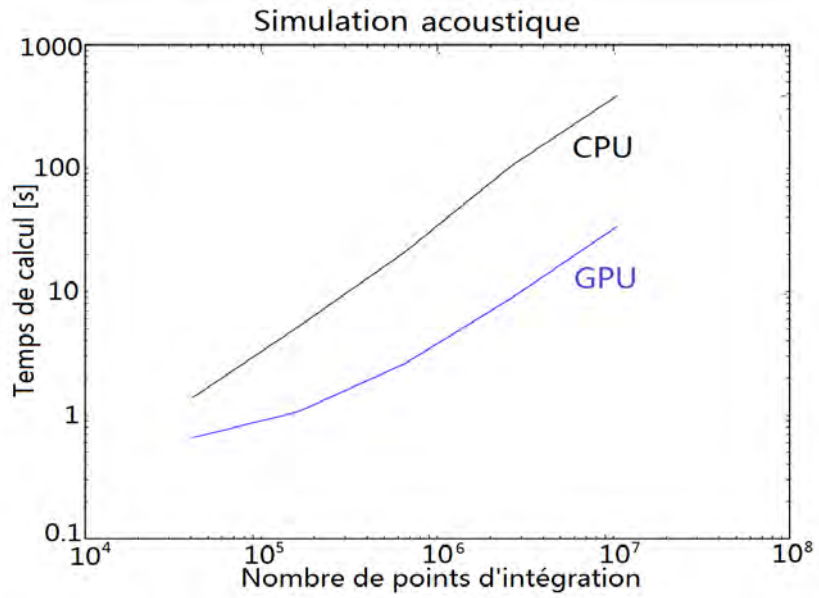
Problèmes de grande taille

A partir d'un certain niveau de finesse de discrétisation, le coût mémoire peut dépasser la quantité de mémoire disponible en RAM, ou sur celle d'un GPU. De plus, la grande quantité d'opérations mathématiques associée justifie aussi la nécessité de séparer le problème en plusieurs morceaux, afin qu'il soit traité sur différents GPUs. Historiquement, cette parallélisation a été effectuée dans le but d'une exécution sur un grand nombre de processeurs. Les difficultés associées sont similaires à celles que l'on rencontre pour une exécution multi-GPUs.

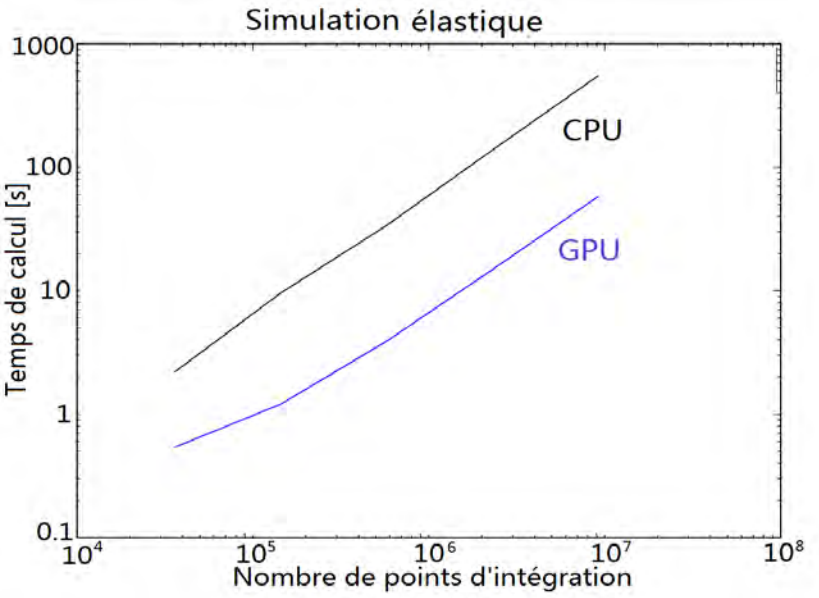
L'utilisation du schéma temporel explicite avec une matrice de masse diagonale permet de décomposer assez facilement le problème : chaque portion (ou partition) du maillage est affectée à un GPU, et chaque GPU reçoit la portion de la matrice de masse qui correspond à ses éléments.

Sur la plupart des structures informatiques parallélisées, les ressources matérielles CPU ou GPU effectuant le calcul sont les mêmes, en plusieurs dizaines, centaines ou milliers d'exemplaires. Il est alors important de s'assurer que les portions du maillage effectuent un nombre semblable d'opérations mathématiques, afin que toutes les ressources soient occupées jusqu'à ce qu'elles doivent communiquer les valeurs des points aux frontières. Dans le cas d'un maillage non structuré, avec des éléments potentiellement acoustiques, (visco)élastiques, ou dans des couches absorbantes, le découpage du maillage constitue en soi un problème d'optimisation. Le code Specfem utilise un partitionneur de maillage externe, nommé Scotch [62], afin de le résoudre.

En calcul parallèle multi-CPU ou multi-GPU, un des principaux freins à la performance est le coût des communications entre unités de calcul qui est directement lié à la bande passante entre ces éléments. Si l'absence de la construction d'une matrice de taille $N_{glob} \times N_{glob}$ évite de devoir communiquer un nombre considérable de valeurs à chaque itération temporelle, le coût de ces communications s'avère déjà impacter le temps global d'exécution. A chaque itération en temps, les valeurs des contributions des points GLL situés sur les bords d'une partition doivent être communiquées aux autres unités de calcul, afin qu'elles soient prises en compte dans le calcul des forces. Pour limiter ce délai pendant lequel les unités de calcul sont inutilisées, une stratégie de communication a été mise en place, détaillée dans [51] : les éléments spectraux internes et ceux situés sur les bords de chaque partition sont séparés en deux groupes. Les éléments externes sont d'abord calculés, puis une communication non bloquante est utilisée pour transmettre entre partitions les valeurs des points GLL situés sur les frontières. Pendant que les valeurs sont transmises, le calcul des forces intérieures est lancé sur les éléments intérieurs des partitions, lesquels consti-



(a) Simulation acoustique



(b) Simulation élastique

FIGURE 3.9 – Représentation du temps de calcul sur GPU et sur CPU parallélisé sur 8 cœurs en fonction du nombre de points GLL utilisé par une simulation à 2D sur 5000 pas de temps.

tuent la plupart des éléments du maillage. En procédant de la sorte, le coût des communications est quasiment masqué. Pour le cas à 2D sur GPU, ces communications demeurent légèrement plus longues que le calcul des forces intérieures sur les éléments intérieurs, car le calcul sur une partition s'effectue très rapidement, et que le coût d'une communication d'un GPU à un autre est plus importante qu'une communication CPU.

Chapitre 4

Application de la FWI à l'échelle ultrasonore : reconstitution de la carte de vitesse d'un milieu inconnu

Le but de ce chapitre est de présenter l'ensemble des résultats obtenus quant à l'utilisation de la FWI à l'échelle ultrasonore, qui concernent en premier lieu la reconstruction de la carte de vitesse de compression dans un milieu inconnu. Pour des raisons matérielles que nous justifierons, seules des données simulées numériquement ont été utilisées lors de ces inversions. Tout d'abord, nous allons expliciter les principales différences que l'on peut trouver entre la FWI appliquée à la géophysique et les configurations expérimentales à l'échelle ultrasonore. Ensuite, deux classes de problèmes seront distinguées : les problèmes dont la solution réelle ne possède qu'un faible écart avec le modèle initial, et ceux présentant de fortes variations imprévues de vitesse. Nous verrons en particulier l'influence de l'ensemble des paramètres décrits dans le chapitre précédent sur la solution obtenue et sur la vitesse de convergence. Enfin, les différences potentielles entre données simulées et données réelles seront explicitées, dans la perspective de l'application concrète de l'inversion à des données réelles.

Les résultats présentés ici sont le fruit d'une collaboration avec le professeur Dimitri Komatitsch du LMA à Marseille et avec le groupe de recherche 'Sismologie théorique et informatique' du département de Géophysique de l'Université de Princeton aux États-Unis, dirigé par le professeur Jeroen Tromp. Dans le cadre de cette thèse, un séjour de six mois a été effectué dans le but de profiter de l'expérience de ce groupe en matière d'inversion de données sismiques. L'ensemble des résultats présentés ici ont été obtenus en utilisant le workflow d'inversion SeisFlows, dont le code est open source et est développé par ce groupe de recherche, en particulier par Ryan Modrak.

4.1 Spécificités de l'échelle ultrasonore

Tout d'abord, notons que la transposition d'une technique de traitement des données de géophysique à des données ultrasonores peut surprendre : entre 5 et 8 ordres de grandeurs séparent ces deux disciplines. Toutefois, le lien très fort qui les relie, à savoir la physique sous-jacente et sa description mathématique à travers l'équation d'onde, sont à l'origine de cette idée de la transposition de la technique.

Dispositifs d'acquisition en géophysique

En géophysique, plus particulièrement en sismologie, nous pouvons distinguer deux sous-disciplines :

— La géophysique globale : elle consiste en l'étude de la Terre à l'échelle la plus large, et où le globe est entièrement modélisé dans la simulation numérique. Pour de telles situations,

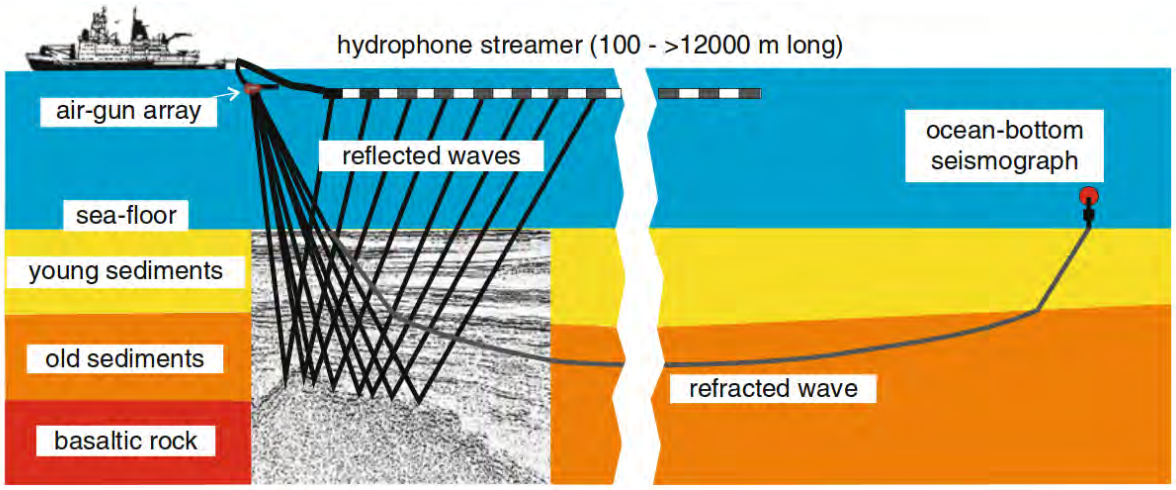


FIGURE 4.1 – Schéma d'acquisition de données offshore en géophysique d'exploration. Dans la perspective d'une inversion, l'enregistrement d'ondes en transmission est recherché. Image issue de [63].

seuls les tremblements de Terre peuvent fournir assez d'énergie pour que les ondes émises soient observables en tout point de la Terre. Ainsi, une partie du travail consiste aussi à retrouver les propriétés de la source, souvent modélisée par un tenseur des moments sismiques. Les séismes sont enregistrés à travers un réseau de stations couvrant inégalement la surface terrestre en raison de la présence des océans, par opposition aux données ultrasonores, récoltées sur des barrettes échographiques constituées de transducteurs disposés à égale distance. La bande passante des signaux enregistrés s'étale sur plusieurs décades, du millième jusqu'à quelques Hertz. Les ondes observées ont la particularité d'avoir traversé la Terre et ne sont souvent pas des ondes réfléchies.

— La géophysique d'exploration : le but est de cartographier une zone régionale, allant de un à quelques centaines de kilomètres, et possède notamment une application en prospection pétrolière. A cette fin, des ondes sont générées activement, par des explosions, des camions vibrants ou des canons à air, et sont enregistrées à travers des réseaux assez bien équirépartis de sismographes, ou de lignes de géophones en mer. La relative maîtrise de la source ainsi que la régularité des points d'enregistrement sont autant de similarités avec les dispositifs ultrasonores. Par ailleurs, la bande passante du signal source est moins large qu'en géophysique globale, elle est de l'ordre de deux décades. Cependant, plusieurs signaux sources au contenu fréquentiel très différent peuvent être utilisés, offrant ainsi un spectre à la largeur contrôlable, propice à l'inversion.

Comme nous l'avons vu dans le premier chapitre, les dispositifs expérimentaux ultrasonores, pour la plupart basés sur l'idée d'imagerie échographique, utilisent le même appareil pour l'émission et la réception des ondes. En géophysique d'exploration les ondes sont aussi émises et écoutées du même côté du milieu inconnu, à savoir de la surface terrestre, même si les recepteurs y sont différents des émetteurs et qu'un certain décalage spatial entre les deux objets existe.

Intérêt d'un dispositif d'acquisition en transmission

Si on se réfère à la littérature de la géophysique d'exploration, une idée cruciale quant au succès de l'inversion réside dans la largeur de la zone sur laquelle effectuer les mesures. En particulier, il est important au début de l'inversion de chercher à obtenir des angles d'ouverture source-receveur les plus larges possibles comme sur la figure 4.1, et avec des basses fréquences [63, 65]. En procédant ainsi, seules les ondes réfractées sont obtenues et utilisées pour les premières itérations du processus d'inversion. Dans [64], ce point clé de l'inversion est formalisé par une formule exprimée en milieu

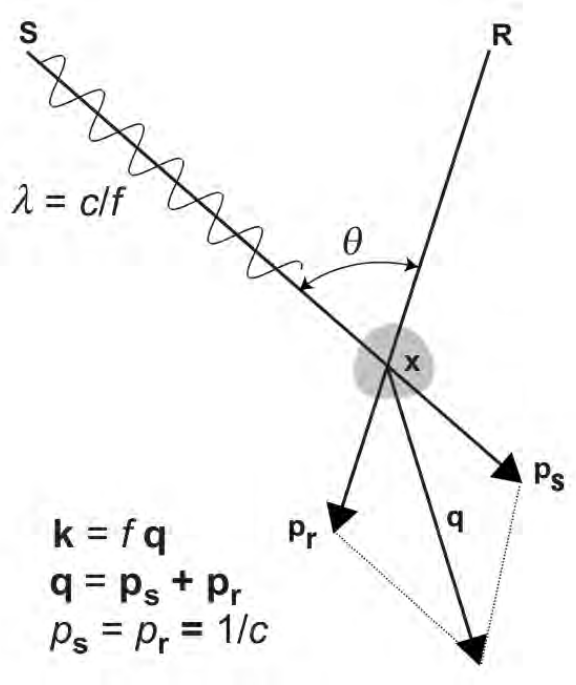


FIGURE 4.2 – Schéma des liens entres les paramètres importants pour l'inversion. Image issue de [64].

acoustique homogène, illustrée sur la figure 4.2 :

$$|k| = |\frac{2f}{c_0}\cos(\frac{\theta}{2})| \tag{4.1}$$

Cette relation exprime le lien entre le nombre d'onde au point de réflexion et la fréquence, la vitesse locale et l'angle de l'onde incidente/réfléchie ou réfractée. Au début de l'inversion, il est important de minimiser cette quantité pour extraire l'information basse fréquence contenue dans les données, ce qui comme nous l'avons vu est d'une importance capitale pour le succès de l'inversion. L'équation 4.1 exprime, comme on pouvait s'y attendre, que ce contenu est révélé par la partie basse du spectre observé, mais surtout par les ondes dites en transmission, à savoir celles minimisant la quantité $\cos(\frac{\theta}{2})$, et qui se produisent donc vers $\theta \simeq 180^\circ$.

Ainsi, on peut en déduire le fait que les dispositifs expérimentaux de la géophysique d'exploration tout comme ceux de l'échographie conventionnelle ne sont pas les plus propices à l'inversion : dans le cas extrême du mono-élément, la quantité $\cos(\frac{\theta}{2})$ est maximisée, ce qui rend les données acquises très difficiles à inverser. On peut également le comprendre en observant que les données ne peuvent être que la signature d'ondes en réflexion.

Dans le cas de la géophysique d'exploration, l'impossibilité de placer des points d'enregistrement à plusieurs dizaines de kilomètres explique la stratégie du travail avec des angles larges d'ouverture source-receveur. Dans le cas de l'acoustique ultrasonore, nous pourrons remarquer que dans de nombreuses configurations, rien n'empêche de placer une seconde barrette échographique derrière le milieu inconnu, et permettant de récupérer une information bien plus utile au bon déroulement du processus d'inversion que celle apportée par les ondes en réflexion. Par ailleurs, l'approche multi-fréquentielle de la géophysique d'exploration n'est possible que parce que les sismomètres, enregistrant un déplacement, peuvent être conçus pour être large bande. En acoustique ultrasonore, la plupart des récepteurs sont des piézoélectriques, lesquels ont une bande passante bien plus faible, de l'ordre de l'octave.

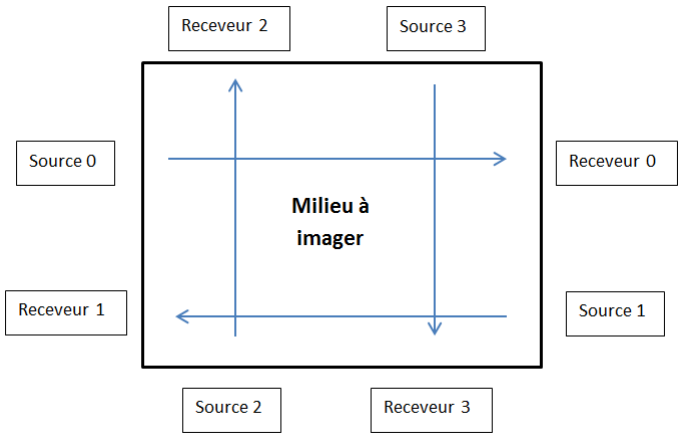


FIGURE 4.3 – Schéma du dispositif utilisé lors des expériences numériques pour une inversion à 2D. L'inspection du milieu se décompose en l'émission de quatre ondes, émises de chaque côté du milieu, et enregistrées du côté opposé à l'émission.

Choix du dispositif d'acquisition et des expériences numériques

A partir de ces différentes considérations, nous avons fait le choix de travailler en modifiant la configuration conventionnelle de l'imagerie ultrasonore, qui privilégie l'utilisation d'une seule et même barrette de capteurs pour l'émission et la réception. Comme illustré sur la figure 4.3, nous avons choisi d'utiliser une seconde barrette échographique en mode réception, située de l'autre côté du milieu à imager. Grâce à la récupération des ondes en transmission, la faible largeur de bande des transducteurs piézoélectriques sera compensée.

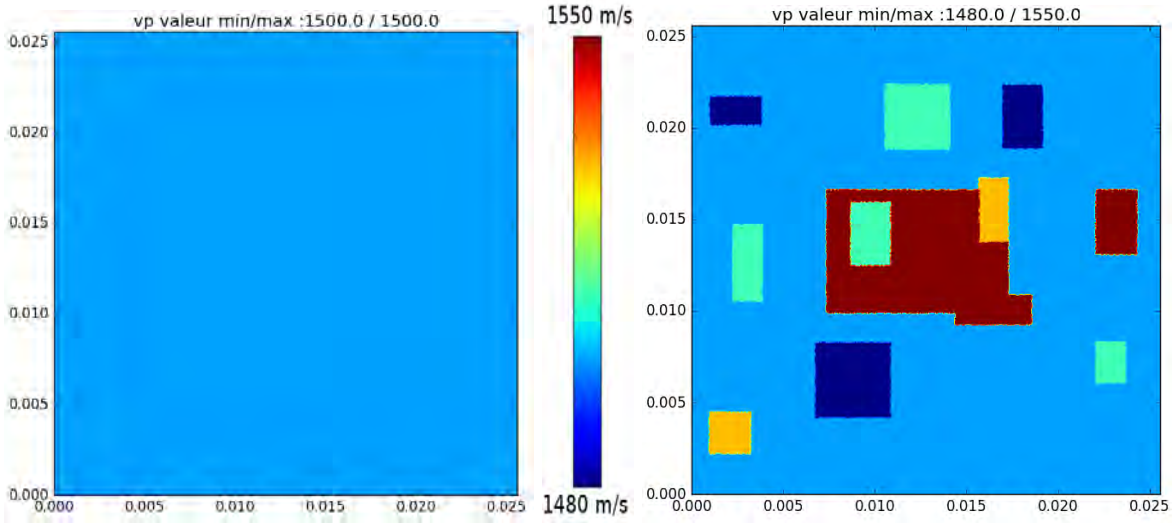
La fréquence dominante de la barrette de transducteurs du laboratoire est de 5 MHz. A 2D, la simulation numérique d'une onde au spectre compris entre 2.5 MHz et 7.5 MHz sur un domaine de 2.5×7.5 cm^2 requiert un maillage de 300×900 éléments spectraux, et environ 50000 pas de temps pour respecter la condition de stabilité du schéma numérique. Avec l'utilisation d'un GPU, la durée d'une telle simulation se situe autour de 5 minutes. Si ce coût peut paraître faible, le recours massif à cette configuration de propagation au cours de l'inversion (de l'ordre de plusieurs centaines) la rend difficilement compatible avec des expérimentations numériques intensives. A la place, nous avons choisi une fréquence centrale de 1 MHz, sur un domaine de 2.56 cm $\times$ 2.56 cm. Le maillage correspondant n'est constitué que de 80×80 éléments spectraux. Pour que l'onde traverse de part et d'autre le maillage, 1100 pas de temps sont suffisants (avec $\Delta t = 2 \times 10^{-8}$s). Sur GPU, le coût d'une telle simulation numérique est de l'ordre de quelques secondes.

4.2 Faibles contrastes de vitesse

Dans cette partie, les résultats d'inversion de cartes de vitesse présentant de faibles variations sont présentés. Chacune de ces inversions se base sur un modèle initial homogène 4.4(a), avec $c = 1500$m/s, ce qui constitue un cas très général : aucune hypothèse n'est nécessaire quant à la position a priori de certaines hétérogénéités. Le signal source employé est une ondelette de Ricker, dont le spectre est compatible avec la bande passante des transducteurs piézoélectriques. Pour limiter la difficulté de l'inversion, la durée d'enregistrement est telle que seule l'onde de transmission est enregistrée. L'algorithme de descente L-BFGS est utilisé, et les conditions de Wolfe sont appliquées pour la recherche du pas de descente.

Premières inversions d'un modèle

La carte de vitesse objectif est illustrée sur la figure 4.4(c). Elle présente l'intérêt de contenir plusieurs zones avec des valeurs différentes de vitesse et avec des variations abruptes, qui seront



(a) Modèle initial (b) Vitesse (c) Modèle objectif

FIGURE 4.4 – Cartes de vitesse du modèle de départ et du modèle recherché.

dans la réalité des interfaces. Les écarts de valeurs ont été choisis tels que la condition de cycle skipping Eq. 3.3 soit respectée à 1MHz.

Sur les résultats présentés sur la figure 4.5, nous faisons varier la vitesse de décroissance du rayon du lissage Gaussien qui est appliqué au gradient à chaque itération. Sur la première ligne, aucun lissage n'est appliqué. Sur la seconde et la troisième, le rayon de lissage est diminué de 50% et respectivement de 10% entre chaque itération. La valeur initiale du rayon est fixée à 0.00375 m, ce qui correspond à environ un septième de la taille du domaine. Comme on peut le voir, l'influence de cette décroissance sur la fonction coût et sur la convergence de la méthode est importante.

Grâce au lissage appliqué au gradient, le modèle oscille moins, par opposition aux premières itérations de la tentative d'inversion sans lissage, où l'on peut voir que le modèle final 4.5(c) reste marqué par les fortes oscillations déjà présentes dans les premières itérations 4.5(a) 4.5(b), et la solution obtenue a été piégée par un minimum local. Au fil des itérations, les variations moins étendues spatialement apparaissent, de même que les contours précis des interfaces. Dans ces simulations, la fonction coût associée diminue entre 3 et 4 ordres de grandeur. On peut voir que celle-ci diminue fortement au cours des premières itérations, et qu'ainsi une centaine d'itérations suffisent pour arriver à convergence. De fait, il n'est pas forcément évident de trouver un critère d'arrêt : un critère du type $\frac{f(x_{k+1})-f(x_k)}{f(x_k)}$ mesurant la vitesse de convergence sur deux itérations successives aurait ici arrêté le processus itératif avant d'atteindre l'asymptote horizontale visible sur les fonctions coût. Sur ces trois inversions présentées, nous pourrions être tentés de conclure qu'une décroissance faible du rayon de lissage conduit à un meilleur résultat. Cependant, lors d'une autre expérience pendant laquelle la décroissance du rayon de lissage était seulement de 1% entre deux itérations, le processus s'est arrêté au bout de 12 itérations. En ne changeant que très peu le niveau de lissage, la résolution spatiale de l'information révélée par le gradient n'évolue que trop lentement, et l'algorithme converge alors trop rapidement vers la solution 'floutée' par rapport au niveau de décroissance imposé au rayon. Dans le cas de l'inversion d'un modèle possédant un faible écart avec le modèle initial, il est important de ne pas faire décroître trop lentement ce paramètre pour éviter ce problème. Ainsi, un compromis en fonction de la connaissance a priori du problème doit être trouvé pour déterminer cette vitesse de décroissance, qui influence directement la solution obtenue.

Sur les résultats de la figure 4.5, on peut remarquer que les valeurs extrêmes sont obtenues près des points d'enregistrement de la pression, qui sont situés près des bords. Ceci est dû à de fortes valeurs du gradient, qui sont engendrées par de fortes valeurs du champ adjoint lors de

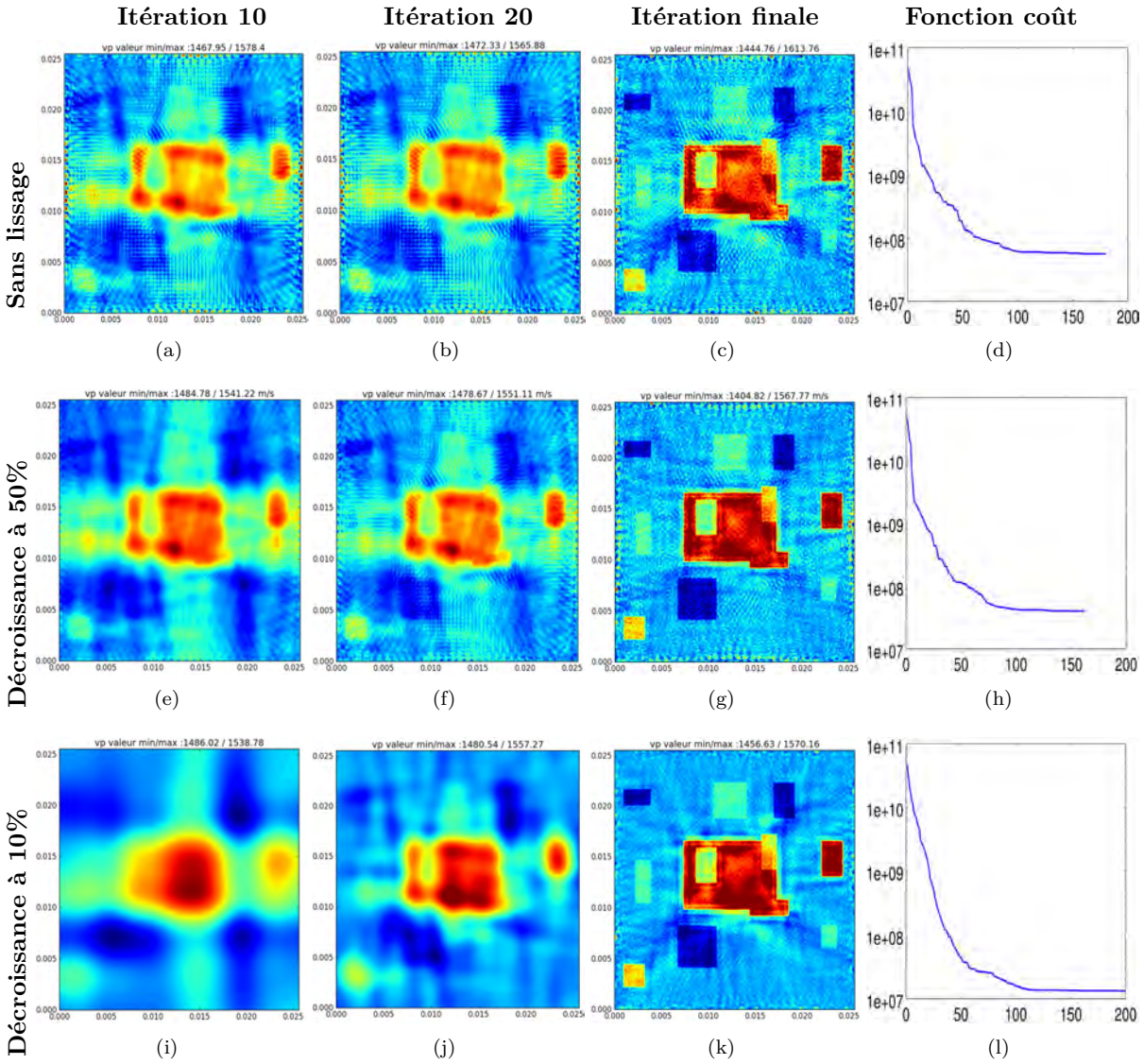
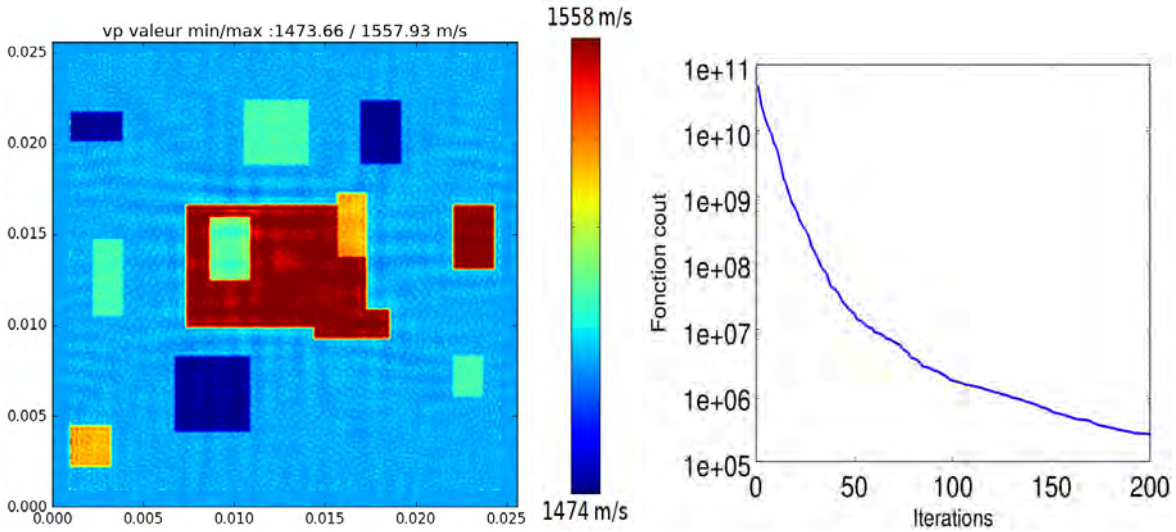


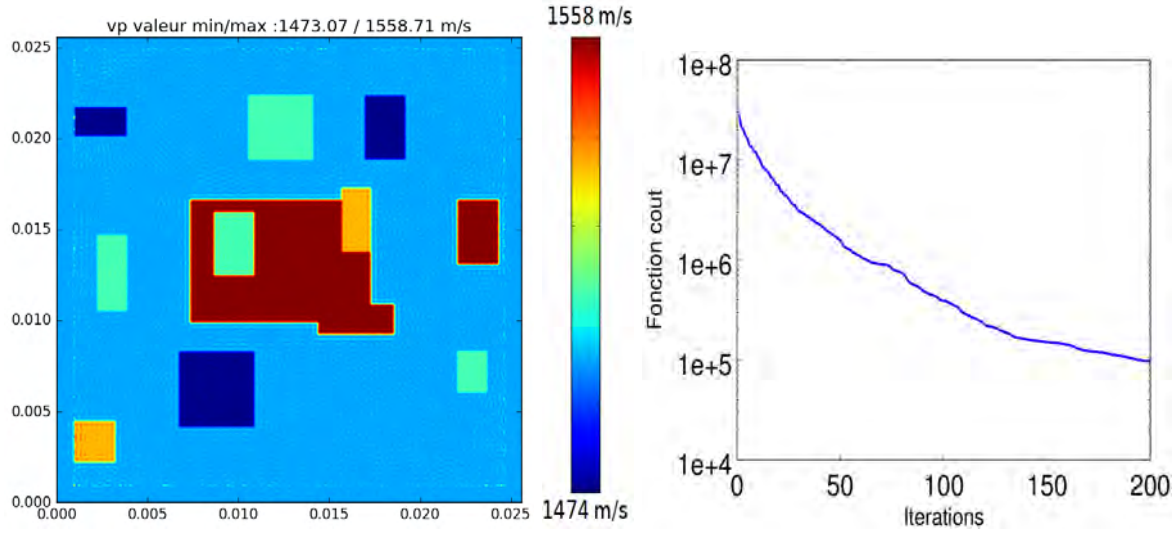
FIGURE 4.5 – Inversions de la forme d'onde pour différentes valeurs de rayons de lissage.

son émission. A cause de la très faible dispersion géométrique du champ adjoint en ces points, le gradient a tendance à y être sur-évalué. Pour pallier ce problème, nous allons appliquer à l'ensemble des éléments spectraux situés sur les bords du domaine un lissage Gaussien de rayon constant (d'environ la taille d'un élément spectral). Les résultats, présentés sur la figure 4.6, sont éloquents : par rapport à la version précédente (4.5(l)), la fonction coût 4.6(c) a décru de deux ordres de grandeur supplémentaires, et les oscillations visibles dans 4.5(k) sont nettement moins marquées.



(a) Modèle inversé (b) Vitesse (c) Fonction coût

FIGURE 4.6 – Inversion des données de la carte 4.4 avec lissage gaussien au niveau des points d'enregistrement de la pression.



(a) Modèle inversé (b) Vitesse (c) Fonction coût

FIGURE 4.7 – Inversion des données de la carte 4.4 en utilisant comme modèle initial le résultat de 4.6(a) et en ajoutant les ondes réfléchies aux données à inverser.

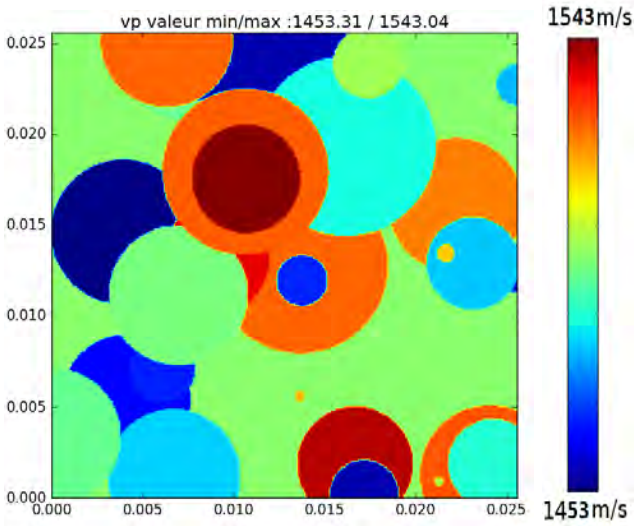
Le modèle inversé 4.6(a) ne contient que des informations d'ondes en transmission. A ce stade, il est possible d'augmenter la durée de la simulation dans le but d'augmenter la quantité d'in-

formation à inverser. En particulier, les ondes réfléchies pourront être prises en compte, ce qui permettra de bien définir la forme des interfaces du milieu. Pour que cela fonctionne correctement, il est important d'observer une décroissance assez nette de la fonction coût lors de la première étape d'inversion des ondes en transmission. Cela permet notamment de s'assurer que la carte de vitesse reconstruite ne contient pas de trop gros écarts par rapport à la réalité, ce qui évite de mal repositionner ces interfaces. Pour augmenter encore plus la quantité d'informations, les points d'enregistrement de la pression vont être placés tout autour du domaine, et pas seulement du côté opposé à celui de l'émission.

Comme prévu par la théorie, l'incorporation des données en réflexion augmente la résolution du modèle inversé. Les oscillations du modèle ne sont quasiment plus visibles, les contours sont bien définis et l'écart local entre vitesse réelle et vitesse inversée est très faible. On peut observer que la fonction coût 4.7(c) a une valeur initiale plus élevée que la valeur finale de la fonction coût 4.6(c), alors que le modèle est le même. Cela est dû à l'apparition de nouvelles données, dont la contribution à la fonction coût est de presque deux ordres de grandeurs plus élevé. Lors des premières itérations, le gradient reflétera donc une information issue de ces nouvelles données, et dont l'apport lors de la recherche du pas de descente sera aussi contrôlé par son impact sur les anciennes données. A titre de comparaison, nous avons évalué "l'ancienne" fonction coût, c'est-à-dire celle qui mesurait l'écart avec les données des 1100 premiers pas de temps, après la convergence de cette deuxième expérience. La valeur de cette dernière est de 2.710×10^4, ce qui montre que le nouveau modèle 4.7(a) est plus réaliste vis-à-vis des 1100 premiers pas de temps que le modèle 4.6(a), même si celui-ci a été obtenu avec une minimisation spécifique à ces données. Cela montre aussi que les nouvelles données contiennent une information supplémentaire vis-à-vis de la réalité.

Inversion d'un modèle sans point anguleux

Pour montrer la généralité de la méthode, nous allons changer le modèle objectif (modèle cible) :



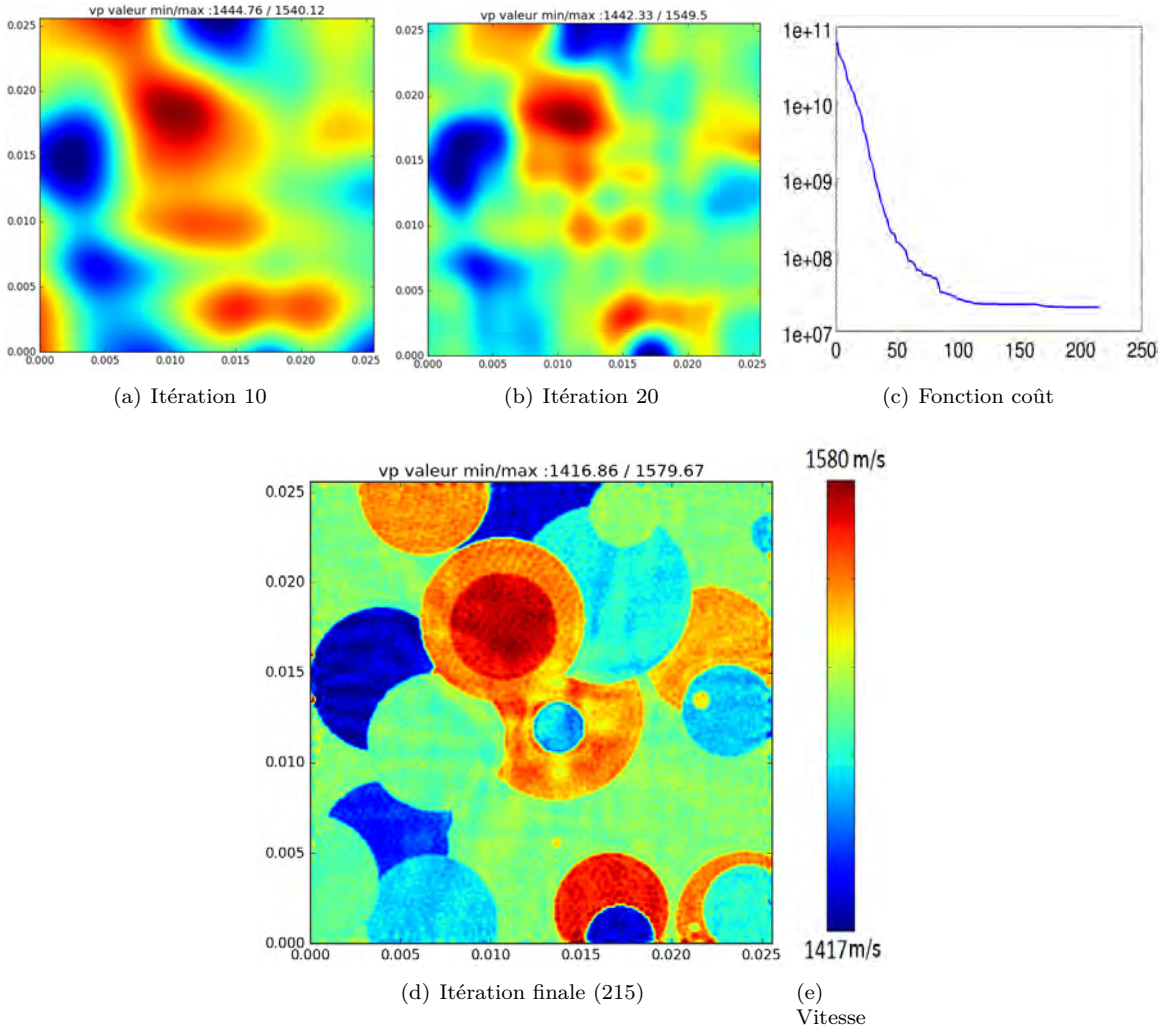
(a) Modèle (b) Vitesse

FIGURE 4.8 – Carte de vitesse du modèle recherché, sans point anguleux.

La carte de vitesse a été générée en définissant la position, le rayon et la vitesse à l'intérieur de 30 cercles de façon aléatoire, et de sorte à ce qu'on ait 1450 m/s$< c(\mathbf{x}) <$ 1550 m/s. L'intérêt de cette expérience est double :

— Ce modèle ne contient pas de points anguleux. Ces derniers influencent en effet particulièrement la propagation de l'onde et on s'interroge sur leur importance lors de l'inversion.

— La carte de vitesse, générée à partir d'un critère de distance à un point, ne dépend pas du maillage éléments spectraux initial. Dans la première simulation, le modèle avait été généré en utilisant le mailleur interne du code Specfem. Ainsi, une interface entre deux blocs de vitesse différente dans la réalité était également à la frontière de deux éléments spectraux. Par ailleurs, cette frontière était aussi présente dans le modèle initial, ce qui peut potentiellement influencer le processus d'inversion à correctement replacer l'interface.



(a) Itération 10 (b) Itération 20 (c) Fonction coût

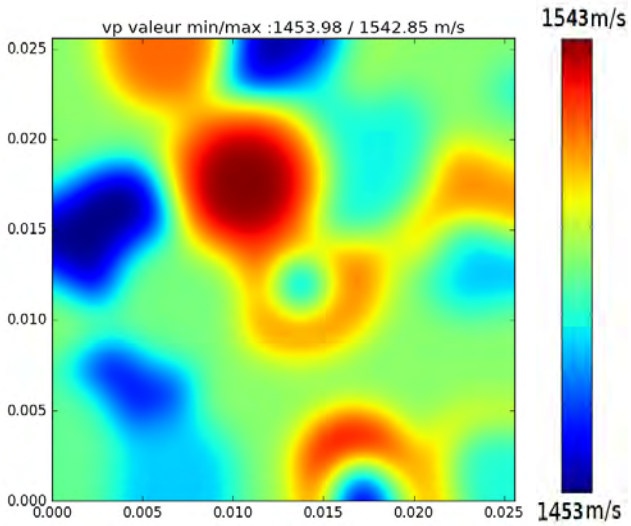
(d) Itération finale (215) (e) Vitesse

FIGURE 4.9 – Inversion de la carte de vitesse du modèle 4.8.

En appliquant un processus d'inversion similaire à celui de 4.5(l), on obtient un modèle inversé 4.9(d) assez proche du modèle recherché. Cependant, il n'a pas été possible d'utiliser le lissage au niveau des points d'enregistrement car le modèle à inverser possède des interfaces franches y compris au niveau de ces points. Dans la pratique, on peut considérer que le modèle initial pourrait être mieux caractérisé : les matériaux visibles sur les bords du domaine peuvent être identifiés et sont souvent caractérisés acoustiquement (vitesse de compression, atténuation...) .

Inversion d'un modèle sans discontinuités

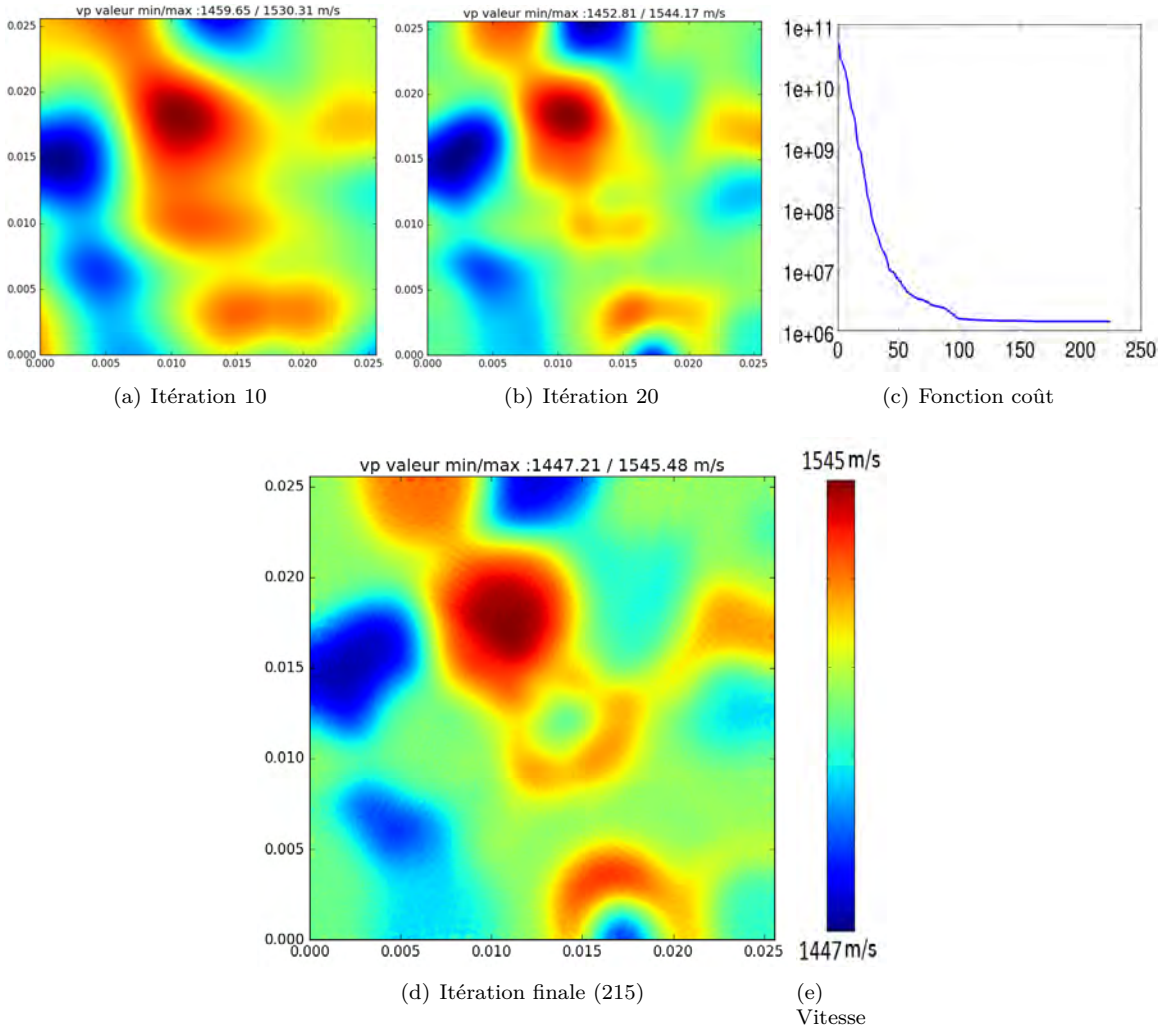
Dans l'expérience qui suit, le modèle objectif ne possède pas de discontinuités de vitesse. Il a été obtenu en lissant la précédente carte de vitesse objectif. Son allure est présentée sur la figure 4.10.



(a) Modèle (b) Vitesse

FIGURE 4.10 – Carte de vitesse du modèle recherché, sans discontinuités.

Les résultats sont assez concluants : par rapport à l'expérience précédente, la fonction coût a davantage décru, et on peut voir que les valeurs minimum et maximum du modèle inversé sont très proches de celles de l'objectif. De fait, la présence de discontinuités dans le modèle à inverser est une difficulté supplémentaire, car elles influencent fortement la propagation d'onde et ne sont pas simples à reconstruire à partir des gradients qui sont continus dès lors que le modèle initial l'est également.



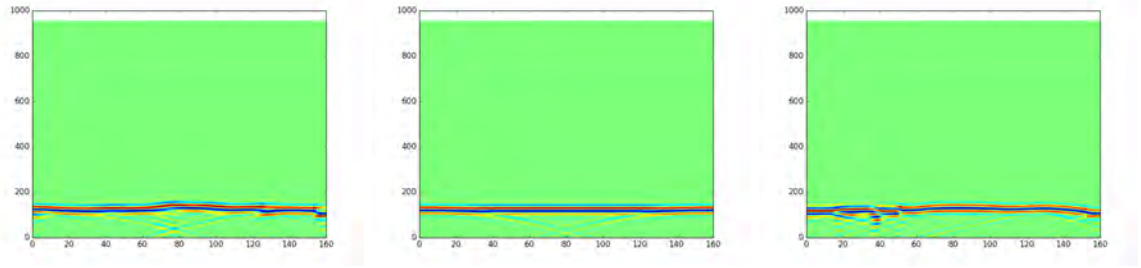
(a) Itération 10 (b) Itération 20 (c) Fonction coût

(d) Itération finale (215) (e) Vitesse

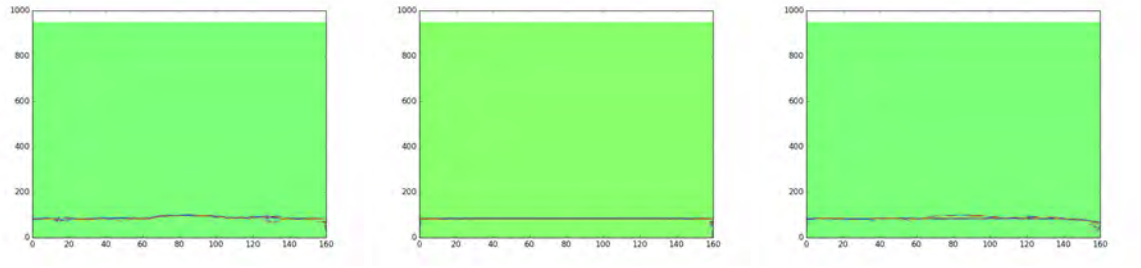
FIGURE 4.11 – Inversion de la carte de vitesse du modèle 4.10.

Inversion à 5 MHz

Pour se rapprocher des dispositifs expérimentaux disponibles au laboratoire PHASE, nous allons augmenter la fréquence dominante du signal source de 1 à 5 MHz. Comme nous l'avons vu, la condition du cycle skipping 3.3 qui dépend de cette fréquence nous impose des restrictions sur le type de domaine qu'il est possible de reconstituer. Comme on peut le voir sur la figure 4.12, la source du modèle adjoint à 1 MHz Fig. 4.12(c) est bien obtenue comme étant la différence des deux fronts d'onde simulés Fig. 4.12(a) et Fig. 4.12(b). A contrario, la source du modèle adjoint à 5 MHz Fig. 4.12(f) apparaît davantage comme étant la superposition de deux fronts d'onde distincts, qui focaliseront chacun sur les mauvaises zones du modèle numérique à cause d'un écart de vitesse trop important entre ce modèle et la réalité.



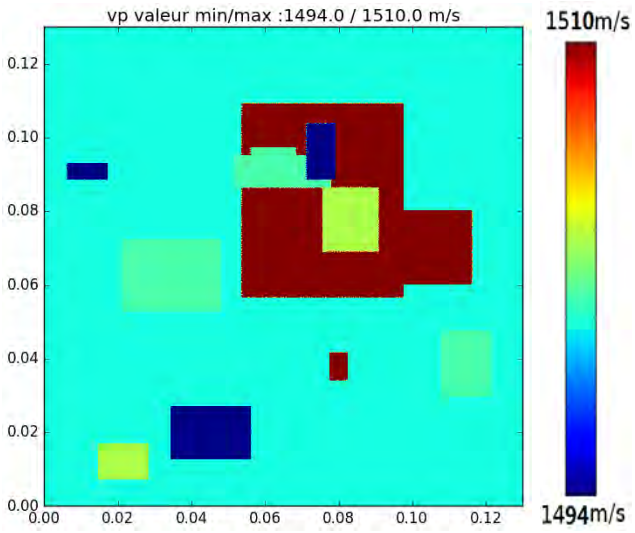
(a) Données réelles à 1MHz (b) Données synthétiques à 1MHz (c) Source adjointe à 1MHz



(d) Données réelles à 5MHz (e) Données synthétiques à 5MHz (f) Source adjointe à 5MHz

FIGURE 4.12 – Simulation des données à 1 et à 5 MHz pour le modèle 4.4(c). Les données simulées à 5 MHz ne respectent pas la condition de cycle skipping.

Pour respecter la condition de cycle skipping, on utilise des variations de vitesse beaucoup plus faibles qu'à 1 MHz. Nous proposons le modèle suivant :



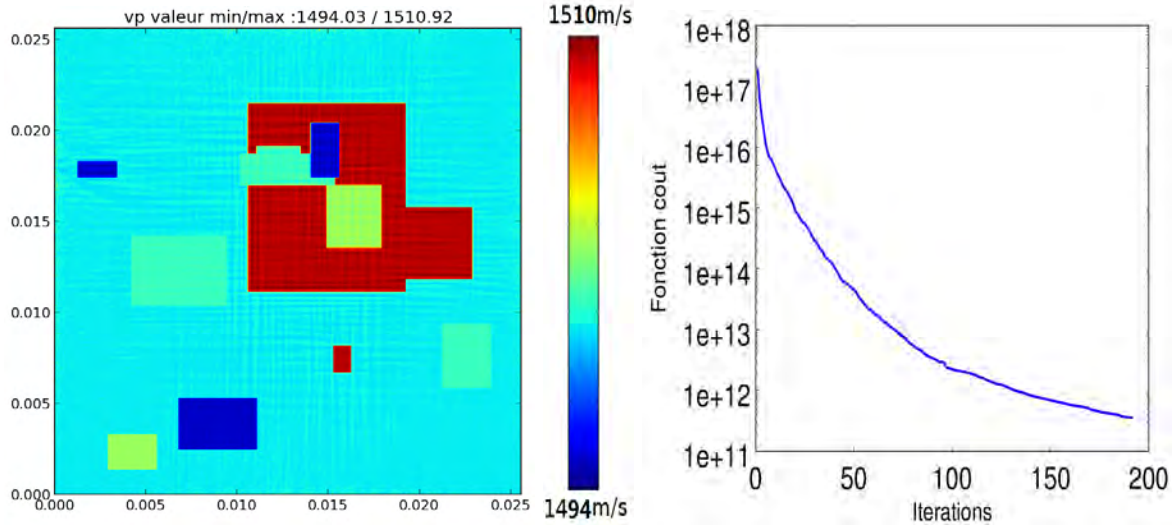
(a) Modèle (b) Vitesse

FIGURE 4.13 – Carte de vitesse à inverser à 5 MHz.

Comme on peut le voir, les variations de vitesse sur cette carte sont très faibles : elles s'échelonnent entre 1494 m/s et 1510 m/s sur un domaine de 2.56 cm $\times$ 2.56 cm. Par ailleurs, pour permettre une modélisation correcte du phénomène de propagation, un maillage de 300 $\times$ 300 éléments spectraux a été choisi, et la durée de simulation discrétisée en 12000 pas de temps pour respecter la condition de stabilité du schéma explicite.

Ainsi, avec une demi-période de 100 ns à 5 MHz, la condition de cycle skipping semble trop

pénalisante pour monter une expérience réelle qui soit compatible. De fait, s'il est possible de créer expérimentalement des gels de type PVA ayant une vitesse de propagation semblable à celle ce l'eau, il est difficile de la contrôler avec une précision de 1 m/s, sur des portions n'excédant pas quelques millimètres. De plus, l'absence d'une seconde barrette transductrice pour enregistrer les ondes en transmission est également trop pénalisante pour espérer obtenir un résultat expérimental convainquant. C'est pourquoi nous présentons dans ce chapitre uniquement des résultats issus d'expériences numériques.



(a) Itération finale (192) (b) Vitesse (c) Fonction coût

FIGURE 4.14 – Inversion à 5 MHz de la carte de vitesse du modèle 4.13.

On peut apprécier sur la figure 4.14(a) la qualité de reconstruction des interfaces alors que seules les ondes en transmission sont utilisées sur cette inversion, ce qui s'explique par la faible longueur d'onde principale de l'onde émise dans le milieu. A 0.3 mm, elle est environ 100 fois plus petite que la longueur du domaine à imager. Des oscillations légères sont également observables dans les zones homogènes, et semblent proportionnelles à la longueur d'onde locale. Elles peuvent être retirées en appliquant un léger floutage, qui peut être appliqué au modèle à un certain moment du processus itératif, par exemple dès que le rayon de lissage atteint un certain seuil, comme la taille d'un élément spectral. Remarquons aussi la forte décroissance de la fonction coût 4.14(c), d'environ 5 ordres de grandeur. Comme on peut le voir, celle-ci démarre à une valeur bien plus élevée ($\simeq 1 \times 10^{17}$) que l'inversion à 1MHz ($\simeq 10^{12}$), notamment parce que l'information est échantillonnée avec davantage de pas de temps et que les oscillations sont plus rapides à 5MHz.

Inversion à trois dimensions

Il s'agit à présent d'étendre à 3D les résultats obtenus jusqu'ici. Sur le plan mathématique et physique, son extension ne présente pas de difficulté particulière. Au contraire, le milieu inconnu est sondé dans davantage de directions, ce qui favorise la diversité de l'information reçue en un point donné du domaine. Les effets de refocalisation du champ adjoint à 3D sont par ailleurs plus importants qu'à 2D, ce qui favorise aussi l'inversion. Sur le plan numérique en revanche, un modèle de $80 \times 80 \times 80$ éléments spectraux représente un coût mémoire important : le nombre de points d'intégrations de GLL correspondant est de 64 millions, soit 240 Mo de mémoire vive pour stocker la carte d'un paramètre donné. En exécution, Specfem3D requiert trop de mémoire RAM pour qu'une simulation tienne sur une seule carte graphique. C'est pourquoi nous avons considéré un problème moins bien défini spatialement, discrétisé en $40 \times 40 \times 40$ éléments spectraux. Au prix d'une légère dégradation de la qualité de la simulation, cette configuration est compatible avec les

GPU et ne prend qu'une dizaine de secondes à l'exécution. Pour rendre compatible le cas à 3D avec des délais acceptables, il a également fallu transposer sur GPU la portion du code exprimant le lissage gaussien. Ce lissage, exprimé avec une convolution exprimée dans l'équation 3.36, ne peut être effectué dans le domaine de Fourier à cause du caractère non structuré du maillage. Son calcul s'opère de façon standard, et implique une double boucle sur les points d'intégration, car la valeur en chaque point d'intégration dépend de la valeur de tous les autres points du maillage. Dans le cas de notre maillage, cela équivaut à une boucle ayant $(125\times40\times40\times40)^2 = 6.4\times10^{13}$ itérations. Comme chaque itération implique un calcul de distance et l'évaluation d'une exponentielle, son coût numérique est élevé, de l'ordre de plusieurs dizaines d'heures sur CPU. Après implémentation sur GPU, le temps de calcul du lissage du gradient était d'environ 5 minutes, ce qui reste lourd, mais a permis de faire tourner une inversion complète à 3D en une dizaine d'heures avec 4 GPUs, ce qui aurait été impensable même avec plusieurs dizaines de CPUs.

Le modèle choisi pour l'inversion à 3D est le suivant :

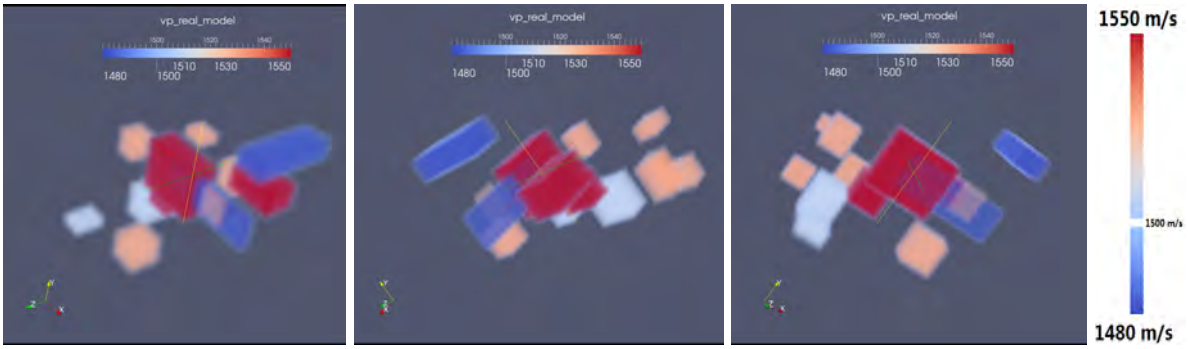


FIGURE 4.15 – Vues en coupe du modèle objectif à 3D. Pour plus de lisibilité, la vitesse de fond (1500 m/s) a été retirée du code couleur, laissant apparaître uniquement les hétérogénéités.

Des différents exemples à 2D que nous avons vus, ce cas de figure avec des discontinuités franches est le cas le plus difficile à traiter, et par ailleurs le plus facile à représenter avec la convention d'une vitesse de fond que l'on retire du code couleur. Le domaine à reconstituer est de taille 2.56 cm × 2.56 cm × 2.56 cm. Pour limiter le nombre de simulations et surtout augmenter l'applicabilité de la configuration, nous avons utilisé seulement quatre des six faces disponibles. On se base sur un modèle initial homogène avec $c = 1500$ m/s.

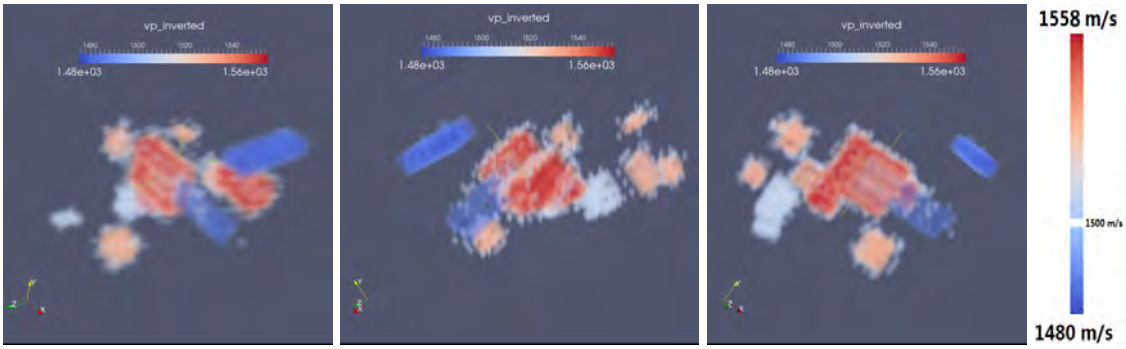


FIGURE 4.16 – Vues en coupe du modèle à 3D reconstruit. Des 'points' apparaissent autour des hétérogénéités, à cause de la continuité du modèle reconstruit.

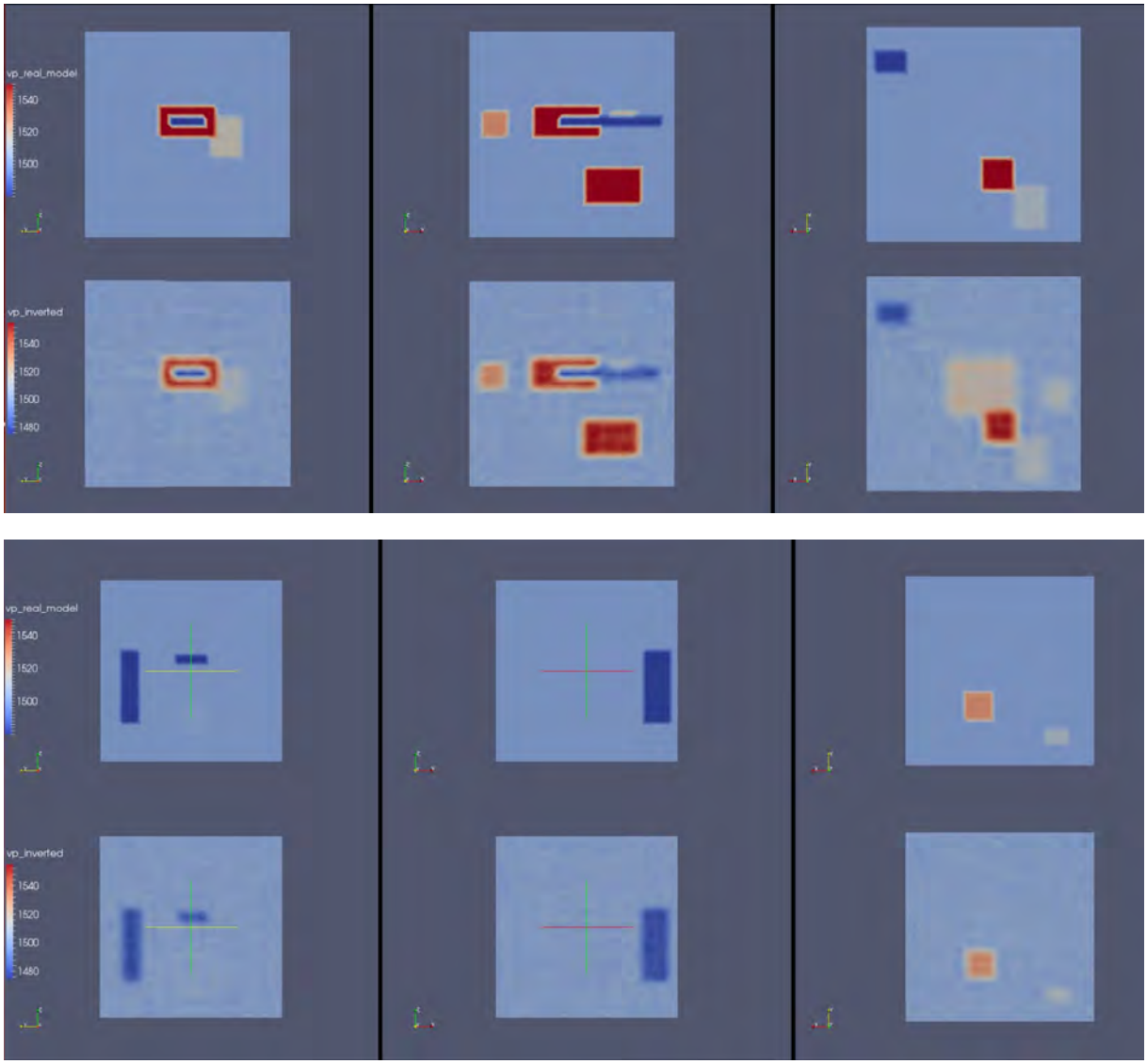


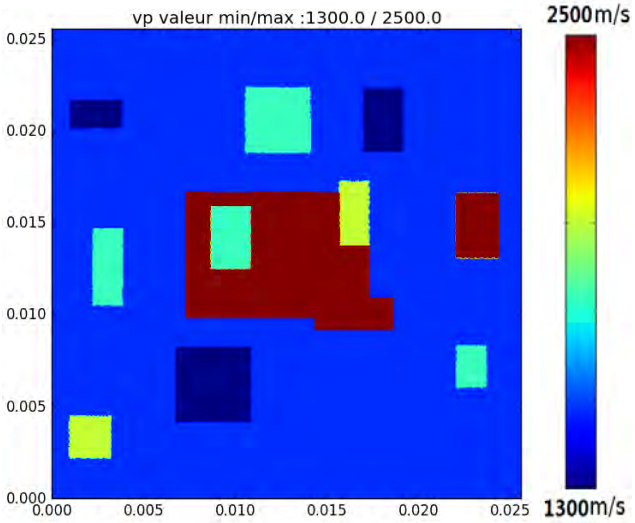
FIGURE 4.17 – Comparaison selon plusieurs plans de coupe orthogonaux entre le modèle réel (ligne du haut de chaque image) et le modèle reconstruit (ligne du bas).

Les figures 4.16 et 4.17 montrent les résultats de l'inversion à 3D. Comme on peut le voir, le résultat obtenu est proche de la réalité, les hétérogénéités ont une valeur de vitesse homogène et les interfaces, un peu moins marquées que dans le cas 2D, restent parfaitement identifiables. L'utilisation de seulement quatre ondes planes suffit à obtenir l'information nécessaire pour l'inversion. La fonction coût, non représentée ici, a décru d'environ 10 ordres de grandeur par rapport à sa valeur initiale.

4.3 Forts contrastes de vitesse

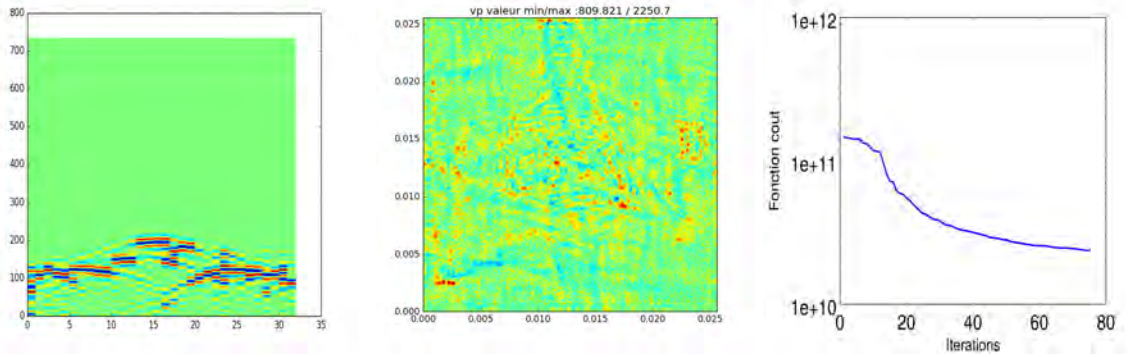
Dans cette partie, nous allons considérer des contrastes de vitesse plus élevés que précédemment. Le modèle objectif 4.18 contient notamment une large hétérogénéité centrale à 2500 m/s.

Modèle avec interfaces



(a) Modèle (b) Vitesse

FIGURE 4.18 – Carte de vitesse du modèle recherché, possédant des hétérogénéités avec un fort contraste.



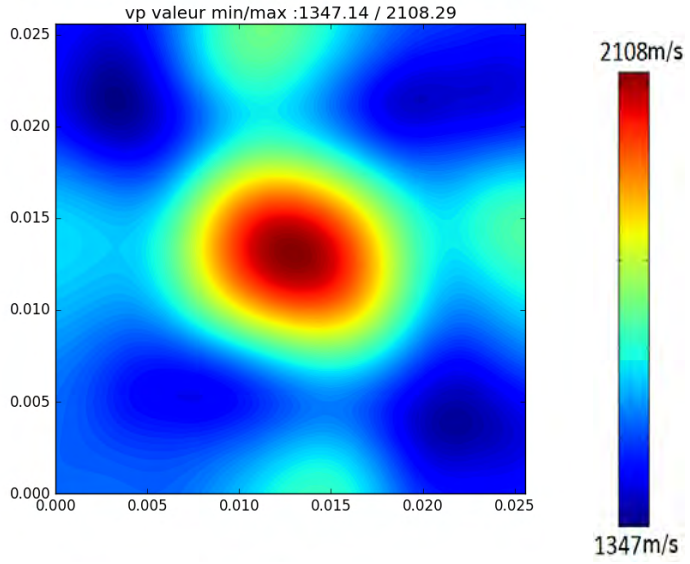
(a) Données à inverser (b) Modèle inversé (c) Fonction coût

FIGURE 4.19 – Tentative d'inversion des données de la carte 4.18 avec la même stratégie que pour les faibles contrastes.

La figure 4.19 montre le résultat de la tentative d'inversion en appliquant le même procédé que pour les faibles contrastes de vitesse, à savoir l'emploi d'un signal source constitué d'une seule ondelette de Ricker de fréquence dominante 1 MHz et combiné à une décroissance progressive du rayon de lissage. Après 70 itérations, la fonction coût 4.19(c) n'a décru que d'un ordre de grandeur, et on peut voir sur 4.19(b) que les directions de descente choisies ne sont pas les bonnes : l'inversion a été piégée dans un minimum local. En observant les données 4.19(a), on peut s'apercevoir que le front d'onde transmis s'est décomposé en plusieurs fronts d'onde sous l'effet de la diffraction, avec des temps de vol différents à cause des gros écarts de vitesse à l'intérieur du modèle. Sans un modèle initial déjà assez proche de la réalité, il n'y a aucune chance pour que ces données respectent la condition de cycle skipping.

Ainsi, la technique de reconstruction du modèle réel doit être repensée pour ce nouveau cas de figure. Compte tenu des dimensions du modèle objectif, on peut estimer la fréquence centrale de l'ondelette de Ricker formant le signal source à utiliser pour que la condition de cycle skipping

3.29 soit respectée. Celle-ci se trouve autour de 100 kHz. En appliquant le processus d'inversion, on obtient :



(a) Modèle (b) Vitesse

FIGURE 4.20 – Résultat de l'inversion de la carte de vitesse 4.18 en utilisant comme signal source une ondelette de Ricker de fréquence dominante 100kHz, qui respecte la condition de cycle skipping.

Comme on peut le voir, si l'allure globale est la bonne, le résultat obtenu figure 4.20 ressemble à une version floutée du modèle recherché. Cela s'explique par la valeur de la longueur d'onde locale, qui nous informe sur le potentiel de résolution spatiale possible qui est de l'ordre de $\frac{\lambda}{2}$: pour $c = 1500$ m/s, on a $\frac{\lambda}{2} = 0.75$ cm, ce qui représente plus d'un quart de la longueur du domaine objectif. Alors, la seule façon de procéder pour augmenter la résolution spatiale est d'augmenter le contenu fréquentiel du signal source.

L'algorithme proposé pour l'inversion du modèle 4.18 se base sur l'augmentation progressive du contenu fréquentiel du signal émis, afin de toujours respecter la condition de cycle skipping. L'idée est d'utiliser à son tour le modèle 4.20 comme modèle initial pour une inversion d'un nouveau jeu de données, avec une fréquence dominante du signal source un peu plus élevée, et ainsi de suite jusqu'à l'obtention d'une résolution spatiale satisfaisante. Après diverses expérimentations numériques, nous sommes arrivés à l'algorithme suivant :

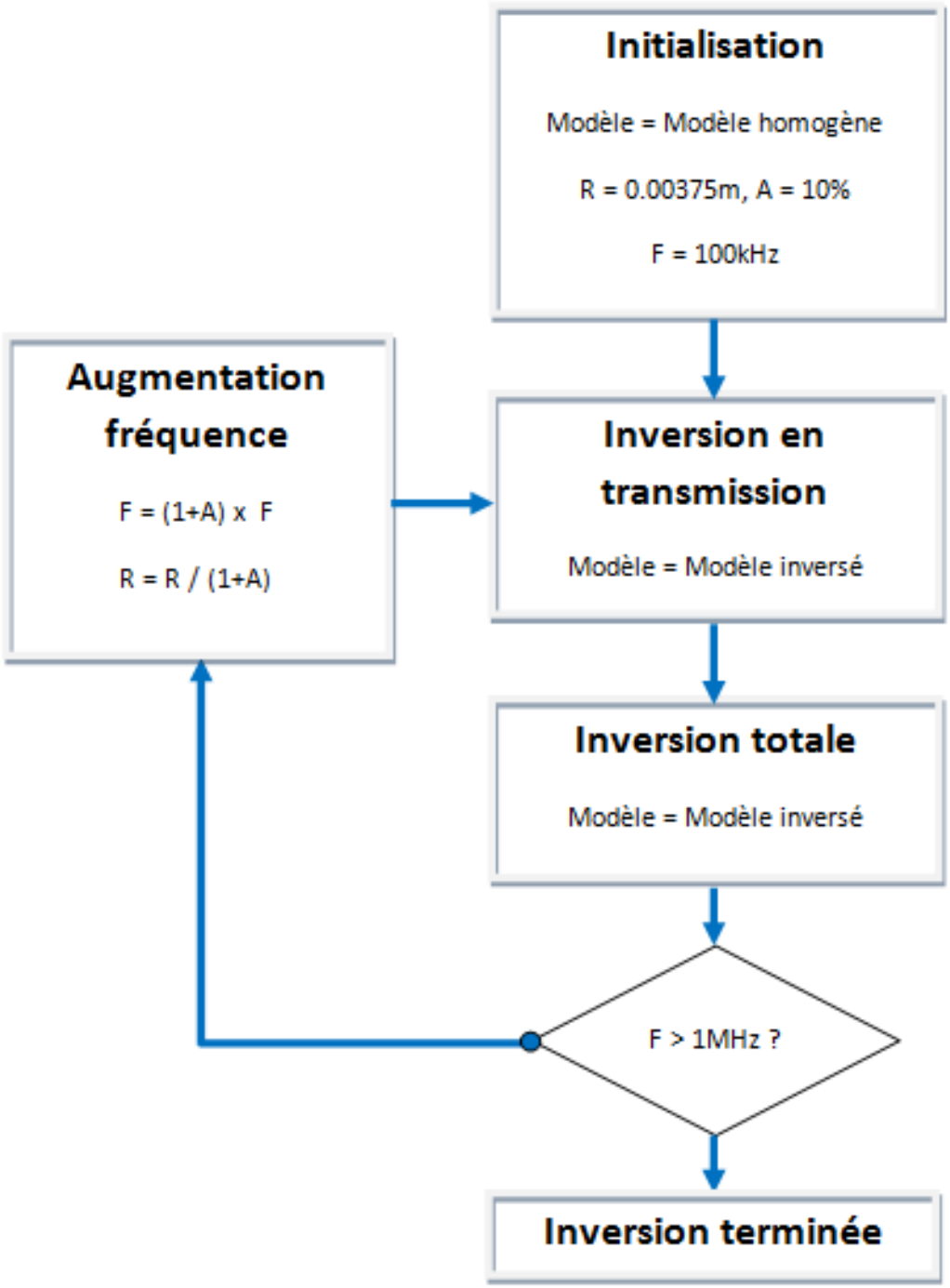
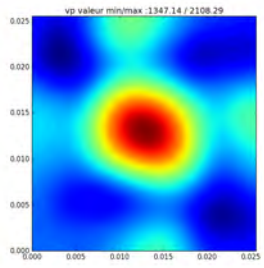
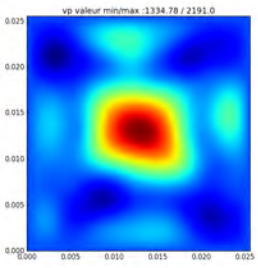


FIGURE 4.21 – Algorigramme de l'inversion de la carte de la figure 4.18. Les constantes utilisées ont été ajustées pour ce problème précis.

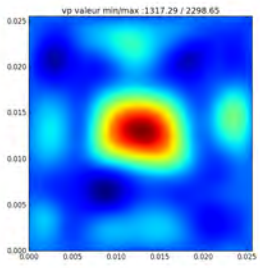
Cet algorithme a la particularité de faire intervenir deux phases lors de l'inversion à une fréquence dominante donnée : on inverse d'abord les ondes en transmission puis la totalité des ondes (transmission + réflexion). Par ailleurs, le rayon de lissage R_0 initial et son taux de décroissance A sont choisis en fonction de la longueur d'onde locale λ : on veille à ce qu'à chaque itération i on ait $(1-A)^i R_0 > \frac{\lambda}{2}$ afin de limiter les oscillations du gradient.



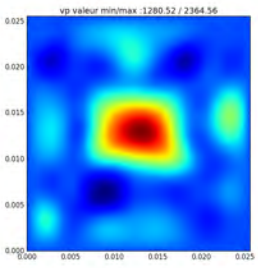
(a) 100kHz, en transmission



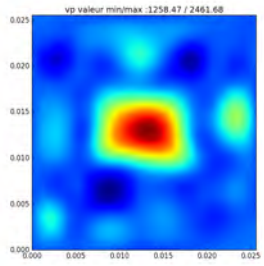
(b) 100kHz



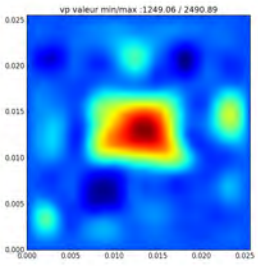
(c) 115kHz, en transmission



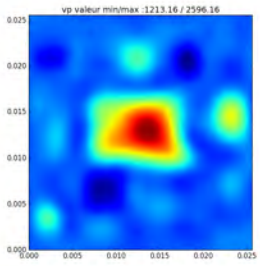
(d) 115kHz



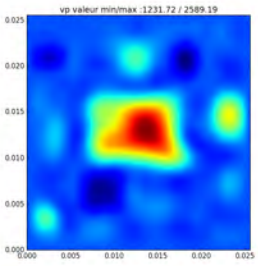
(e) 132kHz, en transmission



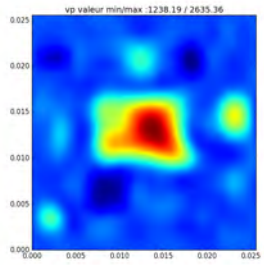
(f) 132kHz



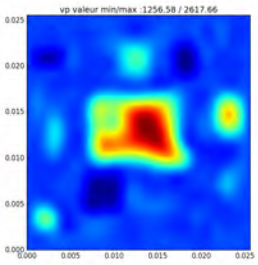
(g) 152kHz, en transmission



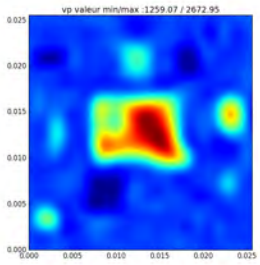
(h) 152kHz



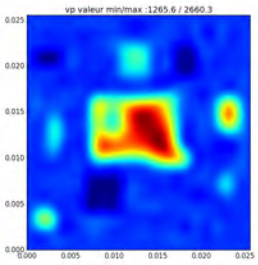
(i) 174kHz, en transmission



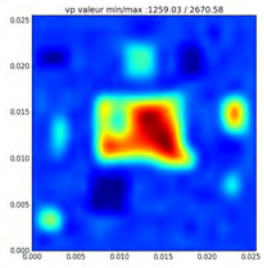
(j) 174kHz



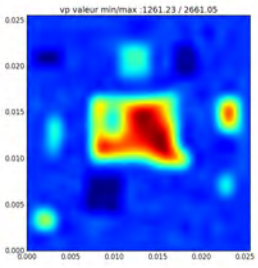
(k) 201kHz, en transmission



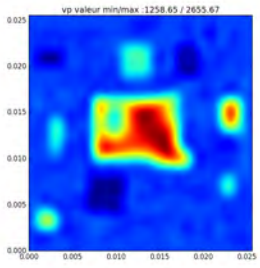
(l) 201kHz



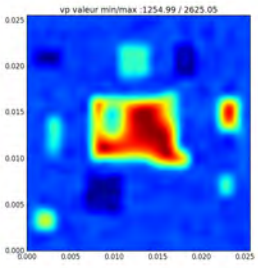
(m) 231kHz, en transmission



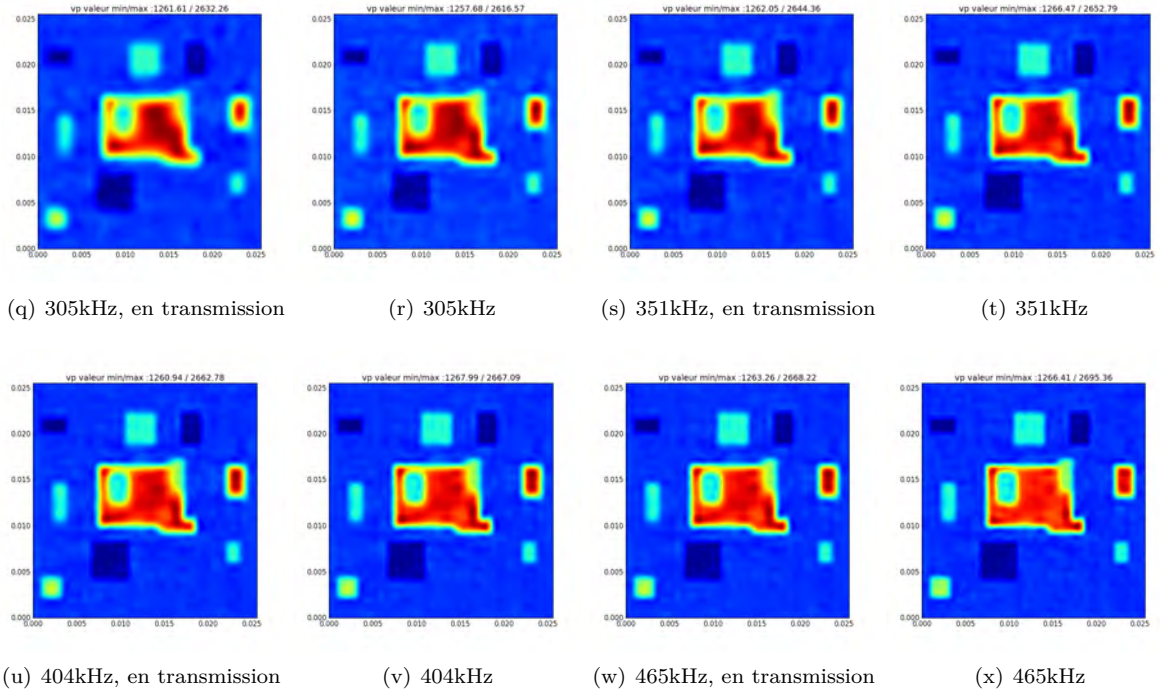
(n) 231kHz



(o) 266kHz, en transmission



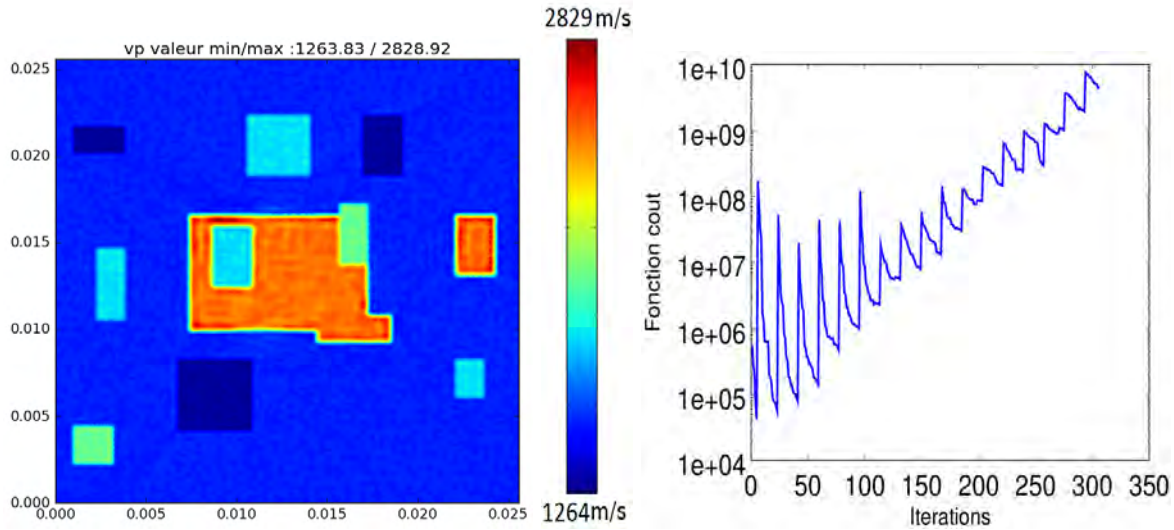
(p) 266kHz



(q) 305kHz, en transmission (r) 305kHz (s) 351kHz, en transmission (t) 351kHz

(u) 404kHz, en transmission (v) 404kHz (w) 465kHz, en transmission (x) 465kHz

FIGURE 4.22 – Formation progressive de la carte de vitesse au cours du processus itératif fréquentiel. Chaque image représente le modèle après inversion à une fréquence donnée, et est utilisée comme modèle initial à la fréquence supérieure.



(a) Modèle inversé (b) Vitesse (c) Fonction coût

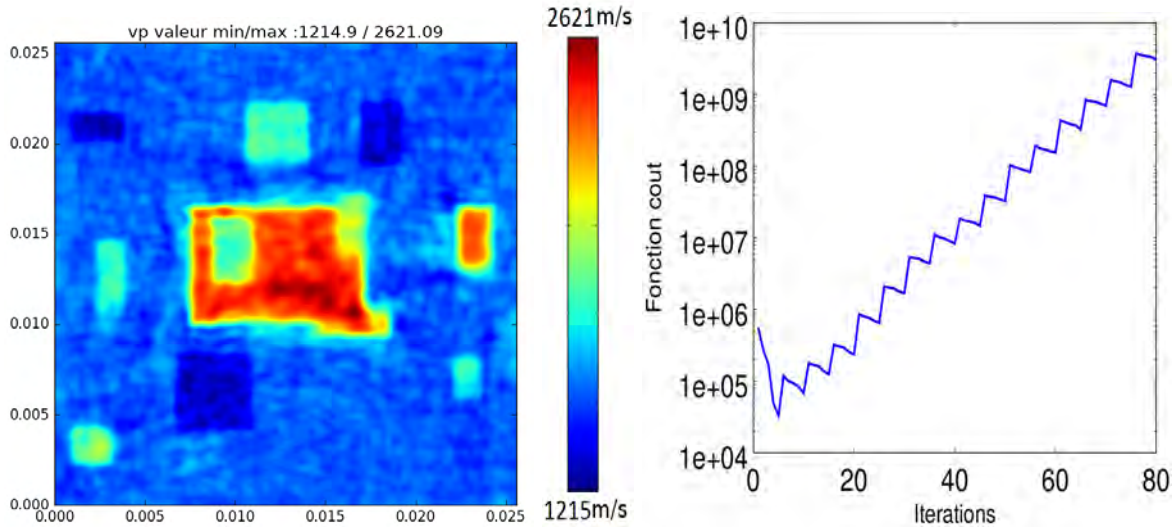
FIGURE 4.23 – Résultat de l'inversion multi-fréquentielle de 4.18.

Comme on peut voir sur la figure 4.22, la convergence vers le modèle réel est très progressive : la structure globale de la carte de vitesse apparaît d'abord, puis au fur et à mesure de la montée en fréquence, et donc de la diminution de la longueur d'onde, les contours des hétérogénéités se précisent. On peut observer sur le modèle inversé 4.23(a) que ces interfaces sont bien positionnées, ce qui aurait été impossible sans une reconstruction précise de la carte de vitesse.

La fonction coût 4.23(c) a une allure en dents de scie car au fur et à mesure de l'inversion, les

jeux de données à inverser ne sont pas les mêmes : ils changent quand la fréquence augmente et surtout quand l'inversion utilise aussi les données en réflexion, qui correspondent ici aux itérations où la fonction coût augmente fortement. Même si l'ajout des nouvelles données accroît davantage la fonction coût par rapport à la décroissance induite par l'inversion, le modèle inversé 4.23(a) semble bien converger vers le bon modèle. Cette observation possible grâce à la connaissance du modèle à reconstruire montre aussi la difficulté à établir un critère de convergence à partir de la fonction coût pour ce type de problème. Une solution peut consister à évaluer les nouveaux modèles par rapport aux anciens jeux de données (de plus basses fréquences) pour s'assurer que la direction de descente qui est choisie est consistante avec ces derniers.

Nous avons constaté à partir des expériences numériques la supériorité de la stratégie d'une augmentation fréquentielle progressive combinée avec une inversion en transmission puis en réflexion. Les tentatives d'inversion utilisant immédiatement des données en transmission et en réflexion s'avèrent diverger dès les premières fréquences. Par ailleurs, la stratégie d'une inversion uniquement en transmission converge, mais la position des différentes hétérogénéités n'est pas la bonne au cours des premières itérations, ce qui se traduit par des oscillations plus importantes dans le modèle final inversé 4.24(a). La fonction coût 4.24(c) croît à une allure similaire à la précédente fonction coût 4.23(c), ce qui montre que l'absence de données en réflexion n'empêche pas la convergence.

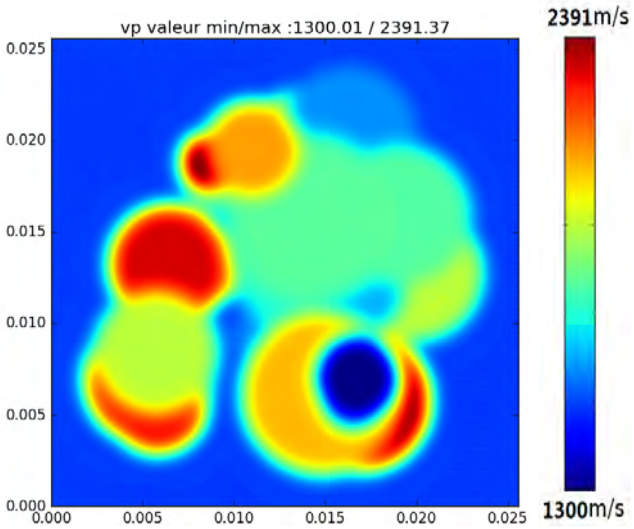


(a) Modèle inversé (b) Vitesse (c) Fonction coût

FIGURE 4.24 – Résultat de l'inversion multi-fréquentielle uniquement en transmission du modèle 4.25.

Modèle sans interfaces

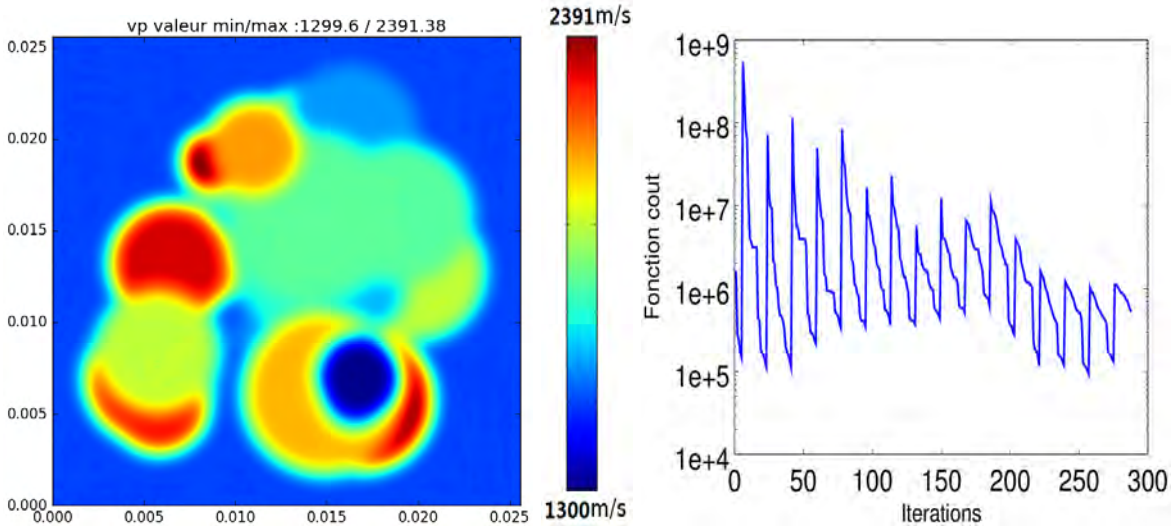
Tout comme pour les faibles écarts de vitesse, nous allons observer la qualité de l'inversion pour une carte de vitesse avec des variations continues. Le modèle à inverser est le suivant :



(a) Modèle (b) Vitesse

FIGURE 4.25 – Carte de vitesse objectif à variations continues. L'échelle de vitesse s'étale de 1300 m/s à 2391 m/s.

Le processus d'inversion est le même que pour le modèle précédent, dans la mesure où les mêmes écarts de temps de vol entre ondes observées et ondes simulées sont à prévoir.



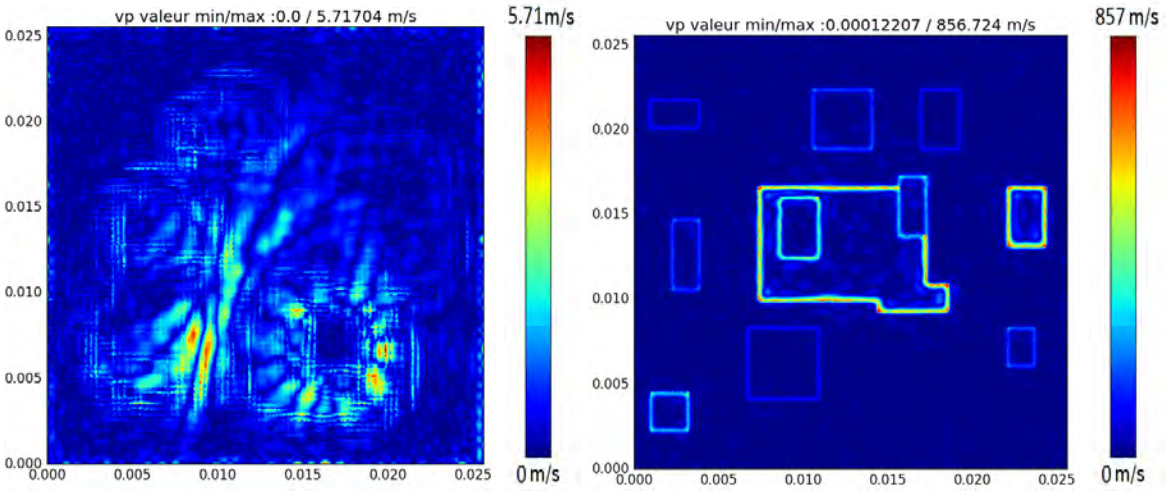
(a) Modèle inversé (b) Vitesse (c) Fonction coût

FIGURE 4.26 – Résultat de l'inversion multi-fréquentielle du modèle 4.25.

Tout comme pour les faibles contrastes de vitesse, la qualité de l'inversion du modèle continu est sensiblement meilleure par rapport au modèle avec discontinuités. Il est difficile de faire la différence entre le modèle réel 4.25 et le modèle inversé 4.26(a). L'échelle de vitesse reconstruite est presque la même, avec un minimum de 1299.6 m/s (resp. 1300.01) et un maximum de 1391.38 m/s (resp. 1391.37). On peut aussi remarquer que la fonction coût 4.26(c) décroît plus rapidement, si bien que sa valeur moyenne ne semble pas augmenter en dépit d'une forte augmentation de la fréquence dominante. Pour s'assurer à nouveau de la convergence de l'inversion, nous testons la valeur de la fonction coût avec le modèle inversé en utilisant la fréquence dominante initiale de

100 kHz. Sa valeur se situe à $9 \times 10^{+0}$, ce qui illustre le bon choix des directions de descente au cours de la montée en fréquence.

Cette expérience numérique confirme la relative simplicité de l'inversion d'un modèle continu par rapport à un modèle discontinu. On peut s'en convaincre avec ces observations supplémentaires dans l'espace du modèle :



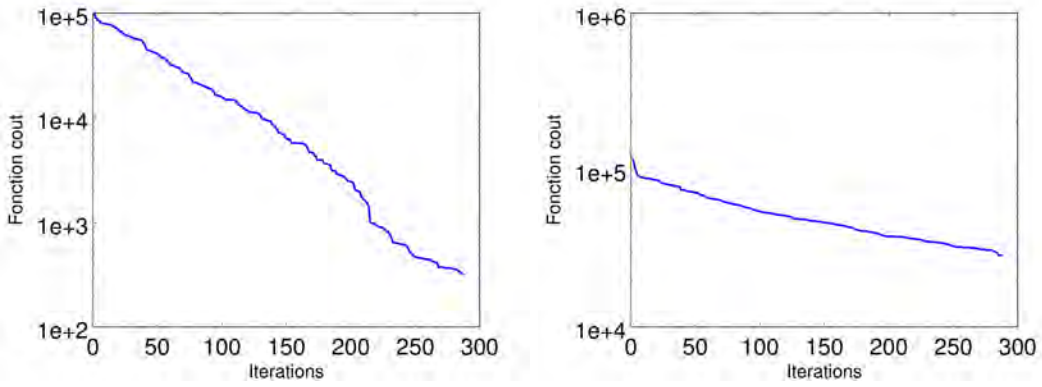
(a) Modèle continu

(b) Vitesse modèle continu

(c) Modèle avec interfaces

(d) Vitesse modèle avec interfaces

FIGURE 4.27 – Valeur absolue de la différence entre modèle réel et modèle inversé pour les cas continus 4.27(a) et discontinus 4.27(c). L'échelle de couleur utilisée n'est pas la même ([0 ;5.7] et [0 ;856.7]).



(a) Modèle continu

(b) Modèle avec interfaces

FIGURE 4.28 – Évolution de la norme L^2 de la différence entre modèle réel et modèle courant.

Dans la mesure où ces expériences sont entièrement numériques, il est possible de représenter la différence entre le modèle réel et le modèle inversé (4.27) et de tracer la norme L^2 de l'écart entre modèle à l'itération i et modèle réel. Ces informations ne sont pas utilisées dans le processus d'inversion mais permettent d'apprécier la qualité de l'inversion. Sur 4.27(a), on peut apprécier le très faible écart entre le modèle réel et le modèle reconstruit. Celui-ci est inférieur à 6 m/s sur la totalité du domaine, qui est constitué de 160000 points, alors que l'écart initial pouvait atteindre 800 m/s par endroits. L'observation de ce faible écart et la décroissance régulière de la fonction coût 4.28(a) sur plusieurs ordres de grandeurs, malgré une carte de vitesse aux variations

continues mais très aléatoires, montre que l'algorithme de reconstruction proposé est tout à fait adapté au problème.

Dans le cas discontinu, on peut observer sur la figure 4.27(c) que de gros écarts subsistent entre le modèle réel et le modèle inversé au niveau des interfaces. L'écart est directement proportionnel au contraste de vitesse entre les deux zones. On peut remarquer que la largeur de ces traits est sensiblement égale à la longueur d'onde associée à la dernière fréquence dominante utilisée, ce qui est cohérent dans la mesure où on ne peut pas s'attendre à obtenir une précision supérieure à $\frac{\lambda}{2}$. On peut supposer que c'est la même raison qui explique le faible écart persistant sur le modèle continu : les plus larges écarts apparaissent là où le modèle varie fortement et où il est difficile au gradient qui ne bénéficie pas d'une résolution spatiale infinie de rendre compte de brusques variations inférieures à $\frac{\lambda}{2}$, même si elles sont continues. Au niveau du processus d'inversion sur le modèle discontinu, on peut penser que la source adjointe est particulièrement élevée à cause de l'écart sur la position précise des interfaces tout au long de l'inversion. C'est ce qui explique sans doute le fait que la fonction coût 4.23(c) augmente au cours de la progression fréquentielle.

4.4 Vers l'inversion de données réelles

Changement de philosophie par rapport au dispositif expérimental échographique

Comme nous l'avons évoqué, la mise en place d'un dispositif expérimental permettant la reconstruction d'une carte de vitesse réelle n'a pas été possible à cause de la difficulté à respecter la condition de cycle skipping 3.29 avec des transducteurs piézoélectriques haute fréquence destinés à l'échographie. En imagerie échographique, l'idée est d'envoyer un signal aussi transitoire que possible dans le but d'assimiler les fronts d'onde réfléchis en tant qu'interfaces ou hétérogénéités du milieu, à partir d'une connaissance a priori de la vitesse du milieu. Pour l'inversion, l'information est essentiellement apportée par les ondes en transmission, et si les hautes fréquences sont aussi appréciées pour leur résolution spatiale, elles ne peuvent être utilisées que lorsque le modèle de propagation reconstruit est suffisamment proche de la réalité, ce qui n'est jamais le cas avec les fréquences actuellement utilisées en échographie. Au delà du Mégahertz, l'écart maximal de temps de vol ne doit pas excéder quelques centaines de nanosecondes, ce qui est très difficile à respecter même avec de bonnes conditions expérimentales. A cette échelle, de faibles écarts de température peuvent influer localement sur la vitesse de compression, et rendre cette technique impossible à utiliser si la dimension du domaine à inspecter excède une dizaine de centimètres. De même, une imprécision d'un dixième de millimètre sur la distance réelle entre émetteur et récepteur peut conduire à une divergence de l'algorithme.

Dans les expériences numériques, la notion de distance entre modèle courant et modèle réel est assez simple à déterminer, en devinant l'écart maximum possible de temps de vol de l'onde émise en fonction des écarts de vitesse dans les deux modèles grâce à la connaissance de ce modèle objectif. Pour une expérience réelle, il faudra faire des hypothèses sur le milieu, et en particulier se donner des dimensions et des écarts de vitesse maximum, afin de déterminer à l'aide de la condition de cycle skipping quelle est la fréquence dominante appropriée pour commencer le processus d'inversion. Pour de grands écarts, il sera nécessaire de commencer à des fréquences bien plus basses que celles utilisées couramment en échographie. Dans notre exemple, un écart de 1000 m/s sur seulement 1 cm de long nous a conduit à travailler à 100 kHz. On peut penser que de nombreux cas concrets, et notamment l'imagerie des os dans le domaine médical, les écarts de temps de vol à prévoir situent la fréquence dominante correspondante dans le domaine audible, voire dans la partie basse de son spectre. Par ailleurs, il ne faut pas descendre non plus trop bas en fréquence, au risque d'augmenter inutilement le coût en temps déjà conséquent associé à l'inversion.

Ainsi, il appartient à l'expérimentateur d'adapter la fréquence initiale en fonction du problème qu'il souhaite résoudre, et pour des incertitudes fortes sur la carte de vitesse à déterminer, une approche multi-fréquentielle est grandement souhaitable. Ce cas de figure peut nécessiter le recours

à un spectre de plusieurs décades, ce qui n'est pas forcément compatible avec des transducteurs de type piézoélectrique, lesquels sont plutôt à faible bande passante. On peut penser que dans le cas de variations fortes, les transducteurs de technologie CMUT (Capacitive Micromachined Ultrasonic Transducers) pourraient jouer un rôle important grâce à leur largeur de bande.

En attendant de pouvoir monter un dispositif expérimental réunissant les hypothèses nécessaires, nous avons mené quelques expériences numériques supplémentaires afin mieux prendre en compte la réalité de l'expérience.

Une première différence entre l'inversion de données réelles et des données simulées sur ordinateur est bien entendu liée aux conditions souvent idéales de la propagation numérique, dès lors que les conditions de stabilité du schéma numérique sont respectées. Afin de rendre ces données plus "réelles", nous allons les bruiter, mais aussi les digitaliser : dans la pratique, le convertisseur analogique-numérique ne retranscrit pas une valeur de pression à virgule flottante (7 chiffres significatifs dans le cas de la simple précision), mais discrétise cette valeur sur seulement 10 ou 12 bits. L'absence de virgule flottante, liée à la difficulté de changer la sensibilité de l'appareil en temps réel, pourrait s'avérer pénalisante pour l'inversion des fronts d'onde d'amplitude modérée.

Digitalisation

Comme attendu, la digitalisation des données expérimentales affecte la qualité de reconstruction proportionnellement au nombre de niveaux de digitalisation que l'on se donne. On peut d'ailleurs voir que la fonction coût 4.29(d) diminue d'un ordre de grandeur supplémentaire par rapport à 4.29(b) avec une information quatre fois plus précise, car digitalisée sur 2 bits supplémentaires. Cela s'explique avec le carré de la fonction coût, qui suggère qu'une information quatre fois plus précise diminue d'un facteur 16 l'écart entre deux jeux de données, ce qui correspond bien à un ordre de grandeur d'écart.

Il est surtout important d'observer que la dégradation du jeu de données n'empêche pas la convergence de l'algorithme pour les premières itérations, et qu'ainsi la structure globale de la carte de vitesse est retrouvée.

Bruit

Pour simuler la présence de bruit au cours de l'expérimentation, nous allons ajouter au signal simulé un bruit gaussien dont la variance σ sera un certain pourcentage de l'amplitude maximum de ce signal. Le but de cette expérience n'est pas de montrer comment réduire le bruit, mais simplement de montrer son influence sur le processus d'inversion.

La présence d'un bruit gaussien de variance à 1% de l'amplitude maximale affecte assez fortement le processus itératif, avec une fonction coût 4.30(b) qui ne décroît plus que d'un seul ordre de grandeur. On peut toutefois reconnaître sur la carte de vitesse 4.30(a) les formes générales des différentes hétérogénéités, avec des valeurs de vitesse différentes, mêmes si ces dernières sont entachées de très fortes oscillations. La présence de bruit gaussien dans les données engendre en particulier l'apparition de très hautes fréquences dans la source adjointe. Comme le maillage est d'une taille modérée, cela génère un fort bruit numérique, qui a tendance à se propager à une vitesse plus faible que la vitesse de propagation de l'onde. De plus, ce bruit est rajouté sur la totalité du signal, ce qui implique que la source adjointe génère en permanence des oscillations qui n'ont pas de sens pour l'inversion. Cela se traduit par un champ adjoint qui perd de son caractère transitoire, et engendre des corrélations non désirées avec le champ direct, et le kernel calculé se trouve alors aussi bruité.

Il doit être possible à travers des techniques de seuillage et de filtrage fréquentiel de diminuer l'impact du bruit sur l'inversion. Cependant, ce problème n'est pas forcément évident à traiter à haute fréquence, car le nombre d'échantillons par période n'est pas très élevé et un filtrage peut fortement affecter les données, en modifiant à la fois phase et amplitude.

Une façon simple de réduire le bruit associé aux acquisitions expérimentales est de moyenner différentes acquisitions. Comme la durée d'acquisition est très faible et que le temps de calcul

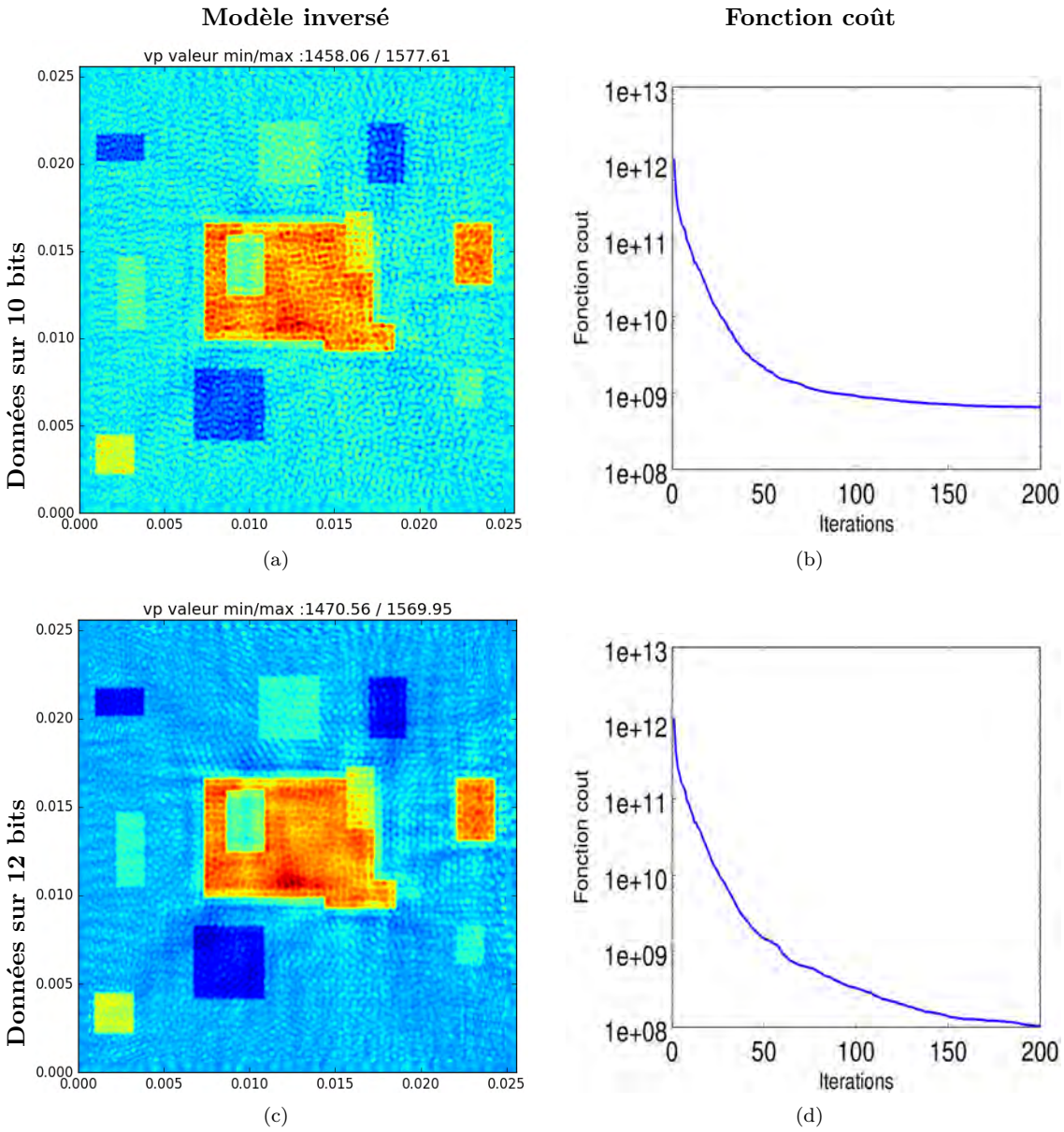
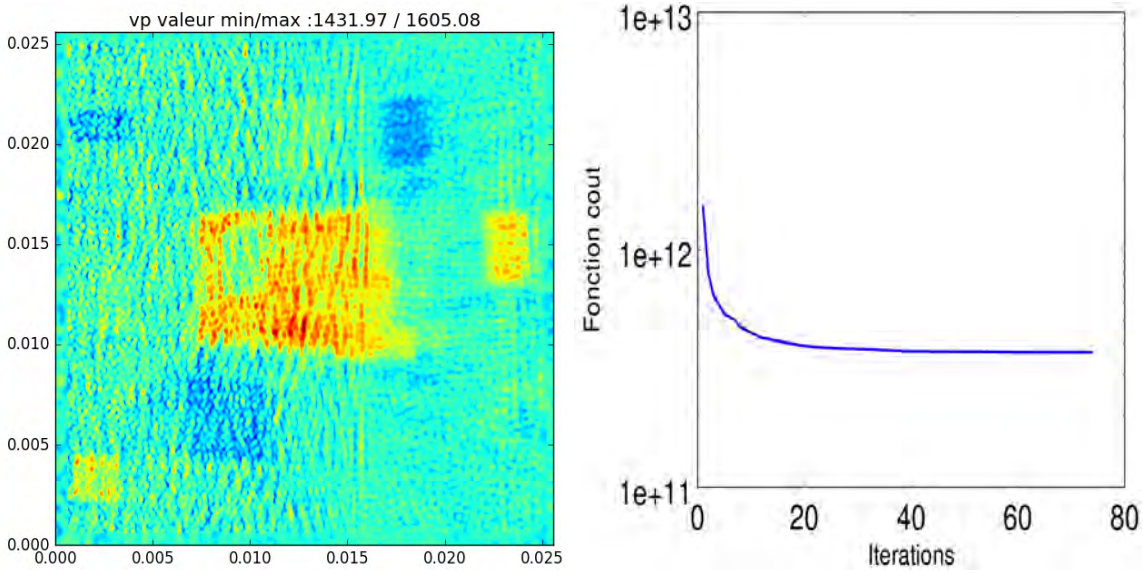


FIGURE 4.29 – Inversion du modèle 4.4(c) avec des données digitalisées.



(a) Modèle inversé

(b) Fonction coût

FIGURE 4.30 – Inversion du modèle 4.4(c) à partir de données contenant un bruit gaussien de variance égale à 1% de l'amplitude maximale de chacun des signaux simulés.

associé à l'inversion est très important, le moyennage semble être une solution pertinente dans ce contexte. On peut voir que la fonction coût 4.31(b) décroît d'un ordre de grandeur supplémentaire par rapport à 4.30(b), ce qui se ressent sur la qualité du modèle inversé. On peut alors se permettre d'augmenter le niveau de bruit, en passant la variance à 10% du maximum. La fonction coût 4.31(d) ne décroît presque pas mais les formes très générales du domaine apparaissent. A ce niveau de bruit, une augmentation du nombre d'expériences devrait être utilisée pour augmenter la qualité des données.

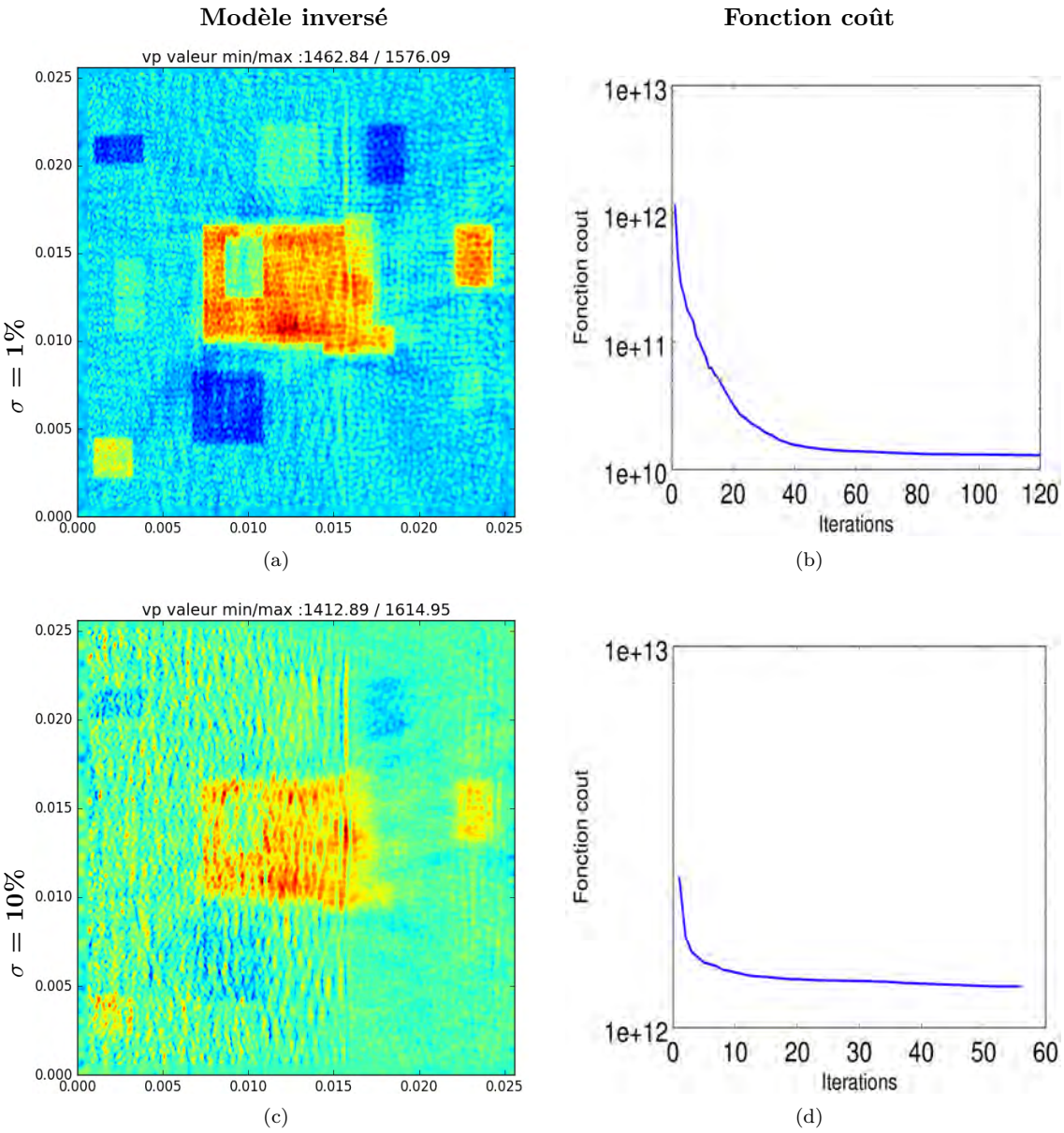


FIGURE 4.31 – Inversion du modèle 4.4(c) en moyennant 30 acquisitions bruitées.

Conclusions et perspectives

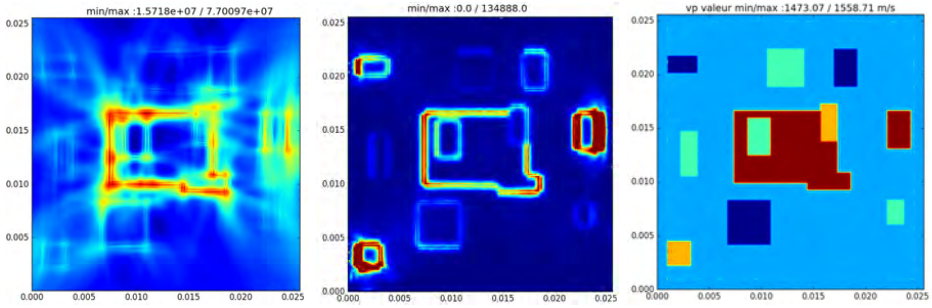
A travers cette thèse, nous avons balayé les diverses méthodes d'imagerie ultrasonore existantes, et utilisé le GPU comme accélérateur de calcul pour deux champs d'application complémentaires que sont l'imagerie échographique et l'imagerie quantitative.

En imagerie échographique, dont le but est de révéler rapidement les contrastes d'impédance à l'intérieur d'un milieu inconnu, nous avons choisi l'accélération de la méthode d'imagerie FTIM (Fast Topological IMaging). Ce procédé présente la particularité de faire intervenir un modèle de propagation d'onde, dont on peut choisir le mode de calcul selon le problème.

Comme il s'agit du cas le plus courant en imagerie échographique, nous avons ciblé le cas des milieux où la vitesse de compression peut être considérée comme homogène. Cela a permis l'emploi du calcul semi analytique qui, combiné à la vitesse de calcul du GPU, a ouvert la porte de l'imagerie en temps réel pour des milieux à 2D, et du quasi temps réel à 3D ($\simeq$ 1s). Ce calcul semi-analytique basé sur l'évaluation numérique de l'intégrale de Rayleigh ne souffre d'aucune approximation, par opposition à la quasi-totalité des méthodes concurrentes, basées sur l'approximation géométrique de l'équation d'onde. Grâce à la précision de ce modèle, une seule illumination du milieu suffit pour obtenir une image échographique précise, ce qui constitue un avantage substantiel pour de nombreuses applications. Par ailleurs, la prise en compte des effets de fréquence finie autorise une utilisation en champ proche, alors que les autres méthodes sont pénalisées par leur approximation qui génère la présence de "moustaches" sur l'image résultat. Un premier travail d'optimisation a consisté à observer les symétries du problème, en particulier au niveau du diagramme de rayonnement, afin de réduire le nombre d'opérations arithmétiques nécessaires. La seconde partie de l'optimisation a consisté à repenser l'écriture de l'algorithme FTIM pour une forme adaptée au paradigme de programmation sur GPU, ce qui a mené à des vitesses d'exécution de l'ordre de 1.5 téraFLOPS sur une carte graphique haut de gamme (Nvidia Titan Black de génération Kepler), soit un temps de calcul 1500 fois plus rapide que sur CPU à configuration égale. Le bénéfice apporté par le temps réel peut être à présent pleinement exploité par diverses applications comme la PIV (Particle Image Velocimetry) dont on peut espérer dans un avenir proche voir des dispositifs offrir aux expérimentateurs de mécanique des fluides une imagerie du champ de vitesse du fluide en temps réel.

L'extension du temps réel au cas des milieux hétérogènes constitue un nouveau défi. Le calcul des diagrammes peut être effectué grâce à un solveur de type éléments finis, et convertis dans le domaine fréquentiel. La principale difficulté est liée à l'utilisation d'autant de diagrammes de rayonnement qu'il y a de sources / récepteurs. On peut espérer voir ce problème résolu par les prochaines générations de GPU, qui devraient profiter d'un changement de technologie en matière de mémoire globale (de la DDR5 vers la HBM), et voir leur bande passante augmenter significativement. Le cas échéant, une stratégie multi-GPU avec des transferts asynchrones pourrait être une solution. L'avènement du temps réel pour ces types de problèmes où les hypothèses des méthodes échographiques sont trop simplificatrices pour espérer une image convenable ouvrirait la voie à de très nombreuses applications.

L'imagerie quantitative, qui permet d'obtenir une information chiffrée à propos des paramètres physiques du milieu, a une portée plus générale que l'imagerie échographique car elle vise à obtenir



(a) Image échographique conventionnelle (b) Image avec FTIM (c) Imagerie quantitative par FWI

une information qui est dans la pratique entachée de nombreuses approximations. Le champ d'application possible s'élargit fortement avec l'emploi de l'inversion de la forme d'onde complète. Son application à l'échelle ultrasonore était jusqu'ici très faiblement employée à cause de son coût numérique très élevé, car impliquant des centaines de simulations de propagation d'onde avec des solveurs de type éléments finis. Un effort important pendant cette thèse a été consenti à la transposition du code Specfem, lequel implémente la méthode des éléments spectraux pour la résolution de l'équation d'onde, en passant de la version parallélisée sur CPU vers une version GPU/multi-GPU. Le facteur d'accélération obtenu est de l'ordre de 10 entre la version parallèle pour CPU avec une utilisation sur 8 cœurs (impliquant un double processeur Intel Xeon E5-2609 v2 cadencé à 2.5 GHz), et la version GPU (exécutée sur une Nvidia Titan Black avec 2880 cœurs cadencés à 889 MHz). Grâce à cette accélération, le coût d'une inversion standard à 2D est passé d'une demi-journée à une demi-heure, ce qui a considérablement facilité les travaux de recherche menés dans la suite de cette thèse.

L'inversion complète de la forme d'onde est formulée à partir de la minimisation de l'écart quadratique entre données réelles et données simulées, lesquelles dépendent des paramètres du modèle de propagation. Sa résolution s'articule dans ce travail autour de la méthode d'optimisation locale L-BFGS qui n'a besoin que du gradient des données par rapport à ces paramètres. A l'aide de la méthode de l'adjoint, ces dérivées sont calculées en seulement trois simulations numériques de propagation d'onde.

Des expériences numériques ont démontré la possibilité de reconstruire à 2 ou 3 dimensions des cartes de vitesse de compression de milieux inconnus, sans connaissance a priori des positions ni même des valeurs potentielles des vitesses de compression dans les hétérogénéités. Cette nouvelle approche requiert un changement de dispositif expérimental par rapport à l'imagerie échographique conventionnelle, qui utilise la même barrette de transducteurs à l'émission et à la réception. A la place, une seconde barrette de transducteurs doit être placée de l'autre côté du milieu à imager, afin de pouvoir enregistrer les ondes en transmission, c'est-à-dire les ondes réfractées par le milieu. Par ailleurs, les fréquences du signal source émis doivent être suffisamment hautes pour obtenir une bonne résolution spatiale, mais aussi suffisamment basses pour respecter la condition de cycle skipping, qui est une contrainte supplémentaire héritée de la non-convexité du problème d'optimisation sous-jacent. Dans de nombreux cas pratiques, cette condition implique l'utilisation de fréquences beaucoup plus basses qu'en imagerie échographique.

Lorsque le milieu réel n'observe que des faibles variations de vitesse (de l'ordre de 5%), il est possible de reconstruire avec précision cette carte en utilisant une seule acquisition par capteur, avec un contenu spectral modéré ($\simeq$ 1 octave). Pour des variations plus fortes, une stratégie multi-fréquentielle est inévitable pour arriver à faire converger le problème vers une solution bien définie spatialement. Plusieurs émissions par source sont nécessaires afin d'avoir un spectre fréquentiel pouvant couvrir plusieurs décades, tout en conservant le caractère transitoire des signaux enregistrés.

L'application de la FWI à l'échelle ultrasonore reste un vaste chantier : si cette thèse a mis en évidence sa viabilité au travers d'expériences numériques, son application expérimentale reste à

démontrer. Par ailleurs, la résolution du problème peut s'articuler autour d'autres fonctions coûts que celles de type waveform, afin d'augmenter la vitesse de convergence qui reste pénalisante (30 minutes à 2D, 10 heures à 3D). Enfin, le choix des paramètres offre diverses opportunités d'extension de la méthode à un cadre plus général : le problème peut être résolu pour un domaine élastique et faire intervenir la vitesse de cisaillement, ou choisir la densité, la porosité, le tenseur d'anisotropie ou encore l'atténuation comme carte de paramètres à reconstruire.

Annexe A

Paradigme de programmation sur GPU

Pour pouvoir implémenter du code sur GPU, il est indispensable de connaître la structure globale du composant hardware pour comprendre sa philosophie. Le but de cette annexe est d'introduire les notions nécessaires ainsi que le vocabulaire lié à la programmation GPU, qui est utilisé par l'API CUDA.

A.1 Relation avec le CPU

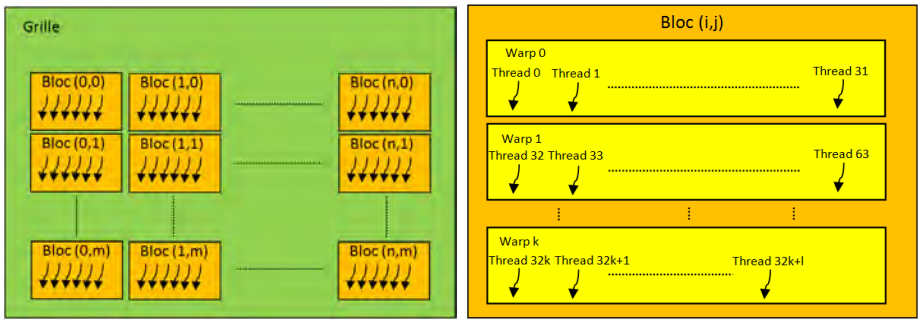
Le CPU, ou host, peut être vu comme l'ordonnanceur des différentes instructions : il gère comme en programmation conventionnelle des opérations de chargement en mémoire vive des données provenant du disque dur, de même que des opérations d'écriture sur le disque. Ensuite, le programmeur est responsable de définir quelles données sont transférées de la mémoire vive, gérée par le CPU, vers la mémoire globale du GPU, communément appelé device. Le CPU ordonne ensuite le déclenchement des différents calculs sur le GPU, et peut gérer simultanément le transfert des données entre la RAM et la mémoire globale du GPU. De même, plusieurs programmes peuvent tourner simultanément sur le GPU, et il appartient au programmeur de déterminer comment balancer de la façon la plus efficace possible transferts et calculs.

A.2 Architecture matérielle du GPU et code associé

FIGURE A.1 – Architecture matérielle d'un CPU et d'un GPU. Images issues de [66].

La hiérarchie entre CPU et GPU étant bien établie, on peut s'intéresser à l'architecture du GPU et à la structure des fonctions destinées à être exécutées sur GPU, nommées kernels.

Tout d'abord, comparons l'architecture d'un CPU et d'un GPU qui sont représentées figure A.1. Comme on peut le voir, le nombre d'unités de calcul (ou ALU) sont bien plus nombreuses sur le GPU, et la quantité de mémoire privée associée à une ALU est bien plus petite. Lorsqu'un kernel est lancé, le jeu d'instructions correspondant est exécuté simultanément sur l'ensemble des unités de calcul disponibles. L'exécution du kernel se répartit en effet sur une grille à 2 ou 3 dimensions (de taille $m \times n$ dans le schéma A.2(a)) qui est composée de blocs. Chacun de ces blocs contient un nombre identique de threads, qui ne peut excéder 1024 dans un même bloc. Les threads y sont regroupés par multiples de 32, au sein d'un warp (A.2(b)). Une unité de calcul exécute un thread à la fois. A l'intérieur du kernel, des variables prédéfinies permettent de récupérer le numéro du bloc et le numéro du thread dans le bloc, qui sont indispensables pour accéder aux tableaux situés en mémoire globale.



(a) (b)

FIGURE A.2 – Schématisation de l'exécution d'un kernel.

Au sein d'un warp, les threads s'exécutent en simultané sur des unités de calcul adjacentes. Comme chaque thread est responsable du chargement en mémoire d'une donnée du tableau, on veille toujours à respecter également l'adjacence des données récupérées dans les tableaux de la mémoire globale, à savoir que si le thread i accède à la case $x + i$ d'un tableau, alors le thread $i + 1$ doit accéder à la case $x + i + 1$ de ce même tableau. Si cette condition n'est pas respectée, il n'est pas possible au contrôleur d'assurer un accès simultané aux données à chacun des threads et impacte la performance. Par ailleurs, pour tirer le meilleur parti de la bande passante interne, qui caractérise la quantité maximale de données transmissible de la mémoire globale vers les registres, il est également recommandé que le thread 0 d'un warp accède à une case du tableau qui soit un multiple de 32.

Pour une performance optimum, une autre règle à respecter est que le code du kernel permette une exécution de code identique sur toutes les ALU chargées d'un warp, avec éventuellement des données différentes. Ainsi, le premier frein à la performance est la divergence de branche, qui arrive dès lors que les instructions à exécuter dépendent d'un test qui fait intervenir le numéro du thread. Lorsque cela est inévitable, des fonctions de synchronisation disposées en certains points du code permettent aux threads de s'attendre, dans le but d'une exécution resynchronisée dans la partie du code suivant leur appel.

A.3 Les différents types de mémoire

L'accès aux données, qui peut déjà être un facteur limitant en programmation conventionnelle, est un facteur d'autant plus critique en programmation sur GPU car chaque cœur de calcul ne dispose pas d'autant de mémoire cache qu'il en est attribué à chacun des quelques cœurs de calcul dont disposent les CPU. Par ailleurs, beaucoup moins de transistors sont alloués à la prédiction des instructions comme sur CPU. C'est pourquoi la mise en cache des données se fait de façon manuelle, et que sa connaissance et sa bonne gestion est indispensable.

Bien qu'il en existe davantage, nous distinguerons ici trois types de mémoire : la mémoire

Type de mémoire	**Quantité typique sur un GPU**	Vitesse d'accès	Durée de vie	But
Mémoire globale	4Go	400-600 cycles	Application	Entrées/sortie CPU-GPU
Mémoire partagée	1.5Mo	Quelques cycles	Bloc	Échange de données entre threads
Registre	4Mo	Quelques cycles	Thread	Registre privé

TABLE A.1 – Résumé des propriétés des différents types de mémoire. Sur GPU, le passage de la mémoire globale vers les mémoires plus proches physiquement des unités de calcul se fait de façon manuelle.

globale, la mémoire partagée, et la mémoire des registres. Leurs propriétés, qui sont résumées dans A.1, va justifier la façon d'utiliser chacune d'entre elles. A cause de sa latence élevée, les accès à la mémoire globale doivent être minimisés, et si une donnée de la mémoire globale est amenée à être utilisée plusieurs fois, elle doit être stockée dans la mémoire partagée ou des registres. Cependant, à cause de leur faible quantité par rapport au nombre d'ALU, la déclaration d'une nouvelle variable locale dans un kernel est à utiliser avec parcimonie.

La mémoire partagée permet quant à elle la communication de valeurs entre thread appartenant à un même bloc. Très souvent, il est nécessaire que des threads se communiquent certaines valeurs de leurs registres privés, par exemple parce chaque thread du warp doit utiliser les mêmes n valeurs d'un tableau situé en mémoire globale. S'il est possible que chaque thread accède n fois à la mémoire globale, il est plus judicieux de charger lors d'une phase d'initialisation chaque thread de remplir une case de la mémoire partagée avec une donnée du tableau. La mémoire globale n'est alors accédée qu'une seule fois. Ainsi, la communication entre threads permet d'augmenter la quantité d'information issue de la mémoire globale pour un thread donné, ce qui limite l'inoccupation du thread. Par ailleurs, la quantité nombre d'opérations flottantes par octet chargé, nommée intensité arithmétique, sert souvent d'indicateur pour estimer si l'algorithme est limité par la bande passante. Si cette quantité est trop faible, il faut généralement repenser la structure de l'algorithme et attribuer davantage de calcul dans le kernel, tout en veillant à ne pas déclarer trop de nouveaux registres.

Bibliographie

[1] P. Curie, J. Curie. Développement, par pression, de l'électricité polaire dans les cristaux hémihèdres à faces inclinées. *Comptes rendus des séances de l'Académie des Sciences*, 91 :294–295, 1880.

[2] P. Langevin, C. Chilowsky. Procédés et appareils pour la production de signaux sous-marins dirigés et pour la localisation à distance d'obstacles sous-marins. *Brevet français n502.913*, 1916.

[3] M. Karaman, P. Li, M. O'Donnell. Synthetic aperture imaging for small scale systems. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 42(3) :429–442, 1995.

[4] S. Bannouf, S. Robert, O. Casula, C. Prada. Data set reduction for ultrasonic tfm imaging using the effective aperture approach and virtual sources. In *11th Anglo-French Physical Acoustics Conference (AFPAC 2012)*. IOP Publishing Journal of Physics : Conference Series, 2012.

[5] R. Ten Grotenhuis, A. Hong. Imaging the weld volume via the total focus method. *Canada 4th International CANDU In-service Inspection Workshop and NDT in Canada 2012 Conference*, 2012.

[6] F. Wu, J-L. Thomas, M. Fink. Time reversal of ultrasonic fields. il. experimental results. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 39(5) :567–578, 1992.

[7] JL. Thomas, F. Wu, M. Fink. Time reversal focusing applied to lithotripsy. *Ultrasonic Imaging*, 18(2) :106–121, 1996.

[8] C. Fan, M. Caleap, M. Pan, BW. Drinkwater. A comparison between ultrasonic array beamforming and super resolution imaging algorithms for non-destructive evaluation. *Ultrasonics*, 54(7), 2014.

[9] Y Labyed, L. Huang. Ultrasound time-reversal music imaging with diffraction and attenuation compensation. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 59(10) :186–200, 2012.

[10] A. Achim, A. Bezerianos, P. Tsakalides. Novel bayesian multiscale method for speckle removal in medical ultrasound images. *IEEE Transactions On Medical Imaging*, 20(8), 2001.

[11] Y. Yu. Speckle reducing anisotropic diffusion. *IEEE Transactions On Image Processing*, 11(11), 2002.

[12] N. Greenberg, R. White, J. Thomas Y. Notomi, P. Lysyansky, R. Setser, T. Shiota, Z. Popovic, M. Martin-Miklovic, J. Weaver, S. Oryszak. Measurement of ventricular torsion by two-dimensional ultrasound speckle tracking imaging. *J Am Coll Cardiol.*, 45(12) :2034–2041, 2005.

[13] P. Godard. Les artefacts ultrasonores. artefacts en échographie. *http ://slideplayer.fr/slide/179937/.*

[14] P. Lasaygues, J.-P. Lefebvre, R. Guillermin, V. Kaftandjian, J.-P. Berteau, M. Pithioux, P. Petit. Advanced ultrasonic tomograph of children's bones. *Acoustical Imaging.*, 31 :31–38, 2012.

[15] A. Sarvazyan, O. Rudenko, S. Swanson, J. Fowlkes, S. Emelianov. Shear wave elasticity imaging : a new ultrasonic technology of medical diagnostics. *J Am Coll Cardiol.*, 45(12) :2034–2041, 2005.

[16] J. Bercoff, M. Tanter, M. Fink. Supersonic shear imaging : a new technique for soft tissue elasticity mapping. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 51(4) :396–409, 2004.

[17] B. Arnal, G. Pinton, P. Garapon, M. Pernot, M. Fink, M. Tanter. Global approach for transient shear wave inversion based on the adjoint method : a comprehensive 2D simulation study. *Physics In Medicine ANd Biology*, 58 :6765–6778, 2013.

[18] Graphique de la loi de Moore. `https://commons.wikimedia.org/wiki/File:Loi_de_Moore.png`, 2006.

[19] Quelques lois sur l'augmentation des performances des ordinateurs. `https://zestedesavoir.com/articles/37/quelques-lois-sur-laugmentation-des-performances-des-ordinateurs/`, 2016.

[20] Q. Nguyen, V. Dang, O. Kilic, E. El-Araby. Parallelizing fast multipole method for large-scale electromagnetic problems using GPU clusters. *IEEE Antennas and Wireless Propagation Letters*, 2013.

[21] D. Komatitsch, D. Göddeke, G. Erlebacher, D. Michéa. Modeling the propagation of elastic waves using spectral elements on a cluster of 192 GPUs. *Computer Science – Research and Development*, 25(1-2) :75–82, 2010.

[22] CPU, GPU and MIC hardware characteristics over time. `https://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/`, 2013.

[23] N. Dominguez. Modélisation de la propagation ultrasonore en milieu complexe - application au contrôle non destructif et à la caractérisation de la porosité dans les matériaux composites stratifies. *Thèse de doctorat, Université de Toulouse III - Paul Sabatier*, 2006.

[24] Vladimir V et Schumacher Axel Eschenauer, Hans A et Kobelev. Bubble method for topology and shape optimization of structures. *Structural optimization*, 8(1) :42–51, 1994.

[25] Antoni Sokolowski, Jan et Zochowski. On the topological derivative in shape optimization. *SIAM journal on control and optimization*, 37(4) :1251–1272, 1999.

[26] S. Amstutz, N. Dominguez. Topological sensitivity analysis in the context of ultrasonic non-destructive testing. *Engineering Analysis with Boundary Elements*, 32 :936–947, 2008.

[27] Marc Bonnet. Topological sensitivity for 3d elastodynamic and acoustic inverse scattering in the time domain. *Computer Methods in Applied Mechanics and Engineering*, 195(37) :5239–5254, 2006.

[28] Marc Guzina, BB et Bonnet. Topological derivative for the inverse scattering of elastic waves. *The Quarterly Journal of Mechanics and Applied Mathematics*, 57(2) :161–179, 2004.

[29] Marc Bonnet. Inverse acoustic scattering by small-obstacle expansion of a misfit function. *Inverse Problems*, 24(3) :035022, 2008.

[30] P. Sahuguet. Imagerie ultrasonore de fantômes biologiques par optimisation topologique. *Thèse de doctorat, Université de Toulouse III - Paul Sabatier*, 2012.

[31] S. Rodriguez, P. Sahuguet, V. Gibiat, X. Jacob. Fast topological imaging. *Ultrasonics*, 52 :1010–1018, 2012.

[32] S. Rodriguez, M. Deschamps, M. Castaings, E. Ducasse. Guided wave topological imaging of isotropic plates. *Ultrasonics*, 54(7) :1880–1890, 2014.

[33] J. W.) Lord Rayleigh (Strutt. The theory of sound. *Dover*, 1974.

[34] M. Harris. Optimizing parallel reduction in CUDA. *presentation packaged with CUDA Toolkit, NVIDIA Corporation*, 2007.

[35] J. Claerbout. Toward a unified theory of reflector mapping. *Geophysics*, 36(3) :467–481, 1971.

[36] A. Tarantola. Inversion of seismic reflection data in the acoustic approximation. *Geophysics*, 49(8) :1259–1266, 1984.

[37] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer-Verlag, 2nd edition, 2006.

[38] C. Broyden. A class of methods for solving nonlinear simultaneous equations. *Math. Comp.*, 19 :577–593, 1965.

[39] Stephen G Nash and Jorge Nocedal. A numerical study of the limited memory bfgs method and the truncated-newton method for large scale optimization. *SIAM Journal on Optimization*, 1(3) :358–372, 1991.

[40] Qinya Liu, Jascha Polet, Dimitri Komatitsch, and Jeroen Tromp. Spectral-element moment tensor inversions for earthquakes in Southern California. *bssa*, 94(5) :1748–1761, 2004.

[41] G. Chavent. *Identification of function parameters in partial differential equations, in Identification of parameter distributed systems.* 1974.

[42] R.-E. Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophys. J. Int.*, 167 :495—-503, 2006.

[43] Y. Luo. Seismic imaging and inversion based on spectral-element and adjoint methods. *Thèse de doctorat, Princeton University*, 2012.

[44] J. Tromp, C. Tape, Q. Liu. Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels. *Geophys. J. Int.*, 160 :195–216, 2005.

[45] Y. Luo, R. Modrak, J. Tromp. Strategies in adjoint tomography. *Handbook of Geomathematics*, 2014.

[46] H. Zhang, C. Thurber. Double-difference tomography : The method and its application to the Hayward fault, California. *Bulletin of the Seismological Society of America*, 93(5) :1875–1889, 2003.

[47] E. Bozdağ, J. Trampert, J. Tromp. Misfit functions for full waveform inversion based on instantaneous phase and envelope measurements. *Geophysical Journal International*, 185(2) :845–870, 2011.

[48] V. Monteiller, S. Chevrot, D. Komatitsch, Y. Wang. Three-dimensional full waveform inversion of short-period teleseismic wavefields based upon the SEM–DSM hybrid method. *Geophysical Journal International*, 202 :811—827, 2015.

[49] Youzuo Lin, L. Huang. Acoustic- and elastic-waveform inversion using a modified total-variation regularization scheme. *Geophysical Journal International*, 200(1) :489–502, 2015.

[50] T. Goldstein, S. Osher. The split Bregman method for L1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2) :323–343, 2009.

[51] J. Tromp, D. Komatitsch, Q. Liu. Spectral-element and adjoint methods in seismology. *Communications in Computational Physics*, 3(1) :1–32, 2008.

[52] D. Peter, D. Komatitsch, Y. Luo , R. Martin, N. Le Goff, E. Casarotti, P. Le Loher, F. Magnoni, Q. Liu, C. Blitz, T. Nissen-Meyer, P. Basini, J. Tromp. Forward and adjoint simulations of seismic wave propagation on fully unstructured hexahedral meshes. *Geophys. J. Int.*, 186(2) :721–739, 2011.

[53] M. Chen, J. Tromp. Theoretical and numerical investigations of global and regional seismic wave propagation in weakly anisotropic earth models. *Geophys. J. Int.*, 168(3) :1130–1152, 2007.

[54] C. Morency, J. Tromp. Spectral-element simulations of wave propagation in porous media. *Geophysical Journal International*, 175(1) :301–345, 2008.

[55] R. Martin , D. Komatitsch, H. Barucq. An optimized convolution-perfectly matched layer (C-PML) absorbing technique for 3D seismic wave simulation based on a finite-difference method. *Eos Trans. AGU*, 86(52) :Fall Meet. Suppl., Abstract NG43B–0574, December 2005. www.agu.org/meetings/fm05/waisfm05.html.

[56] D. Komatitsch, J. Tromp. Spectral-element simulations of global seismic wave propagation-II. 3-D models, oceans, rotation, and self-gravitation. *Geophysical Journal International*, 150(1) :303–318, 2002.

[57] A. T. Patera. A spectral element method for fluid dynamics : laminar flow in a channel expansion. *J. Comput. Phys.*, 54 :468—488, 1984.

[58] E. M. Rønquist. Optimal spectral element methods for the unsteady three-dimensional incompressible navier- stokes equations. *Thèse de doctorat, Massachussets Institute of Technology, Massachussets.*, 1988.

[59] D. Komatitsch, S. Tsuboi, J. Tromp. The spectral-element method in seismology. *G. Nolet and A. Levander (Eds.), The Seismic Earth, AGU, Washington DC.*, pages 205–227, 2005.

[60] T.J.R. Hughes. The finite element method, linear static and dynamic finite element analysis. *NJ : Prentice-Hall International.*, 1987.

[61] D. Komatitsch, G. Erlebache, D. Göddeke, D. Michéa. High-order finite-element seismic wave propagation modeling with mpi on a large GPU cluster. *Journal of Computational Physics*, 229(20) :7692–7714, 2010.

[62] F. Pellegrini, J. Roman. Scotch : A software package for static mapping by dual recursive bipartitioning of process and architecture graphs. In *High-Performance Computing and Networking*, pages 493–498. Springer, 1996.

[63] C Hübscher,K. Gohl. Reflection/refraction seismology. *(Encyclopedia of Earth Sciences Series)*, pages 1–15, 2014.

[64] J. Virieux, S. Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6) :WCC1–WCC26, 2009.

[65] L. Sirgue, R. Pratt. Efficient waveform inversion and imaging : A strategy for selecting temporal frequencies. *Geophysics*, 69(1) :231–248, 2004.

[66] V. Kindratenko. Introduction to GPU programming.

Doctorat de l'Université de Toulouse
Délivré par l'Université Toulouse 3 Paul Sabatier
Ecole doctorale MEGeP
Spécialité Dynamique des fluides
Le 7 Octobre 2016
Etienne BACHMANN

Imagerie ultrasonore 2D et 3D sur GPU : application au temps réel et à l'inversion de forme d'onde complète

Si les avancées majeures en imagerie ultrasonore ont longtemps été liées à la qualité de l'instrumentation, l'avènement de l'informatique a incontestablement changé la donne en introduisant des possibilités croissantes de traitement des données pour obtenir une meilleure image. Par ailleurs, les GPUs, composants principaux des cartes graphiques, offrent de par leur architecture des vitesses de calcul bien supérieures aux processeurs, y compris à des fins de calcul scientifique.

Le but de cette thèse a été de tirer parti de ce nouvel outil de calcul, en ciblant deux applications complémentaires. La première est d'autoriser une imagerie en temps réel de meilleure qualité que les autres techniques d'imagerie échographique, en parallélisant le procédé d'imagerie FTIM (Fast Topological IMaging). La seconde est d'introduire l'imagerie quantitative et en particulier la reconstruction de la carte de vitesse du milieu inconnu, en utilisant l'inversion de la forme d'onde complète.

Mots-clés : optimisation, imagerie acoustique, problème inverse, GPU, calcul haute performance

---

2D and 3D ultrasound imaging using GPU : toward real-time and Full Waveform Inversion.

If the most important progresses in ultrasound imaging have been closely linked to the instrumentation's quality, the advent of computing science revolutionized this discipline by introducing growing possibilities in data processing to obtain a better picture. In addition, GPUs, which are the main components of the graphics cards deliver thanks to their architecture a significantly higher processing speed compared with processors, and also for scientific calculation purpose.

The goal of this work is to take the best benefit of this new computing tool, by aiming two complementary applications. The first one is to enable real-time imaging with a better quality than other sonographic imaging techniques, thanks to the parallelization of the FTIM (Fast Tpological IMaging) imaging process. The second one is to introduce quantitative imaging and more particularly reconstructing the wavespeed map of an unknown medium, using Full Waveform Inversion.

Key words : optimization, acoustic imaging, inverse problem, GPU, high performance computing

---

Laboratoire PHASE (Physique de l'Homme Appliquée à Son Environnement)
Université Toulouse 3 Paul Sabatier
118, Route de Narbonne 31062 TOULOUSE CEDEX 9