Accélération du traitement numérique de l'équation de helmholtz par équations intégrales et parallélisation.

Armel de La Bourdonnaye

Introduction

Lorsque l'on cherche à modéliser le comportement d'une onde de fréquence donnée qui se diffracte sur un objet, on a à sa disposition divers types de méthodes. Nous pouvons citer parmi celles-ci l'*optique physique* qui vise à trouver une approximation du courant ou du saut de pression dans la limite des hautes fréquences, la *théorie géométrique de la diffraction* qui utilise un modèle de rayons qui se réfléchissent sur la surface ou bien, après un certain cheminement le long des géodésiques de celle-ci, diffractent un nouveau rayon, et la méthode des *équations intégrales* qui a pour but de calculer "exactement" le courant ou le champ de pression sur toute la surface de l'objet. Nous avons choisi ici d'utiliser les équations intégrales pour plusieurs raisons. D'une part, dans les problèmes de *surface équivalente radar* ou *surface équivalente sonar* on cherche à étudier des objets furtifs. On a alors besoin d'une grande précision et seule cette méthode peut l'apporter dans le cas de corps complexes. D'autre part, elle est une des seules à pouvoir être mise en œuvre facilement quelque soit le corps. En effet, les autres méthodes demandent, soit de suivre une multitude de rayons à travers les diffractions et reflexions multiples sur toute la surface, soit de connaître les courants crées par tous les types de singularités de la surface de l'objet. De plus, elle peut tenir compte de détails de la géométrie de l'ordre de la longueur d'onde, ce qui n'est pas le cas des méthodes asymptotiques. Ceci est important par exemple dans le cas des antennes à cornet (c.f. [Ben84]). Enfin, seules les *équations intégrales* peuvent être couplées de manière assez naturelle avec des éléments finis dans l'intérieur de l'objet pour étudier un corps complexe com-

REMERCIEMENTS

Au terme de ces années, je tiens à remercier toutes les personnes qui m'ont aidé à réaliser ce travail.

Je tiens tout d'abord à exprimer ma gratitude à J.C. Nedelec qui a dirigé cette thèse. Ses conseils et l'attention qu'il m'a apportée m'ont été d'une grande utilité.

Mes remerciements vont aussi à la division *Calcul parallèle* de l'ONERA, où j'ai trouvé chaleur et compétence, et qui m'a hebergé pendant ces dernières années, et plus particulièrement à Pierre Leca son chef et à François-Xavier Roux qui m'ont mis le pied à l'étrier et qui ont toujours montré de l'intérêt pour mon travail.

J'exprime ma reconnaissance au Corps des Ponts et Chaussées qui, en la personne de L.M. Sanche m'a permis de continuer dans la voie de la recherche.

Bernard Larrouturou m'a accueilli avec chaleur et bienveillance au CERMICS, qu'il trouve ici l'expression de ma gratitude.

Philippe Destyunder s'est depuis longtemps interessé a mes travaux, depuis mes débuts à l'ONERA jusqu'au rapport qu'il a fait avec enthousiasme sur cette thèse. Je l'en remercie vivement.

Je remercie P.A. Raviart de l'honneur qu'il me fait en acceptant de présider le jury. Ce fut toujours un plaisir de le rencontrer.

Mikhael Balabane et Gérard Meurant ont accepté la lourde tâche d'être rapporteurs. Qu'ils soient remerciés pour le sérieux avec lequel ils l'ont réalisée.

Je ne peux oublier les joyeux drilles de l'ONERA et du CERMICS avec qui il a toujours été très agréable d'être. Et parmi ceux-ci une mention très spéciale pour mon épouse qui m'a supporté avec courage.

posé de différents types de matériaux. Devant une telle description, on peut se demander pourquoi subsistent d'autres méthodes que celle des *équations intégrales.* En fait cette dernière supporte actuellement une grosse limitation. Elle demande de stocker un volume de données énorme et qui croît comme k^4 si k est proportionnel à la fréquence à laquelle on étudie l'objet. Ceci a comme corollaire qu'il faut un temps C.P.U. important pour résoudre ce problème sur n'importe quel ordinateur, dès que la fréquence devient importante. Le résultat de tout ceci est que l'on est limité dans les fréquences que l'on peut atteindre.

C'est dans ce cadre que se situe notre travail. Nous avons eu en effet pour but, à la fois d'étudier des algorithmes qui puissent accélérer la résolution des systèmes linéaires qui viennent des équations intégrales et de réduire la taille de ces systèmes. Dans cette optique, nous nous sommes restreints à l'équation de Helmholtz scalaire, afin de mieux cerner la difficulté. Nous avons voulu d'autre part, nous situer dans la perspective des calculateurs à venir. Nous avons donc choisi d'étudier plus particulièrement les calculateurs parallèles à mémoire distribuée. En effet, dans l'état actuel de la technique électronique, et même en envisageant une croissance raisonnable de celle-ci, on constate que seule l'agrégation d'un grand nombre de processeurs permettra d'augmenter notablement la puissance des *supercalculateurs.* De plus, pour faire coopérer des processeurs en grand nombre de manière efficace sur le plan du parallélisme, il apparaît nécessaire que la mémoire soit distribuée au moins partiellement entre les processeurs.

Ainsi après un chapitre où nous faisons quelques rappels sur l'équation de Helmholtz, nous commençons par l'étude d'un préconditionneur de méthode itérative. Nous donnons tout d'abord ses propriétés mathématiques puis nous étudions son implémentation sur un ordinateur parallèle. Dans une seconde partie, nous montrons comment réduire le volume de données tout en diminuant le temps de calcul. L'idée principale est d'envelopper le corps dans une surface axisymétrique et de coupler un problème d'équations intégrales sur cette sur-

face avec un problème d'éléments finis classiques entre la surface de l'objet et la surface enveloppante. On utilise alors la forme particulière de cette dernière pour réduire le volume de données. On verra qu'alors, on a ramené le coût de calcul d'un problème en trois dimensions à celui d'un problème axisymétrique. Un dernier chapitre est destiné à éliminer le problème de singularité des éléments finis au voisinage des pôles de l'objet axisymétrique.

Chapitre Premier

Rappels

Nous allons présenter ici quelques rappels sur les formules de représentation intégrale et les équations intégrales pour l'équation de Helmholtz, et sur l'équation des ondes. Ce qui est présenté ici se trouve par exemple dans [BH87], ou, de manière plus détaillée dans [DL85a].

I.1 Formules de représentation intégrale

Soit Ω_i un ouvert borné de $\mathbb{R}^3$, Γ sa frontière et Ω_e l'ouvert extérieur de même frontière (cf. fig I.1).

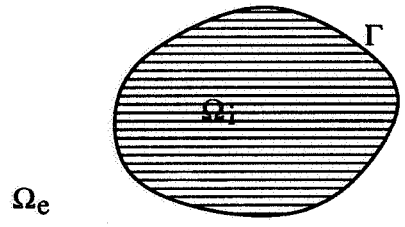


Figure I.1 :

Supposons qu'une fonction u vérifie

$\Delta u + k^2 u = 0$ dans $\Omega_i \cup \Omega_e$

$u \in H^1(\Omega_i) \cap H^1_{loc}(\Omega_e)$

$\frac{\partial u}{\partial r} + iku = o(\frac{1}{r})$pour $r \to \infty$

On appelera u_e la fonction u restreinte à Ω_e et u_i la fonction u restreinte à Ω_i. Nous notons alors $\phi = [u]_{|\Gamma} = u_{i|\Gamma} - u_{e|\Gamma}$ et $p = [\frac{\partial u}{\partial n}]_{|\Gamma} = \frac{\partial u_i}{\partial n} - \frac{\partial u_e}{\partial n}$ où $\frac{\partial}{\partial n}$ est la dérivée par rapport à la normale extérieure à Γ. On sait que l'on a $\phi \in H^{1/2}_\Gamma$ et $p \in H^{-1/2}_\Gamma$. Nous notons encore $G(r) = \frac{e^{ikr}}{4\pi r}$ le noyau de Green de l'équation de Helmholtz en 3 dimensions. Celui-ci est dans $\mathcal{S}'(\mathbb{R}^3)$ qui est l'ensemble des distributions tempérées (i.e. le dual des fonctions de $\mathbb{R}^3$ a décroissance rapide). Alors dans $\mathbb{R}^3$, $\Delta G(r) + k^2 G(r) = \delta_0$ où δ_0 est la masse de Dirac en 0. Nous allons, à l'aide de ce noyau de Green, définir quatre opérateurs de Γ sur Γ.

$$Pour\ p \in H^{-1/2}(\Gamma),\ Sp(x) = \int_\Gamma G(|x-y|)p(y)dy$$

$$Pour\ p \in H^{-1/2}(\Gamma),\ Kp(x) = \int_\Gamma \frac{\partial G}{\partial n_x}(|x-y|)p(y)dy$$

$$Pour\ \phi \in H^{1/2}(\Gamma),\ K'\phi(x) = \int_\Gamma \frac{\partial G}{\partial n_y}(|x-y|)\phi(y)dy$$

$$Pour\ \phi \in H^{1/2}(\Gamma),\ D\phi(x) = \oint_\Gamma \frac{\partial^2}{\partial n_x \partial n_y} G(|x-y|)\phi(y)dy$$

où n_x, n_y sont les normales extérieures à Γ en x et en y (cf. figure I.2), et $\oint$ est

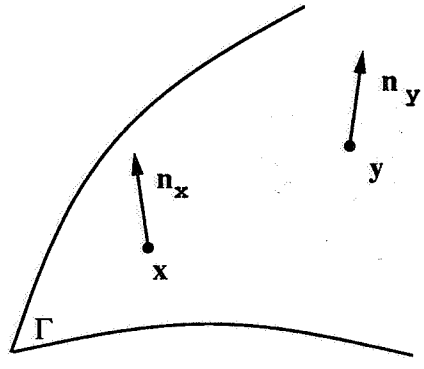


Figure I.2 :

une partie finie au sens de Hadamard de l'intégrale singulière. On sait de plus que S, K, K' sont des opérateurs pseudo-différentiels d'ordre -1 et que D est un opérateur pseudo-différentiel d'ordre +1 (cf. [BH87]).

On peut maintenant donner des formules de représentation intégrale de u. Pour l'établissement de celles-ci, on peut se référer à [Ned78].

$$\begin{aligned}
&\textit{Pour } x \in \Omega_i \cup \Omega_e,\ u(x) = Sp(x) - K'\phi(x) \\
&\textit{Pour } x \in \Gamma,\ \frac{u_i(x) + u_e(x)}{2} = Sp(x) - K'\phi(x).
\end{aligned}$$

Pour la dernière égalité il est essentiel que la surface soit régulière. En effet, il faut qu'en chaque point le cône tangent à Γ sépare l'espace en deux angles solides de même valeur. C'est le cas lorsque ce cône est un plan; ça ne l'est pas

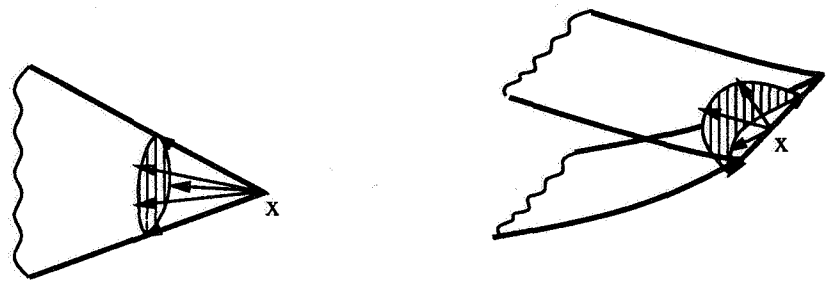


Figure I.3 : Exemples de surfaces singulières

lorsque Γ présente un pli où une pointe (cf. figure I.3). Sur ce point particulier, on peut regarder le traitement qui est fait pour le laplacien en dimension 2 dans [Ned78] et qui est analogue à notre cas.

On peut de même exprimer en fonction de ϕ et p les traces et traces normales de u_i et u_e. Sur Γ,

$$\begin{aligned}
u_e &= Sp - \left(\frac{I}{2} + K'\right)\phi \\
u_i &= Sp + \left(\frac{I}{2} - K'\right)\phi \\
\frac{\partial u_e}{\partial n} &= \left(-\frac{I}{2} + K\right)p - D\phi \\
\frac{\partial u_i}{\partial n} &= \left(\frac{I}{2} + K\right)p - D\phi.
\end{aligned}$$

On va maintenant présenter quelques propriétés des opérateurs S,K,K' et D. On a

$$\begin{cases} SD &= K'^2 - \frac{I}{4} \\ DS &= K^2 - \frac{I}{4} \\ K'S &= SK \\ DK' &= KD \end{cases}$$

Si bien que, si l'on définit les deux opérateurs P^+ et P^- comme suit, alors ils sont deux projecteurs complémentaires de $H^{1/2}(\Gamma) \times H^{-1/2}(\Gamma)$. Ces deux opérateurs sont couramment appelés dans la littérature *Projecteurs de Calderon.*

$$P^{\pm} = \begin{pmatrix} \frac{I}{2} \pm K' & \mp S \\ \pm D & \frac{I}{2} \mp K \end{pmatrix}$$

.

I.2 L'équation de Helmholtz

On va ici rappeler comment l'on dérive l'équation de Helmholtz à partir de celle des ondes et, en particulier, comment on passe de l'hypothèse de corps dur à la condition de Neumann. Ceci permettra d'interpréter de manière plus physique la condition de radiation de Sommerfeld :

$$\frac{\partial u}{\partial r} + iku = o(\frac{1}{r}) \text{pour } r \to \infty$$

que nous avons vue plus haut. Cependant nous ne dériverons pas l'équation des ondes des équations générales de la mécanique des fluides. Pour cela nous renvoyons par exemple à [DL85b]. Nous rappelons simplement que les hypothèses sont celles d'un fluide parfait barotrope (la pression ne dépend que de la densité) et de petits déplacements.

On est toujours dans le cadre géométrique fixé à la figure I.1. On suppose que Ω_i est un obstacle dur. Il est alors facile de constater que $\frac{\partial u}{\partial n} = 0$ sur Γ si u représente le potentiel de la vitesse du gaz. En effet, sur la paroi la vitesse normale du gaz est nulle; comme $v = \nabla u$ où v est la vitesse on a bien la condition de Neumann sur u. Notons que pour interpréter une condition homogène de

Dirichlet, il faut changer l'interprétation de u. En effet, u représente alors la variation de la pression et la condition de Dirichlet signifie que la variation de la pression est nulle sur l'obstacle, ce qui est la caractéristique d'un corps mou. Si c est la vitesse du son dans Ω_e, que l'on suppose constante et indépendante de la direction de propagation, alors l'équation des ondes s'écrit :

$$\Delta u - \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} = 0 \text{ pour } t > 0 \text{ et } x \in \Omega_e.$$

On recherche maintenant des solutions périodiques en temps. On écrit alors $u(x,t) = U(x)e^{-i\omega t}$ où ω, nombre positif, est la pulsation à laquelle on veut observer le phénomène. U vérifie alors l'équation de Helmholtz $\Delta U + k^2 U = 0$ avec $k^2 = \frac{\omega^2}{c^2}$. Comme le problème est ici de connaître l'onde diffractée par l'objet en connaissant l'onde incidente, on décompose en fait U en une onde incidente U_{inc} et une onde diffractée U_{diff}. On sait alors que U_{diff} vérifie à son tour l'équation de Helmholtz avec comme condition sur Γ, $\frac{\partial U_{diff}}{\partial n} = -\frac{\partial U_{inc}}{\partial n}$. Si on suppose U_{diff} dans $H^1_{loc}(\Omega_e)$, on peut, pour assurer l'unicité de la solution, rajouter une condition supplémentaire portant sur le comportement à l'infini de la solution; c'est la condition de radiation d'onde sortante de Sommerfeld.

$$\lim_{r\to\infty} r(\frac{\partial U_{diff}}{\partial r} - ikU_{diff}) = 0.$$

Sans rentrer dans les détails de la solution de l'équation des ondes regardons ce que signifie cette condition de radiation. L'équation des ondes dans $\mathbb{R}^3$ admet deux solutions élémentaires indépendantes qui sont les suivantes.

$$E^+(x,t) = \frac{1}{4\pi rc}\delta_0(ct + |x|)$$

$$E^-(x,t) = \frac{1}{4\pi rc}\delta_0(ct - |x|)$$

La condition d'onde sortante consiste alors à ne sélectionner que E^-. En effet, $e^{-i\omega t}\frac{\partial U}{\partial r} - ikU = \frac{\partial u}{\partial r} + c\frac{\partial u}{\partial t}$. L'appellation d'onde sortante devient maintenant naturelle. En effet le support de $E^-(.,t)$ est la sphère de rayon ct. Il s'écarte donc de l'origine au fur et à mesure que le temps croît avec la vitesse c.

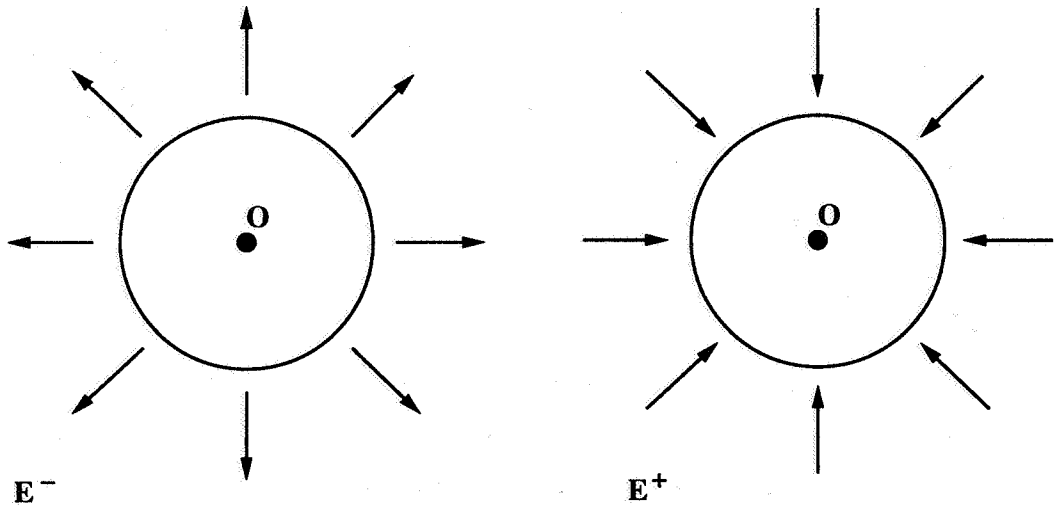


Figure I.4 : Supports de E^- et E^+.

I.3 Equations intégrales et formulations variationnelles

Nous allons montrer maintenant comment l'on utilise les formules de représentation intégrale pour résoudre des problèmes aux limites pour l'équation de Helmholtz. Nous étudierons principalement le problème de diffraction extérieur avec conditions au bord de Neumann. Il se formule ainsi. Trouver u dans $H^1_{loc}(\Omega_e)$ vérifiant

$$\Delta u + k^2 u = 0 \; dans \; \Omega_e$$
$$\frac{\partial u}{\partial n} = g \; sur \; \Gamma$$
$$\lim_{r\to\infty} r(\frac{\partial u}{\partial r} - iku) = 0$$

Rappelons que $H^1_{loc}(\Omega_e)$ est l'espace des fonctions de Ω_e qui sont dans $H^1(K)$ pour tout compact K de Ω_e.

Afin d'utiliser les formules de représentation intégrale que l'on a rappelées, et qui utilisent les sauts à travers Γ de la fonction représentée, il faut d'abord prolonger u à Ω_i. Pour cela, nous allons choisir un problème intérieur que vérifiera u dans Ω_i et qui ainsi sera couplé à notre problème. Ceci peut être fait de

plusieurs manières qui ont chacunes leurs avantages. Une première méthode consiste à chercher u dans $H^1(\Omega)$ vérifiant l'équation de Helmholtz et la condition de Dirichlet $u_i = u_e$ sur Γ. Dans ce cas, avec les notations de la page 8, u vérifie $\phi = 0$ et donc $g = (-\frac{I}{2} + K)p$. On a ainsi une équation en p qui est du type Fredholm. On fait ici deux remarques. D'une part il n'est pas toujours possible de demander la continuité de u à travers Γ. En effet, si celle-ci est une surface ouverte, comme une plaque par exemple on a alors imposé à la fois ϕ nul par le choix du problème intérieur et p nul à cause de la condition aux limites qui s'applique alors aux deux côtés de la plaque. Dans ce cas, u est forcément nul. D'autre part, on s'aperçoit que l'on transforme un problème qui a toujours une solution unique (le problème extérieur) en un problème qui peut avoir des valeurs propres nulles. En effet, il est connu que les problèmes intérieurs (de Neumann comme de Dirichlet) peuvent avoir des fréquences de résonance (i.e. qui génèrent des valeurs propres nulles). Il suffit en effet que $-k^2$ soit une valeur propre du Laplacien avec la condition aux limites correspondante.

Une autre manière de prolonger le problème est de chercher u dans $H^1(\Omega)$ vérifiant l'équation de Helmholtz et la condition de Neumann $\frac{\partial u_i}{\partial n} = g$ sur Γ. Dans ce cas, p est nul et on a comme équation intégrale $g = -D\phi$. Ici aussi on a le problème des éventuelles valeurs propres nulles du problème intérieur. Cependant, lorsque g est l'opposé de la trace normale d'une onde incidente dont la source ne se trouve pas dans Ω_i, on sait que l'on a toujours une solution même si elle n'est pas unique. En effet, on sait alors que $-U_{inc}$ est analytique dans Ω_i et vérifie la bonne condition sur Γ. En d'autres termes, supposer que l'onde totale est nulle dans Ω_i est toujours une hypothèse autorisée pour un problème de diffraction.

Venons maintenant aux formulations variationnelles. Dans le cas du prolongement par un problème de Dirichlet ceci s'écrit simplement :
Trouver p dans $H^{-1/2}(\Gamma), \forall\, \psi \in H^{1/2}(\Gamma), < g, \psi > = < (-\frac{I}{2} + K)p, \psi >$.
Nous avons noté $<,>$ le produit de dualité entre H^s et H^{-s} sur Γ. Remarquons

que l'on n'a pas ici une formulation symétrique ou hermitienne.
Dans le cas du prolongement par un problème de Neumann les choses sont un peu plus compliquées. On pourrait certes essayer de procéder exactement comme ci-dessus, mais l'opérateur D est hypersingulier et sur le plan numérique il se pose alors de sérieux problèmes. Au lieu de cela, on utilise une formulation différente due à Hamdi (cf. [Ham81]).
On cherche ϕ dans $H^{1/2}(\Gamma), \forall \psi \in H^{1/2}(\Gamma)$

$$< g, \psi >=< S rot_\Gamma \phi . rot_\Gamma \psi > -k^2 < S\phi n . \psi n >$$

où $< a.b >= \int_\Gamma a(x).b(x)dx$ si a et b sont des champs de vecteurs sur Γ. ϕn est le vecteur produit de la fonction ϕ et du vecteur n, normale extérieure à Γ. rot_Γ est le rotationnel surfacique, il est défini par $rot_\Gamma \phi(x) = n(x) \wedge \nabla_\Gamma \phi(x)$. Une dérivation différente de cette formulation variationnelle est donnée dans [Ben84]. L'idée principale est de considérer $v = \nabla u$, et de remarquer que v vérifie encore l'équation de Helmholtz et la condition de radiation. On utilise alors une formule de représentation intégrale adaptée aux vecteurs. On remarque qu'ici la formulation est symétrique mais pas hermitienne.

I.4 Champ lointain et amplitude limite

Dans divers problèmes de propagation d'onde on s'intéresse uniquement au champ loin de l'obstacle diffractant et il est alors courant de supposer que l'on est à une distance infinie de celui-ci. On introduit alors l'amplitude limite dans une direction. Si ξ est un élément de la sphère S^2, alors l'amplitude $A(\xi)$ d'une onde acoustique diffractée u sur un obstacle de surface Γ, est classiquement définie par $A(\xi) = \frac{1}{4\pi} \int_\Gamma e^{-ik\xi.y} (\frac{\partial u_e}{\partial n}(y) + ik\xi.n_\Gamma u_e(y)) dy$. Ceci dérive directement des formules de représentation intégrale à l'extérieur de l'obstacle. On énonce ici quelques propositions que l'on peut retrouver dans [BH87].

Proposition 1 *Si* $A(\xi) = 0, \forall \xi \in S^2$ *alors* u *et* $\frac{\partial u}{\partial n}$ *sont nuls sur* Γ.

Proposition 2 $\int_{S^2} |A(\xi)|^2 d\xi = \mathcal{I}m \left(\frac{1}{k} \int_\Gamma \frac{\partial u_e}{\partial n} . \bar{u}_e \right)$

Ces propositions montrent que la norme L_2 de l'amplitude limite est une norme sur l'espace des fonctions vérifiant l'équation de Helmholtz dans Ω_e et la condition de radiation.

Première Partie

Préconditionnement

Chapitre Premier

Etude théorique

Introduction

Nous nous intéressons ici au préconditionnement d'une méthode numérique qui a pour but de résoudre un problème de diffraction en domaine extérieur avec une condition de corps dur par équations intégrales. La méthode numérique utilisée est celle des "Résidus conjugués généralisés" autrement notée "Orthomin" dans la littérature (cf. [Jol88]). L'objet de ce premier chapitre est de tenter de comprendre l'action du préconditionneur que nous avons choisi en regardant comment il modifie le spectre de l'opérateur. Nous allons commencer par regarder le cas d'une surface Γ plane afin de simplifier les calculs. Nous passerons ensuite au cas d'une surface quelconque compacte dans R^3, puis nous regarderons de manière plus précise le cas de la sphère afin de comprendre l'action du préconditonneur. Le résultat principal sera que le haut du spectre de l'opérateur préconditionné sera beaucoup moins dense que celui de l'opérateur seul.

Nous utilisons ici la formulation variationnelle de Hamdi rappelée au chapitre précédent. Plus précisement, soit, comme dans le chapitre précédent, Ω_i un ouvert relativement compact de $\mathbb{R}^3$, Γ son bord que l'on supposera régulier et $\Omega_e = \mathbb{R}^3 - Adh(\Omega_i)$. On cherche u dans $H^1(\Omega_i) \cap H^1_{loc}(\Omega_e)$ vérifiant

$$\Delta u + k^2 u = 0 \text{ dans } \Omega_i \cap \Omega_e$$

$$\left[\frac{\partial u}{\partial n}\right]_{|\Gamma} = 0$$
$$\frac{\partial u}{\partial n}_{|\Gamma} = g$$

où g est dans $H^{-1/2}(\Gamma)$.

Nous prenons donc comme inconnue $\phi = [u]_{|\Gamma}$ le saut de u à travers Γ, ϕ est alors dans $H^{1/2}(\Gamma)$. Ceci nous conduit alors à la formulation variationnelle suivante.

$\forall \psi \in H^{1/2}(\Gamma)$

$$\int_{\Gamma\times\Gamma} \frac{e^{ik|x-y|}}{4\pi|x-y|}(rot_\Gamma\psi(x).rot_\Gamma\phi(y) - k^2 n_x.n_y\psi(x)\phi(y))dxdy = \int_\Gamma \psi(x)g(x)dx$$

On pourra utilement se référer sur ce sujet à [Ham81] ou [Ned82] .

Comme on l'a déjà remarqué dans le chapitre de rappels, cette formulation variationnelle est symétrique mais pas hermitienne. Sa discrétisation par une méthode d'éléments finis (cf. [Ben84]) conduit alors à une matrice pleine complexe symétrique et donc non hermitienne. La matrice est pleine car, du fait de l'intégration sur $\Gamma \times \Gamma$, chaque fonction de base interagit avec toutes les autres. Si on veut utiliser dans ce cas une méthode itérative du type multigradient, alors il est doublement intéressant de préconditionner si l'on obtient une réduction effective du nombre d'itérations. En effet, d'une part, réduire le nombre d'itérations permet d'espérer améliorer le temps de calcul. D'autre part, dans le cadre de ces méthodes, on est obligé de stocker une direction de descente supplémentaire à chaque itération afin d'assurer l'orthogonalité ou la conjugaison de ces directions. Il est alors clair que diminuer le nombre d'itérations réduit aussi la place prise en mémoire pour la phase de résolution.

Pour définir le préconditionneur on va d'abord prendre $\chi(r)$ une fonction "cut-off" au voisinage de 0, c'est à dire une fonction $C^\infty(\mathbb{R}^3, \mathbb{R})$ à support

compact valant 1 au voisinage de 0, positive, et ne dépendant que du rayon r si on l'exprime en fonction des coordonnées sphériques r, θ, ϕ centrées en 0. Puis on note $K_1(r) = \frac{e^{ikr}}{4\pi r}$ et $K_2(r) = \frac{1}{k} K_1(r)\chi(kr)$. Le noyau K_2 est alors une version locale de K_1. Il consiste à ne s'occuper que des interactions qui ont lieu entre points distants d'une longueur d'onde au plus (à une constante multiplicative près). Le terme en $\frac{1}{k}$ n'intervient que pour la normalisation. Alors pour préconditionner on va remplacer dans la formulation variationnelle le noyau $K_1(|x-y|)$ par $K_2(|x-y|)$.

Dans ce qui suit on va d'abord regarder comment se comporte le préconditionneur lorsque k, le nombre d'onde, tend vers l'infini, puis on regardera les qualités numériques de cette méthode, tant du point de vue de l'amélioration du nombre d'itérations, que de la parallélisation effective. Les implémentations montrées ici ont été réalisées sur un processeur du Cray-2 du CCVR et sur l'hypercube iPSC2 de l'ONERA.

I.1 Etude du préconditionneur pour une surface plane.

On va commencer par regarder le comportement du préconditionneur lorsque la surface Γ est un plan. L'intérêt de la surface plane est d'avoir à sa disposition le puissant outil d'analyse qu'est la transformée de Fourier. Celui-ci est particulièrement bien adapté dans ce cas puisque les opérateurs s'expriment sous forme de convolution. Par ailleurs, dans le cas d'un plan, le rotationnel surfacique d'une fonction se réduit à son gradient surfacique tourné de $\frac{\pi}{2}$ dans le sens donné par la normale au plan (cf. figure I.1). Il suffit pour cela de regarder comment est défini le rotationnel surfacique (cf. p. 14). Cependant cette étude n'est là que pour nous donner une idée de ce qui se passe dans le cas de surfaces plus complexes. Il faut de plus faire attention à ne pas faire de généralisations trop hâtives. En effet, seuls les résultats qui dépendent de propriétés locales sont susceptibles d'être généralisés.

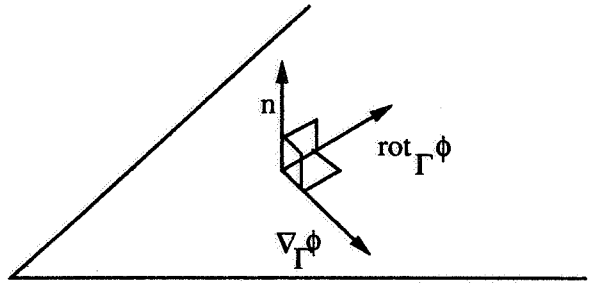


Figure I.1 :

Nous noterons A l'opérateur venant du noyau K_1 et B celui venant de K_2. Ainsi le problème d'équations intégrales s'écrit :
trouver $\phi \in H^{1/2}(\Gamma)$ telle que $< A\phi, \psi >=< g, \psi >$, $\forall \psi \in H^{1/2}(\Gamma)$, où $< , >$ est le crochet de dualité sur Γ. B sera le préconditionneur.

On va d'abord montrer que $k^2\hat{K}_2(\xi)$, qui est la transformée de Fourier de $k^2K_2(r)$, tend vers une constante C_χ qui ne dépend que du cut-off χ lorsque k tend vers l'infini, puis on montrera que $< B\phi, \psi >$ tend vers C_χ. $< \Delta_\Gamma\phi + k^2\phi, \psi >$ où Δ_Γ est l'opérateur de Laplace-Beltrami sur la surface Γ. Dans le cas du plan, cet opérateur est le Laplacien. Nous allons commencer par montrer une série de lemmes.

Lemme 1 *$k^2\hat{K}_2(\xi)$ ne dépend que de $\frac{|\xi|}{k}$ et est continu en fonction de ξ.*

Preuve :
On a

$$\begin{aligned} k^2\hat{K}_2(\xi) &= \frac{1}{2\pi}\int_{R^2} e^{i\xi.x}\frac{e^{ik|x|}}{4\pi|x|}k\chi(k|x|)dx \\ &= \frac{1}{2\pi}\int_0^{2\pi}\int_0^{\infty} e^{i|\xi|r\cos\theta}\frac{e^{ikr}}{4\pi r}k\chi(kr)rdrd\theta \\ &= \int_0^{2\pi}\int_0^{\infty} e^{i\frac{|\xi|}{k}\rho\cos\theta}\frac{e^{i\rho}}{8\pi^2}\chi(\rho)d\rho d\theta. \end{aligned}$$

On passe de la première ligne à la seconde en passant en coordonnées polaires et de la seconde à la troisième en faisant le changement de variables $(\rho, \theta) = (kr, \theta)$.

Sur cette formule on voit clairement que $k^2\hat{K}_2(\xi)$ est continu en ξ, il est même C^∞ en dehors de 0.

Lemme 2 *$k^2\hat{K}_2(\xi)$ est borné uniformément en k.*

Preuve : D'après le lemme 1 on a

$$k^2\hat{K}_2(\xi) = \int_0^{2\pi}\int_0^{\infty} e^{i\frac{|\xi|}{k}\rho\cos\theta}\frac{e^{i\rho}}{8\pi^2}\chi(\rho)d\rho d\theta.$$

Donc

$$\left|k^2\hat{K}_2(\xi)\right| < \frac{1}{8\pi^2}\int_0^{2\pi}\int_0^{\infty}\chi(\rho)d\rho d\theta.$$

Soit en intégrant en θ,

$$\left|k^2\hat{K}_2(\xi)\right| < \frac{1}{4\pi}\int_0^{\infty}\chi(\rho)d\rho.$$

Ceci achève la preuve du lemme.

Lemme 3 $\exists C_\chi \in R^+, \forall \xi \in R^3 \lim_{k\to\infty} k^2\hat{K}_2(\xi) = C_\chi.$

Néanmoins, la convergence n'est pas uniforme en ξ comme nous le verrons par la suite.

Preuve :

On va employer le théorème de convergence dominée de Lebesgue. On a déjà montré que $k^2\hat{K}_2(\xi)$ est uniformément borné en k. Il reste à montrer que l'intégrande a une limite et voir laquelle. On a d'après le lemme 1

$$k^2\hat{K}_2(\xi) = \int_0^{2\pi}\int_0^{\infty} e^{i\frac{|\xi|}{k}\rho\cos\theta}\frac{e^{i\rho}}{8\pi^2}\chi(\rho)d\rho d\theta.$$

On fait maintenant tendre k vers l'infini dans l'intégrande. Ce dernier tend alors vers $e^{i\rho}\frac{\chi(\rho)}{8\pi^2}$. Le théorème de convergence dominée permet alors de conclure que $\forall \xi \in \mathbb{R}^2$, $\lim_{k\to\infty} k^2\hat{K}_2(\xi) = \frac{1}{4\pi}\int_0^{\infty} e^{i\rho}\chi(\rho)d\rho$. Ceci achève la preuve du lemme.

Il est d'autre part facile de voir que la convergence n'est pas uniforme en ξ. En effet, comme $k^2\hat{K}_2(\xi)$ ne dépend que de $\frac{|\xi|}{k}$, la convergence uniforme implique que $k^2\hat{K}_2(\xi)$ est constante puisque $k^2\hat{K}_2(\xi)$ est continu en ξ, donc en $\frac{|\xi|}{k}$, et converge donc vers $k^2\hat{K}_2(0)$ lorsque k tend vers l'infini. Or $k^2\hat{K}_2(\xi)$ n'est pas constante. En effet, dans le cas contraire, K_2 serait une masse de Dirac.

Néanmoins on a le résultat suivant.

Lemme 4 $\forall \xi \in I\!R^2, \exists C' > 0, |k^2\hat{K}_2(\xi) - C_\chi| < C' min(1, \frac{|\xi|}{k})$

Preuve :

$$k^2\hat{K}_2(\xi) - C_\chi = \frac{1}{2\pi}\int_0^{2\pi}\int_0^{\infty}\chi(\rho)\frac{e^{i\rho}}{4\pi}(e^{i\rho\frac{|\xi|}{k}\cos\theta} - 1)d\rho d\theta.$$

Comme $|e^{i\rho\frac{|\xi|}{k}\cos\theta} - 1| < 2\min(1, \rho\frac{|\xi|}{k})$,

on a $|k^2\hat{K}_2(\xi) - C_\chi| < \min(1, \frac{|\xi|}{k})\frac{1}{\pi}\int_0^{2\pi}\int_0^{\infty}\frac{\chi(\rho)}{4\pi}\rho d\rho d\theta$. Ceci termine la preuve du lemme.

Notons donc $C_\chi = \frac{1}{4\pi}\int_0^{\infty} e^{i\rho}\chi(\rho)d\rho$ la constante trouvée au lemme 3. Nous allons étudier maintenant $B + C_\chi(Id + \frac{\Delta_\Gamma}{k^2})$.

Théorème 1 $\forall\, s, \frac{1}{2} \geq s > 0, \exists \alpha > 0, \forall \phi, \psi \in H^{s+1}(R^2)$,

$$\left| < B\phi, \psi > + C_\chi < Id + \frac{\Delta_\Gamma}{k^2}\phi, \psi > \right| < \alpha\left(\frac{||\phi||_{H^s}||\psi||_{H^s}}{k^{2s}} + \frac{||\phi||_{H^{s+1}}||\psi||_{H^{s+1}}}{k^{2s+2}}\right).$$

Nous prendrons ici comme norme de H^s la norme suivante.

$$||\phi||^2_{H^s} = \int_\Gamma (1 + |\xi|^2)^s \hat{\phi}(\xi)^2 d\xi$$

Avant de montrer le théorème nous allons montrer deux lemmes qui nous permettront de conclure. Le premier est un lemme technique.

Lemme 5 $\forall s, \frac{1}{2} \geq s \geq 0, \forall k > 0, \max(\frac{\min(1, \frac{|\xi|}{k})}{(1+\xi^2)^s}) < \frac{1}{k^{2s}}$

Preuve :

Pour montrer ce lemme, nous allons commencer par supposer que $|\xi| > k$. Alors, on a facilement que $\frac{1}{(1+|\xi|^2)^s} < \frac{1}{(1+k^2)^s}$.

Pour $|\xi| \leq k$, on va dériver $\frac{\frac{|\xi|}{k}}{(1+|\xi|^2)^s}$.

$$\frac{d}{d|\xi|}\left(\frac{\frac{|\xi|}{k}}{(1+|\xi|^2)^s}\right) = \frac{1}{k}\frac{(1+|\xi|^2)-2s|\xi|^2}{(1+|\xi|^2)^{s+1}}.$$

Le signe de cette dérivée est donc celui de $1+|\xi|^2(1-2s)$. On voit donc, que pour $s \leq \frac{1}{2}$, la fraction est croissante et atteint alors son maximum en k. Ainsi,

pour $|\xi| \leq k$, $\frac{\frac{|\xi|}{k}}{(1+|\xi|^2)^s} \leq \frac{1}{(1+k^2)^s} \leq \frac{1}{k^{2s}}$.

Lemme 6 $\forall\, s,\ \frac{1}{2} \geq s > 0, \exists \alpha > 0, \forall \phi, \psi \in H^s(R^2)$,
$\left| < k^2 K_2 * \phi, \psi > - C_\chi < \phi, \psi > \right| < \alpha \frac{||\phi||_{H^s}||\psi||_{H^s}}{k^{2s}}$.

Ici, $u * v$ est le produit de convolution de u et v. Preuve :

Nous avons $< k^2 K_2 * \phi, \psi > - C_\chi < \phi, \psi > = < (k^2 \hat{K}_2(\xi) - C_\chi)\hat{\phi}, \hat{\psi} >$.
Donc d'après le lemme 4,

$$\left| < k^2 K_2 * \phi, \psi > - C_\chi < \phi, \psi > \right| < C' \int_{R^2} \frac{min(1, \frac{|\xi|}{k})}{(1+\xi^2)^s}(1+\xi^2)^s |\hat{\phi}(\xi)\hat{\psi}(\xi)| d\xi.$$

Or $\max(\frac{\min(1, \frac{|\xi|}{k})}{(1+\xi^2)^s}) < \frac{1}{k^{2s}}$. Ainsi nous avons

$$\begin{aligned} \left| < k^2 K_2 * \phi, \psi > - C_\chi < \phi, \psi > \right| &< \frac{C'}{k^{2s}} \int_{R^2} (1+\xi^2)^s |\hat{\phi}(\xi)\hat{\psi}(\xi)| d\xi \\ &\leq \alpha \frac{||\phi||_{H^s}||\psi||_{H^s}}{k^{2s}} \end{aligned}$$

Ceci achève la preuve du lemme. Revenons à la preuve du théorème :
$< B\phi, \psi > = - < k^2 K_2 * \phi, \psi > + < K_2 * \nabla\phi, \nabla\psi >$, alors

$$\left| < B\phi, \psi > + C_\chi < Id + \frac{\Delta_\Gamma}{k^2}\phi, \psi > \right| \leq \left| < k^2 K_2 * \phi, \psi > - C_\chi < \phi, \psi > \right|$$

$$+ \frac{1}{k^2} \left| < k^2 K_2 * \nabla\phi, \nabla\psi > - C_\chi < \nabla\phi, \nabla\psi > \right|$$

Le lemme précédent appliqué à chacun des deux termes de droite de l'inégalité permet alors de conclure la preuve du théorème. Avant de poursuivre faisons une remarque. Pour $s > 1/2$, dans le lemme 5 on a le même genre de résultat mais au lieu d'un exposant 2s on stationne à l'exposant 1 en k. Ceci signifie qu'il n'est pas intéressant d'aller au delà de s=1/2 car après, la vitesse de convergence de B ne s'améliore plus et alors le résultat est dégradé puisqu'il a lieu dans une norme plus fine.

I.1.1 Application aux espaces d'éléments finis

On va maintenant regarder comment notre résultat s'applique dans les espaces d'éléments finis. On choisit un paramètre ϵ, $0 \leq \epsilon < \frac{1}{2}$. On appelle $H_h^{3/2-\epsilon}$ un espace d'approximation de $H^{3/2-\epsilon}$, où h est un petit paramètre qui caractérise la finesse de l'approximation. Lorsque l'on considère le cas d'approximations de la surface Γ par des éléments finis triangulaires, h représente la plus grande longueur d'arête du maillage. On va pouvoir, dans ce cas, affiner le résultat du théorème précédent. En effet, dans $H_h^{3/2-\epsilon}$ qui est un espace de dimension finie, toutes les normes sont équivalentes. De plus, avec la définition de h que l'on a donnée, on a même mieux. En effet, on sait qu'il existe a et b, deux constantes positives telles que $a.||\ ||_{H^{1/2-\epsilon}} < ||\ ||_{H^{3/2-\epsilon}} < \frac{b}{h}.||\ ||_{H^{1/2-\epsilon}}$. Pour ceci on pourra se référer au théorème 17.2 de [Cia91] dont on supposera que les hypothèses sont réalisées. Pour les problèmes de propagation d'ondes en régime harmonique, il est classique de prendre h proportionnel à 1/k. Alors, dans ce cadre, le résultat du théorème I.1 devient

$$\left| < B\phi, \psi > + C_\chi < Id + \frac{\Delta_\Gamma}{k^2}\phi, \psi > \right| < \alpha \left(\frac{||\phi||_{H^{1/2-\epsilon}} ||\psi||_{H^{1/2-\epsilon}}}{k^{1-2\epsilon}} \right).$$

Dans le cas d'approximations par des éléments finis P^1, on ne peut pas aller jusqu'à ϵ nul. En effet, ces éléments, qui ne sont pas C^1, ne sont par conséquent pas dans $H^{3/2}$ et le raisonnement qui vient d'être fait ne s'applique plus. On a ainsi obtenu la convergence de B vers $-C_\chi(Id + \frac{\Delta_\Gamma}{k^2})$ en contrôlant la norme $H^{1/2-\epsilon}$ de ϕ et ψ pour $0 < \epsilon < 1/2$.

I.2 Cas d'une surface courbée

Dans ce cas les techniques de la partie précédente sont inutilisables. Cependant on va obtenir un résultat similaire. Ceci est "moral" puisque le support du noyau tend vers 0 et donc au premier ordre on peut encore assimiler la surface et son plan tangent. Comme la surface est compacte la convergence sera "uniforme". On va commencer par quelques lemmes. Nous utilisons les mêmes notations que dans la partie précédente. On va commencer par quelques rappels sur la géométrie riemanienne des surfaces.

I.2.1 Rappels de géométrie

Nous nous restreindrons ici aux surfaces de $\mathbb{R}^3$ compactes et sans bord. On sait qu'elles sont alors orientables. Commençons par rappeler la définition d'une variété.

Définition 1 *M est une variété de dimension n si $\forall x \in M, \exists U$, un voisinage de x dans M et θ un difféomorphisme de U dans $\mathbb{R}^n$. La classe de continuité minimale des difféomorphismes caractérise la classe de la variété.*
Un couple U, θ est appelé une carte de M.
Une collection de cartes recouvrant M est un atlas.

Notons que, pour une variété compacte, il existe des atlas qui ont un nombre fini de cartes et même, de tout atlas un peut extraire un tel atlas. De plus, on peut se contenter dans ce cas de difféomorphismes propres (qui envoient un

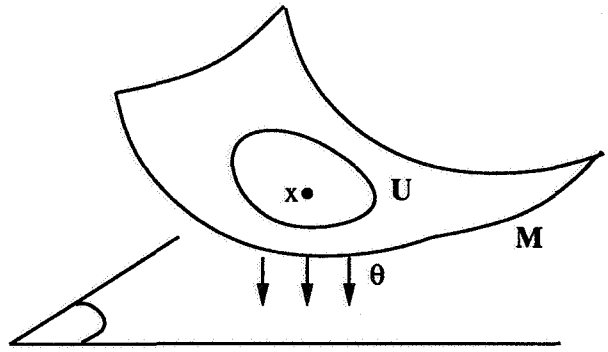


Figure I.2 : Une carte

compact sur un relativement compact) ainsi que leurs réciproques. On peut pour toute variété au moins C^1 exhiber la notion de plan tangent. Il est alors clair que l'on peut remplacer un difféomorphisme de U dans un voisinage de x dans $\mathbb{R}^n$, par un difféomorphisme de U dans T_xM le plan tangent à M en x et qui envoie x sur lui même.

Nous allons maintenant préciser un système de coordonnées locales. Nous revenons pour plus de simplicité dans l'écriture au cas d'une variété de dimension 2 dans $\mathbb{R}^3$.

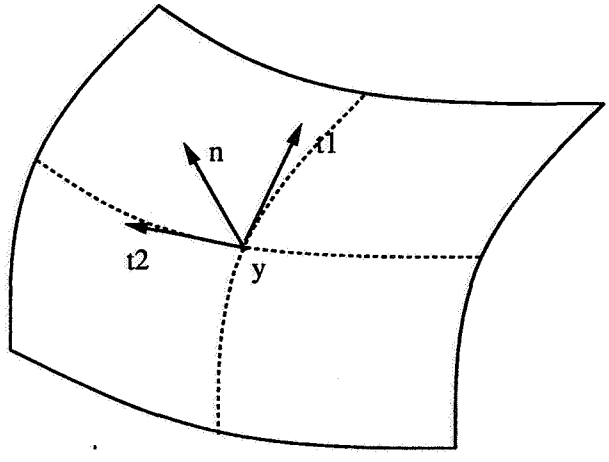


Figure I.3 : Paramétrisation par le plan tangent

Soit x un point de Γ. Appelons t_1, t_2 deux vecteurs formant une base orthonormée de T_xM et τ_1, τ_2 le système de coordonnées associé à cette base en prenant x comme origine. On note encore n la normale à M, choisie de sorte que (t_1, t_2, n) est un trièdre direct et z la coordonnée dans la troisième dimension. Alors $\forall y \in U$, $\exists\ (\tau_1, \tau_2)$, $y = \theta^{-1}(\tau_1, \tau_2)$. Ceci constitue une paramétrisation locale de la surface. Si on a choisi θ comme étant la projection orthogonale de Γ sur T_xM et que l'on suppose que la surface est C^3, on peut alors écrire $y = x + \tau_1 t_1 + \tau_2 t_2 + L(\tau, \tau)n + \mathcal{O}(|\tau|^3)$ comme paramétrisation où L est une forme bilinéaire symétrique, τ est le vecteur du plan tangent de coordonnées (τ_1, τ_2) et $|\tau|^2 = {\tau_2}^2 + {\tau_1}^2$. Il n'y a pas de terme du second ordre sur le plan tangent car θ est une projection orthogonale. On peut alors donner comme équation locale de M : $z = L(\tau, \tau)n + o(|\tau|^2)$. On peut choisir t_1 et t_2 comme étant deux vecteurs propres de L. L s'écrit alors classiquement $L(\tau, \tau) = \frac{{\tau_1}^2}{2R_1} + \frac{{\tau_2}^2}{2R_2}$. R_1 et R_2 sont les deux rayons principaux de courbure de M en x.

I.2.2 Analyse du préconditionneur

Nous garderons ici les notations précédentes pour la géométrie. Nous supposerons que la surface Γ est C^3. Nous allons commencer par quelques lemmes.

Lemme 7 *Supposons donc Γ au moins C^3, alors*

$(i)\ |x - y| = |\tau| + \mathcal{O}(|\tau|^3)$

$(ii)\ \frac{Dx}{D\tau} = 1 + \mathcal{O}(|\tau|^2)$

où $\frac{Dx}{D\tau}$ est le jacobien de la transformation $\tau \longrightarrow x$

Preuve:

On utilise donc le paramétrage de la surface au voisinage de y par le plan tangent donné plus haut. Nous avons $x - y = (\tau_1, \tau_2, \frac{\tau_1^2}{2R_1} + \frac{\tau_2^2}{2R_2}) + \mathcal{O}(|\tau|^3)$. Alors

$|x-y|^2 = |\tau|^2 + \mathcal{O}(|\tau|^4)$. Ainsi $|x-y| = |\tau| + \mathcal{O}(|\tau|^3)$. Ceci montre *(i)*. Pour *(ii)* il nous faut calculer les deux vecteurs tangents t1 et t2 en un point paramétré par τ engendrés par les vecteurs (0,1) et (1,0) dans le plan tangent en y à Γ. $t1 = (1, 0, \frac{\tau_1}{R_1}) + \mathcal{O}(|\tau|2)$ et $t2 = (0, 1, \frac{\tau_2}{R_2}) + \mathcal{O}(|\tau|^2)$. Alors le jacobien $\frac{Dx}{D\tau}$ vaut $|t1 \wedge t2|$, soit $|(-\frac{\tau_1}{R_1}, -\frac{\tau_2}{R_2}, 1)|$. On a donc bien $\frac{Dx}{D\tau} = 1 + \mathcal{O}(|\tau|^2)$, ce qui achève la preuve du lemme.

Lemme 8 $\int_\Gamma k^2 K_2(|x-y|)dx = C_\chi(1 + \mathcal{O}(\frac{1}{k^2}))$ *où* C_χ *est la constante de la partie précédente; de plus la convergence est uniforme en y.*

Preuve :

D'après le lemme précédent, au voisinage de y, $x(\tau) = y + \tau L(\tau, \tau)n + \mathcal{O}(|\tau|^3)$ où τ est un vecteur du plan tangent en y à Γ. Le $\mathcal{O}(|\tau|^3)$ est uniforme en y, pour x dans un voisinage de rayon inférieur à $\frac{1}{k_0}$ donné. Remarquons qu'il suffit de considérer les τ qui vérifient $|\tau| < \alpha\frac{1}{k}$ où α est une constante qui ne dépend que de la taille du support de χ. De même le jacobien de la transformation $x \to \tau$, $J(\tau)$ vaut $1 + \mathcal{O}(\frac{1}{k^2})$ au voisinage de y. Alors

$$\begin{aligned}
\int_\Gamma k^2 K_2(|x-y|)dx &= \int_\Gamma k\chi(k|x-y|)\frac{e^{ik|x-y|}}{4\pi|x-y|}dx \\
&= \int_{R^2} k\chi(k|\tau| + \mathcal{O}(\frac{1}{k^2}))\frac{e^{ik|\tau|}}{4\pi|\tau|}(1 + \mathcal{O}(\frac{1}{k^2}))d\tau \\
&= \left(\int_{R^2} k\chi(k|\tau|)\frac{e^{ik|\tau|}}{4\pi|\tau|}d\tau\right)(1 + \mathcal{O}(\frac{1}{k^2})) \\
&= C_\chi(1 + \mathcal{O}(\frac{1}{k^2}))
\end{aligned}$$

Et toutes les convergences sont ici uniformes en y, ce qui est dû au fait que Γ est compacte. En effet la convergence est uniforme carte par carte, et il y a un nombre fini de cartes. Ceci achève la preuve du lemme.

Lemme 9 *On suppose encore que la surface est au moins C^3, de dimension 2, alors*
$\int_\Gamma n_x.n_y k^2 K_2(|x-y|)dx = C_\chi(1+\mathcal{O}(\frac{1}{k^2}))$ *où C_χ est la constante de la partie précédente; de plus la convergence est uniforme en y.*

Preuve :

Les choses fonctionnent exactement comme pour le lemme précédent, il suffit de remarquer que le champ des normales est uniformément C^1 sur Γ et que la norme de n_x vaut 1.

Lemme 10 *Toujours sous les mêmes hypothèses, pour $0 < s < 1$,*
$\int_\Gamma k^4 {K_2}^2(|x-y|)|x-y|^{2+2s}dx = \mathcal{O}(\frac{1}{k^{2s}})$, *la convergence est encore uniforme en y.*

Preuve :

On procède comme précédemment, en paramétrant la surface par son plan tangent en y. On a alors :

$$\begin{aligned}\int_\Gamma k^4 {K_2}^2(|x-y|)|x-y|^{2+2s}dx &\leq \int_\Gamma k^2 \frac{\chi^2(k|\tau|)}{16\pi^2|\tau|^2}|\tau|^{2+2s}d\tau(1+\mathcal{O}(\frac{1}{k^2})) \\ &\leq C'\frac{1}{k^{2s}}.\end{aligned}$$

C' est ici une constante qui ne dépend que de χ et de s, elle vérifie l'inégalité suivante : $C' > \frac{1}{8\pi}\int_0^\infty \chi^2(r)r^{2s+1}ds$.
Ceci achève la preuve du lemme.

On rappelle ici une norme $H^s(\Gamma)$.

Proposition 3 *Si $\phi \in H^s(\Gamma)$ pour $0 < s < 1$, alors*
$||\phi||^2_{H^s(\Gamma)} = ||\phi||^2_{L^2} + \int_{\Gamma\times\Gamma} \frac{|\phi(x)-\phi(y)|^2}{|x-y|^{n+2s}}dxdy$ *où n est la dimension de la variété Γ.*

On peut se référer pour ceci à [LM68]. Nous allons maintenant montrer un lemme qui va utiliser cette norme.

Lemme 11 *Pour* $0 < s < 1$, $\exists C_1 > 0$,
(i) $|| \int_\Gamma k^2 K_2(|x-y|)(\phi(x)-\phi(y))dy||_{L^2}^2 \leq C_1 \frac{||\phi||_{H^s}^2}{k^{2s}}$
(ii) $|| \int_\Gamma k^2 K_2(|x-y|)(\phi(x)n_x.n_y-\phi(y))dy||_{L^2}^2 \leq C_1 \frac{||\phi||_{H^s}^2}{k^{2s}}$

Preuve :

Pour (i), on a $|| \int_\Gamma k^2 K_2(|x-y|)(\phi(x)-\phi(y))dy||_{L^2}^2$

$$\leq \int_\Gamma \left| \int_\Gamma k^2 K_2(|x-y|)(\phi(x)-\phi(y))dy \right|^2 dx$$
$$\leq \int_\Gamma \left(\int_\Gamma k^4 {K_2}^2 |x-y|^{2+2s} dy \int_\Gamma \frac{|\phi(x)-\phi(y)|^2}{|x-y|^{2+2s}} dy \right) dx \leq \frac{C_1}{k^{2s}} ||\phi||_{H^s}^2$$

La première inégalité s'obtient en utilisant l'inégalité de Cauchy-Schwarz, la seconde, les deux lemmes précédents. Pour (ii) on va évaluer
$|| \int_\Gamma k^2 K_2(|x-y|)(n_x.n_y-1)\phi(x)dx||_{L^2}$ puis en le combinant à (i) on aura le résultat souhaité.

$|| \int_\Gamma k^2 K_2(|x-y|)(n_x.n_y-1)\phi(x)dx||_{L^2}^2$

$$= \int_\Gamma dy \left(\int_\Gamma k^2 K_2(|x-y|)(n_x.n_y-1)\phi(x)dx \right)^2$$
$$\leq \int_\Gamma dy ||\phi||_{L^2}^2 \int_\Gamma k^2 \chi(k|x-y|)^2 \frac{(n_x.n_y-1)^2}{16\pi^2|x-y|^2} dx.$$

Or, pour x voisin de y, on a $n_x.n_y - 1 = \mathcal{O}(|x-y|)^2$ uniformément en y. En faisant à nouveau le désormais usuel changement de variable $x \to \tau$, on a alors

$|| \int_\Gamma k^2 K_2(|x-y|)(n_x.n_y-1)\phi(x)dx||_{L^2}^2$

$$\leq C_2||\phi||_{L^2}^2 \int_0^\infty k^2\chi(kr)^2\frac{r^4}{r^2}rdr$$
$$\leq C_3\frac{||\phi||_{L^2}^2}{k^2}.$$

Finalement, $||\int_\Gamma k^2K_2(|x-y|)(n_x.n_y-1)\phi(x)dx||_{L^2}^2 \leq C_3\frac{||\phi||_{L^2}^2}{k^2} \leq C_3\frac{||\phi||_{H^s}^2}{k^{2s}}$.
En ajoutant ce résultat à (i) on obtient (ii). On peut maintenant énoncer le théorème qui est semblable à celui de la partie précédente.

Théorème 2 *Pour* $0 < s < 1$, $\exists C_1 > 0$,
$\forall \psi \in H^1(\Gamma)$, $\forall \phi \in H^{s+1}(\Gamma)$,
$| < B\phi, \psi > +$

$$< C(Id + \frac{\Delta_\Gamma}{k^2})\phi, \psi > | \leq C_1(\frac{1}{k^s}||\phi||_{H^s}||\psi||_{L^2} + \frac{1}{k^{s+2}}||\phi||_{H^{s+1}}||\psi||_{H^1})$$

Preuve :

Nous allons commencer par considérer
$| < \int_\Gamma k^2K_2\phi(x)n_x.n_ydx, \psi > -C < \phi, \psi > |$. On a

$$| < \int_\Gamma k^2K_2\phi(x)n_x.n_ydx, \psi > -C < \phi, \psi > |$$

$$\leq ||\psi||_{L^2}||C\phi - \int_\Gamma k^2K_2(|x-y|)\phi(x)n_x.n_ydx||_{L^2}$$

$$\leq ||\psi||_{L^2}(||(C - \int_\Gamma k^2K_2n_x.n_y(|x-y|)dx)\phi||_{L^2}$$

$$+||\int_\Gamma k^2K_2(\phi(x)n_x.n_y - \phi(y))dx||_{L^2}).$$

La première partie se majore grâce au lemme 9, la seconde avec le lemme 11. On a ainsi

$$| < \int_\Gamma k^2K_2(|x-y|)\phi(x)n_x.n_ydx, \psi > -C < \phi, \psi > |$$

$$\leq C_1||\psi||_{L^2}||(\frac{||\phi||_{L^2}}{k^2}+\frac{||\phi||_{H^s}}{k^s})$$
$$\leq C_1||\psi||_{L^2}\frac{||\phi||_{H^s}}{k^s}.$$

Pour le deuxième terme on fait de même.

$$|<\int_\Gamma K_2(|x-y|)rot_\Gamma\phi(x)dx, rot_\Gamma\psi> -\frac{C}{k^2}<rot_\Gamma\phi, rot_\Gamma\psi>|$$

$$\leq ||\psi||_{H^1}||\frac{C}{k^2}rot_\Gamma\phi-\int_\Gamma K_2(|x-y|)rot_\Gamma\phi(x)dx||_{L^2}$$
$$\leq ||\psi||_{H^1}(||(\frac{C}{k^2}-\int_\Gamma K_2(|x-y|)dx)rot_\Gamma\phi||_{L^2}$$
$$+||\int_\Gamma K_2(|x-y|)(rot_\Gamma\phi(x)-rot_\Gamma\phi(y))dx||_{L^2})$$

De même que pour le premier terme la première partie se majore grâce au lemme 8, la seconde avec le lemme 11. On a ainsi
$|<\int_\Gamma K_2(|x-y|)rot_\Gamma\phi(x)dx, rot_\Gamma\psi> -\frac{C}{k^2}<\phi,\psi>|$

$$\leq C_1||\psi||_{H^1}||(\frac{||\phi||_{H^1}}{k^4}+\frac{||\phi||_{H^{s+1}}}{k^{s+2}})$$
$$\leq C_1||\psi||_{H^1}\frac{||\phi||_{H^{s+1}}}{k^{s+2}}.$$

L'addition des calculs faits sur chacune des deux composantes de l'expression initiale permet de terminer la preuve de ce théorème.

Le résultat obtenu dans cette partie est un peu différent de celui obtenu lorsque l'on a étudié le cas où la surface était un plan. Ceci est dû à l'absence de produit scalaire simple pour $H^s(\Gamma)$ d'une part et à l'inexistence d'un outil équivalent à la transformée de Fourier d'autre part. Cependant, dans le cas plan, nous aurions pu montrer un résultat exactement identique à celui que nous venons de montrer à la nuance près que ici on a $0 < s < 1$ alors que pour une surface plane nous aurions obtenu $0 < s \leq 1$.

Il y a une différence beaucoup plus fondamentale cependant entre les surfaces qui sont planes et celles qui ne le sont pas. En effet, on peut faire apparaître le

rôle de la courbure dans le calcul du commutateur du gradient avec B. Alors que ce commutateur est nul dans le cas d'une surface plane, comme on peut le voir avec la transformée de Fourier, il dépend, dans le cas général, de la courbure moyenne. Ce commutateur intervient dans le calcul de la norme de B pour des normes de Sobolev avec des indices s supérieurs à 1 (au lieu de $0 < s < 1$). Nous allons montrer le théorème suivant qui formalise un peu cette assertion.

Théorème 3 *Supposons que* Γ *est au moins* C^2. *Alors* $\exists C'_\chi$, $\forall s, 0 < s < 1, \exists \alpha > 0$, $\forall \phi \in H^1(\Gamma)$,

$$(i)||\nabla_\Gamma \int_\Gamma k^2 K_2(|x-y|)\phi(x)dx - \int_\Gamma k^2 K_2(|x-y|)\nabla_\Gamma\phi(x)dx - C'_\chi J\phi||_{L^2} \leq \alpha\frac{||\phi||_{H^s}}{k^s}$$

$$(ii)||\nabla_\Gamma \int_\Gamma k^2 K_2(|x-y|)n_x.n_y\phi(x)dx - \int_\Gamma k^2 K_2(|x-y|)n_x.n_y\nabla_\Gamma\phi(x)dx - C'_\chi J\phi||_{L^2} \leq \alpha\frac{||\phi||_{H^s}}{k^s}$$

où J *est la courbure moyenne :* $J = \dfrac{1}{R_1} + \dfrac{1}{R_2}$.

Ce théorème montre que, pour chacune des deux parties de l'opérateur de préconditionnement B, le commutateur avec le gradient est non nul et dépend explicitement de la courbure moyenne.

On verra, de plus, que la constante C'_χ s'exprime simplement en fonction de la constante C_χ. Avant de montrer ce théorème nous allons montrer quelques lemmes.

Lemme 12 *Si on note* π_{T_x}, π_{T_y} *les projections orthogonales sur les plans tangents à* Γ *en* x *et* y, *alors* $(\pi_{T_x} - \pi_{T_y})(x-y) = -(\dfrac{\tau_1^2}{R_1} + \dfrac{\tau_2^2}{R_2}) + \mathcal{O}(|\tau|^3)$ *où l'on a supposé encore une fois que la surface est paramétrée localement au voisinage de* y *par* $\tau = (\tau_1, \tau_2)$ *dans la base des directions principales de courbure de* Γ *en* y *(on peut toujours choisir un tel paramétrage).*

Preuve :

Rappelons d'abord que $\pi_{T_x} z = z - (z.n_x)n_x$ et $\pi_{T_y} z = z - (z.n_y)n_y$. On sait que $x - y = (\tau_1, \tau_2, \frac{\tau_1^2}{2R_1} + \frac{\tau_2^2}{2R_2}) + \mathcal{O}(|\tau^3|)$. On a alors que $\pi_{T_y}(x-y) = (\tau_1, \tau_2, 0)$. Regardons maintenant pour l'autre projection. Il est facile de voir que $n_x = (-\frac{\tau_1}{R_1}, -\frac{\tau_2}{R_2}, 1) + \mathcal{O}(|\tau|^2)$. Alors $((x-y).n_x) = -(\frac{\tau_1^2}{2R_1} + \frac{\tau_2^2}{2R_2}) + \mathcal{O}(|\tau|^3)$ et donc $((x-y).n_x)n_x = -(\frac{\tau_1^2}{2R_1} + \frac{\tau_2^2}{2R_2})n_y$. Par suite

$$(\pi_{T_x} - \pi_{T_y})(x-y) = -(\frac{\tau_1^2}{R_1} + \frac{\tau_2^2}{R_2})n_y + \mathcal{O}(|\tau|^3).$$

Lemme 13 *Sous les mêmes hypothèses,* $\nabla_\Gamma(n_x.n_y) = \mathcal{O}(|\tau|)$. *Ici le gradient est pris indépendamment en x ou en y.*

Preuve :

D'après l'expression de n_x on a $n_x.n_y = 1 + \mathcal{O}(|\tau|^2)$. Alors on a facilement que $\nabla_{\Gamma x}(n_x.n_y) = \mathcal{O}(|\tau|)$. Pour des raisons de symétrie, on a alors le même résultat pour le gradient surfacique en y.

Lemme 14 *On a*

$$(i) \int_\Gamma k^2(\pi_{T_x} - \pi_{T_y})\nabla_x K_2(|x-y|)dx = 2C_\chi nJ + \mathcal{O}(\frac{1}{k})$$

$$(ii) \int_\Gamma k^2(\pi_{T_x} - \pi_{T_y})\nabla_x K_2(|x-y|)n_x.n_y dx = 2C_\chi nJ + \mathcal{O}(\frac{1}{k}).$$

On a noté ∇_x *le gradient par rapport à x dans* $\mathbb{R}^3$.

Preuve :

Montrons d'abord (i) On note $\Xi(\rho) = e^{i\rho}\chi(\rho)$. On a alors

$$k^2\nabla_x K_2(|x-y|) = \frac{(x-y)}{4\pi|x-y|}(\frac{k^2\Xi'(k|x-y|)}{|x-y|} - \frac{k\Xi(k|x-y|)}{|x-y|^2}).$$

Alors

$$k^2(\pi_{T_x} - \pi_{T_y})\nabla_x K_2(|x-y|) =$$

$$-n_y[(\frac{\tau_1^2}{R_1} + \frac{\tau_2^2}{R_2}) + \mathcal{O}(|\tau|^3)](\frac{k^2\Xi'(k|\tau|)}{4\pi|\tau|^2} - \frac{k\Xi(k|\tau|)}{4\pi|\tau|^3})(1 + \mathcal{O}(|\tau|))$$

Ainsi, en passant en coordonnées polaires on a

$$\int_\Gamma k^2(\pi_{T_x} - \pi_{T_y})\nabla_x K_2(|x-y|)dx$$

$$\begin{aligned}
&= \int_0^\infty -n_y \frac{1}{4\pi}(k^2\Xi'(k\rho) - \frac{k\Xi(k\rho)}{\rho})\left(\int_0^{2\pi} \frac{\cos^2\theta}{R_1} + \frac{\sin^2\theta}{R_2} d\theta\right)\rho d\rho + \mathcal{O}(\frac{1}{k}) \\
&= -n_y \frac{J\pi}{4\pi}\int_0^\infty (\rho\Xi'(\rho) - \Xi(\rho))d\rho + \mathcal{O}(\frac{1}{k}) \\
&= -n_y \frac{J\pi}{4\pi}\int_0^\infty -2\Xi(\rho)d\rho + \mathcal{O}(\frac{1}{k}) \\
&= 2\pi n_y J C_\chi + \mathcal{O}(\frac{1}{k}).
\end{aligned}$$

Le terme en $\mathcal{O}(\frac{1}{k})$ vient de l'intégration de

$$\int_{k|\tau|<a} \mathcal{O}(|\tau|)|\tau|^2\left(\frac{k^2}{|\tau|^2} + \frac{k}{|\tau|^3}\right)d\tau = \int_{\rho<\frac{a}{k}} \mathcal{O}(\rho)(k^2\rho + k)d\rho = \mathcal{O}(\frac{1}{k}).$$

Pour (ii) la démonstration est identique.

Lemme 15 *On a*

$$\int_\Gamma k^2 K_2(|x-y|)(\nabla_{\Gamma_x}(n_x.n_y) + \nabla_{\Gamma_y}(n_x.n_y))dx = \mathcal{O}(\frac{1}{k})$$

Preuve :

Il suffit d'utiliser le lemme 13 et de faire l'intégrale en coordonnées polaires sur le plan tangent.

Lemme 16 $\forall s,\ 0 < s < 1, \exists \alpha > 0,\ \forall \phi \in H^s(\Gamma)$,

$$(i)||\int_\Gamma k^2(\pi_{T_x}-\pi_{T_y})\nabla_x K_2(|x-y|)(\phi(x)-\phi(y))dx||_{L^2} \leq \alpha\frac{||\phi||_{H^s}}{k^s}$$

$$(ii)||\int_\Gamma k^2(\pi_{T_x}-\pi_{T_y})\nabla_x K_2(|x-y|)n_x.n_y(\phi(x)-\phi(y))dx||_{L^2} \leq \alpha\frac{||\phi||_{H^s}}{k^s}$$

Preuve :

On a

$$||\int_\Gamma k^2(\pi_{T_x}-\pi_{T_y})\nabla_x K_2(|x-y|)(\phi(x)-\phi(y))dx||^2_{L^2} \leq$$

$$\int_\Gamma dy\left(\int_\Gamma k^4((\pi_{T_x}-\pi_{T_y})\nabla_x K_2(|x-y|))^2|x-y|^{2+2s}dx\right)\left(\int_\Gamma \frac{|\phi(x)-\phi(y)|^2}{|x-y|^{2+2s}}dx\right).$$

Il est facile de voir, en passant en coordonnées polaires, que la première intégrale est uniformément en y un $\mathcal{O}(\frac{1}{k^{2s}})$. La proposition 3 permet alors de conclure pour (i). Pour (ii) les choses se passent de manière identique.

Lemme 17 $\forall s,\ 0<s<1, \exists\alpha>0,\ \forall\phi\in H^s(\Gamma)$,

$$||\int_\Gamma k^2 K_2(|x-y|)(\nabla_{\Gamma_x}+\nabla_{\Gamma_y})n_x.n_y(\phi(x)-\phi(y))dx||_{L^2} \leq \alpha\frac{||\phi||_{H^s}}{k^{s+1}}.$$

Preuve :

On a

$$||\int_\Gamma k^2 K_2(|x-y|)(\nabla_{\Gamma_x}+\nabla_{\Gamma_y})n_x.n_y(\phi(x)-\phi(y))dx||^2_{L^2} \leq$$

$$\int_\Gamma dy\left(\int_\Gamma k^4 {K_2}^2(|x-y|)((\nabla_{\Gamma_x}+\nabla_{\Gamma_y})n_x.n_y)^2|x-y|^{2+2s}dx\right).$$

$$\left(\int_\Gamma k^2\frac{|\phi(x)-\phi(y)|^2}{|x-y|^{2+2s}}dx\right).$$

On procède, comme pour le lemme précédent, en remarquant (après passage en coordonnées polaires et usage du lemme 13) que la première intégrale est un $\mathcal{O}(\frac{1}{k^{2+2s}})$.

Lemme 18 $\forall s,\ 0 < s < 1, \exists \alpha > 0,\ \forall \phi \in H^s(\Gamma)$,

$$|| \int_\Gamma k^2 K_2(|x-y|)(\nabla_{\Gamma_x} + \nabla_{\Gamma_y}) n_x . n_y \phi(x) dx ||_{L^2} \leq \alpha \frac{||\phi||_{H^s}}{k}.$$

Preuve :

D'après le lemme 15 on sait que ,

$$|| \int_\Gamma k^2 K_2(|x-y|)(\nabla_{\Gamma_x} + \nabla_{\Gamma_y}) n_x . n_y dx \phi(y) ||_{L^2} \leq \alpha \frac{||\phi||_{L^2}}{k}$$

En combinant ce résultat avec celui du lemme précédent, on obtient que

$$|| \int_\Gamma k^2 K_2(|x-y|)(\nabla_{\Gamma_x} + \nabla_{\Gamma_y}) n_x . n_y \phi(x) dx ||_{L^2} \leq \alpha (\frac{||\phi||_{L^2}}{k} + \frac{||\phi||_{H^s}}{k^{s+1}}).$$

Ceci suffit à prouver le lemme car s est compris entre 0 et 1.

On peut maintenant revenir à la preuve du théorème. Remarquons d'abord que

$$\nabla_\Gamma \int_\Gamma k^2 K_2(x-y)\phi(x)dx = - \int_\Gamma k^2 \pi_{T_y} \nabla_x K_2(|x-y|)\phi(x)dx$$

et

$$- \int_\Gamma k^2 K_2(x-y) \nabla_\Gamma \phi(x) dx = \int_\Gamma k^2 \pi_{T_x} \nabla_x K_2(|x-y|)\phi(x)dx.$$

Alors

$$\nabla_\Gamma \int_\Gamma k^2 K_2(x-y)\phi(x)dx - \int_\Gamma k^2 K_2(x-y)\nabla_\Gamma \phi(x)dx$$

$$= \int_\Gamma k^2 (\pi_{T_x} - \pi_{T_y}) \nabla_x K_2(|x-y|)\phi(x)dx.$$

On conclut alors la preuve en additionnant les résultats du lemme 14 (i) et du lemme 16 (i). Pour le (ii) du théorème on procède de manière similaire. En effet, on remarque que

$$\nabla_\Gamma \int_\Gamma k^2 K_2(|x-y|) n_x.n_y \phi(x) dx - \int_\Gamma k^2 K_2(|x-y|) n_x.n_y \nabla_\Gamma \phi(x) dx =$$

$$\int_\Gamma k^2 (\pi_{T_x} - \pi_{T_y}) \nabla_x K_2(|x-y|) n_x.n_y \phi(x) dx + \int_\Gamma k^2 K_2(|x-y|)(\nabla_{\Gamma_x} + \nabla_{\Gamma_y}) n_x.n_y \phi(x) dx.$$

La combinaison des résultats des lemmes 14 (ii) et 16 (ii) pour la première intégrale et du lemme 18 pour la seconde, permet alors de conclure la preuve du théorème.

I.2.3 Application aux espaces d'éléments finis

Lorqu'on applique le résultat du théorème 2 aux espaces d'éléments finis, on a un résultat similaire à celui dans le cas plan. Ici on supposera que ψ est dans $H_h^1(\Gamma)$ et ϕ dans $H_h^{s+1}(\Gamma)$. Sous l'hypothèse que h.k est constant, on a alors, en utilisant les résultats déjà cités de [Cia91], que

$$< C(Id + \frac{\Delta_\Gamma}{k^2})\phi, \psi > | \leq C_1(\frac{1}{k^s}||\phi||_{H^s}||\psi||_{L^2}) \text{ pour } s,\ 0 < s < 1.$$

Ce résultat est un peu différent du cas plan. En effet, comme on n'a pu atteindre que la norme L^2 de ψ, au lieu de la norme H^s, pour s entre 0 et 1/2, l'exposant de k est de s au lieu de 2s. Cependant, le résultat est presque semblable puisque, dans le cas courbe, on a pu atteindre pour ϕ la norme H^s pour s, $0 < s < 1$ alors que dans le cas plan on avait la norme H^s de ϕ uniquement pour s,$0 < s \leq 1/2$.

I.3 Calcul du terme d'ordre suivant

Avant d'analyser ce que signifient ces résultats sur le plan de l'amélioration du préconditionnement ou de la vitesse de convergence de la méthode de résolution, nous allons regarder ce que vaut le terme d'ordre supérieur. Nous allons plus précisement montrer que celui-ci est nul. Les mêmes notations géométriques que

précédemment sont utilisées. Faisons tout d'abord remarquer qu'à des changements mineurs près dans les démonstrations de [LM68] pp.54-59, on a une proposition équivalente à la proposition 3.

Proposition 4 *Si* $\phi \in H^s(\Gamma)$ *pour* $0 < s < 1$, *alors* $\int_0^1 \left(\int_{\Gamma \times \Gamma} \frac{|\phi(y + t(y-x)) - \phi(y)|^2}{(t|x-y|)^{n+2s}} dxdy \right) dt \leq ||\phi||^2_{H^s}$ *où* n *est la dimension de la variété* Γ.

Nous allons ensuite présenter quelques lemmes.

Lemme 19 *Supposons* Γ C^2, *alors* $\exists\, k_0$, $\forall k \geq k_0$, $\exists \alpha > 0$,

$$(i) \quad \left| \int_\Gamma k^2 K_2(|x-y|)(x-y)dx \right| \leq \frac{\alpha}{k^2}$$

$$(ii) \quad \left| \int_\Gamma k^2 K_2(|x-y|) n_x.n_y (x-y)dx \right| \leq \frac{\alpha}{k^2}.$$

Preuve :

Remarquons que $(x-y).n = \frac{\tau_1^2}{2R_1} + \frac{\tau_2^2}{2R_2} + \mathcal{O}(|\tau|^3)$. Dans le support de K_2, $|x-y|$ devient suffisamment petit à partir d'un k que nous nommerons k_0 pour que $|(x-y).n| \leq \beta|\tau|^2$ avec $\beta > 0$. Alors $|x-y-\tau| \leq \beta|\tau|^2$. On a donc, en passant en coordonnées polaires,

$$\left| \int_\Gamma k^2 K_2(|x-y|)(x-y)dx \right| \leq \int_0^\infty \rho d\rho (k|\chi(k\rho)| \frac{1}{4\pi\rho} | \int_0^{2\pi} \tau d\theta |)(1 + \mathcal{O}(\frac{1}{k}))$$

$$+\beta \int_0^\infty k \frac{|\chi(k\rho)|}{2\rho} \rho^2 \rho d\rho$$

L'intégrale en θ est clairement nulle, puisqu'il s'agit d'intégrer un vecteur tournant de norme constante. Le calcul de la seconde intégrale termine la démonstration pour (i). Pour (ii) on procède exactement de même, en remarquant que $n_x.n_y - 1 = \mathcal{O}(|\tau|)$.

Lemme 20 *Avec les mêmes hypothèses que ci-dessus sur* Γ, *on a* $\forall s,\ 0 < s < 1, \exists C > 0,\ \forall \phi \in H^{s+1}$,

$$(i)|| \int_\Gamma k^2 K_2(|x-y|)(\phi(x) - \phi(y) - (x-y).\nabla\phi(y))dx||_{L^2} \leq C\frac{||\phi||_{H^{s+1}}}{k^{s+1}}$$

$$(ii)|| \int_\Gamma k^2 K_2(|x-y|)n_x.n_y(\phi(x) - \phi(y) - (x-y)\nabla\phi(y))dx||_{L^2} \leq C\frac{||\phi||_{H^{s+1}}}{k^{s+1}}$$

Preuve :

Remarquons d'abord que $(\phi(x)-\phi(y)-(x-y).\nabla\phi(y)) = \int_0^\infty (x-y).(\nabla\phi(y+t(x-y)) - \nabla\phi(y))dt$. On a alors que

$$\left(\int_\Gamma k^2 K_2(|x-y|)(\phi(x) - \phi(y) - \nabla\phi(y))dx\right)^2$$

$$= \int_0^1 dt \left(\int_\Gamma k^2 K_2(|x-y|)(x-y).(\nabla\phi(y+t(x-y)) - \nabla\phi(y))dx\right)^2 \leq$$

$$\int_0^1 t^{2+2s}dt \left(\int_\Gamma k^4 {K_2}^2(|x-y|)|x-y|^{4+2s}dx\right).$$

$$\left(\int_\Gamma \frac{(\nabla\phi(y+t(x-y)) - \nabla\phi(y))^2}{(t|x-y|)^{2+2s}}dx\right)$$

$$\leq \frac{C}{k^{2+2s}} \int_0^1 t^{2+2s}dt \left(\int_\Gamma \frac{(\nabla\phi(y+t(x-y)) - \nabla\phi(y))^2}{(t|x-y|)^{2+2s}}dx\right).$$

En utilisant la proposition 4 on obtient alors facilement que

$$|| \int_\Gamma k^2 K_2(|x-y|)(\phi(x) - \phi(y) - (x-y).\nabla\phi(y))dx||^2_{L^2} \leq C.\frac{||\phi||^2_{H^{s+1}}}{k^{2+2s}}.$$

Ceci montre le (i) du lemme. Pour le (ii), encore une fois on refait la même chose, sans modifications. Ceci achève la preuve du lemme.

Nous allons maintenant pouvoir énoncer le résultat principal.

Théorème 4 $\forall s,\ 0 < s < 1,\ \exists C,$

$$(i)\forall \phi \in H^{s+1}(\Gamma),\ || \int_\Gamma k^2 K_2(|x-y|)n_x.n_y\phi(x)dx - C_\chi\phi||_{L^2} \leq C\frac{||\phi||_{H^{s+1}}}{k^{s+1}}$$

$$(ii)\forall \phi \in H^{s+2}(\Gamma),\ || \int_\Gamma k^2 K_2(|x-y|) rot_\Gamma \phi(x) dx - C_\chi rot_\Gamma \phi||_{L^2} \leq C \frac{||\phi||_{H^{s+2}}}{k^{s+1}}$$

$$(iii)\forall \phi, \psi \in H^{s+2}(\Gamma),\ | < B\phi, \psi > + C_\chi < (Id + \frac{\Delta_\Gamma}{k^2})\phi, \psi > |$$

$$\leq C(\frac{||\phi||_{H^{s+2}}}{k^{s+2}} \frac{||\psi||_{H^1}}{k} + \frac{||\phi||_{H^{s+1}}}{k^{s+1}} ||\psi||_{L^2})$$

Preuve :

Il est déjà facile de voir que (iii) est une conséquence de (i) et (ii). Pour (i), en combinant les résultats du lemme 19 et du lemme 20 on obtient que

$$|| \int_\Gamma k^2 K_2(|x-y|) n_x . n_y (\phi(x) - \phi(y)) dx ||_{L^2} \leq C(\frac{||\phi||_{H^{s+1}}}{k^{s+1}} + \frac{||\phi||_{H^1}}{k^2})$$

$$\leq C' \frac{||\phi||_{H^{s+1}}}{k^{s+1}}.$$

En effet, il suffit de multiplier la deuxième inégalité du lemme 19 par $\nabla\phi$ puis d'en prendre la norme L^2, et de l'additionner à la deuxième inégalité du lemme 20. En combinant encore ce résultat avec celui du lemme 9 on obtient (i). Pour (ii), en combinant les résultats du lemme 19 (i) et du lemme 20 (i) appliqué à $rot_\Gamma \phi$ on obtient que

$$|| \int_\Gamma k^2 K_2(|x-y|)(rot_\Gamma \phi(x) - rot_\Gamma \phi(y)) dx ||_{L^2} \leq C(\frac{||\phi||_{H^{s+2}}}{k^{s+1}} + \frac{||\phi||_{H^2}}{k^2})$$

$$\leq C' \frac{||\phi||_{H^{s+2}}}{k^{s+1}}.$$

On combine encore ce dernier résultat avec celui du lemme 8 et on obtient (ii). Ceci termine la preuve du théorème.

I.4 Application au cas de la sphère

Essayons maintenant de voir comment agit le préconditionneur et sur quelle partie du spectre de l'opérateur A. Nous allons pour cela nous étudier le cas de la sphère de rayon 1 et regarder, pour k fixé, les spectres de A et de la limite de B, $Id + \frac{\Delta_\Gamma}{k^2}$, afin de les comparer.

Commençons par rappeler quelques particularités de la sphère et certaines fonctions spéciales qui lui sont liées. On pourra consulter à ce propos [NU88] ou [AS64].

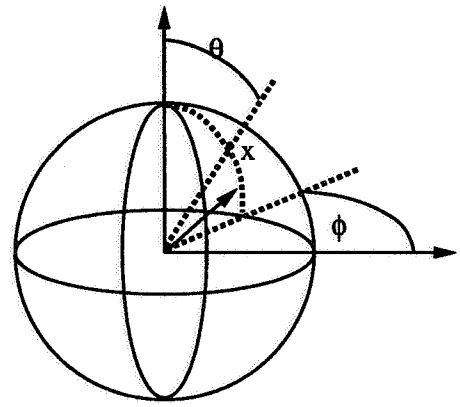


Figure I.4 : coordonnées de la sphère

Si θ, ϕ sont les coordonnées angulaires sur la sphère avec θ comme angle polaire (cf. figure I.4), on rappelle alors l'existence d'une base orthonormée de L^2 de la sphère formée des vecteurs propres de l'opérateur de Laplace-Beltrami sur la sphère, appelés harmoniques sphériques et notés $Y_{n,l}(\theta, \phi)$.

On a $Y_{n,l}(\theta,\phi) = \sqrt{\frac{2\pi}{2n+1}\frac{(n-l)!}{(n+l)!}} e^{il\phi} P_n^l(cos\theta)$ où les P_n^l sont les polynômes de Legendre d'indices n,l avec $-n \leq l \leq n$. $P_n^l(x)$ vérifie

$$(1-x^2)\frac{d^2P_n^l}{dx^2}(x) - 2x\frac{dP_n^l}{dx}(x) + (n(n+1) - \frac{l^2}{1-x^2})P_n^l(x) = 0.$$

On a $P_n^l(x) = \frac{(1-x^2)^{l/2}}{2^n n!}\frac{d^{n+l}}{dx^{n+l}}(x^2-1)^n$.

Il est alors facile de vérifier que $\Delta_S Y_{n,l}(\theta,\phi) = -n(n+1)Y_{n,l}(\theta,\phi)$ où Δ_S est l'opérateur de Laplace-Beltrami sur la sphère. On voit donc que $Y_{n,l}$ est un vecteur propre de $I + \frac{\Delta_\Gamma}{k^2}$ lorsque Γ est la sphère. En effet $(I + \frac{\Delta_\Gamma}{k^2})Y_{n,l} = (1 - \frac{n(n+1)}{k^2})Y_{n,l}$.

En fait, on va constater que l'ensemble des harmoniques sphériques est aussi une base formée de vecteurs propres de A. C'est ici une spécificité de la sphère que de connaître explicitement une base propre pour A et même d'en savoir l'existence. Introduisons auparavant deux séries de fonctions spéciales. Nous noterons $h_n(kr)$ les fonctions de Hankel sphériques et $j_n(kr)$ les fonctions de Bessel sphériques. Ces deux séries de fonctions sont solutions de l'équation de Bessel modifiée

$$\frac{d^2f}{dz^2} + \frac{2}{z}\frac{df}{dz} + (k^2 - \frac{n(n+1)}{z^2})f = 0.$$

Chaque série vérifie les relations de récurrence

$$\frac{2n+1}{kz}T_n(kz) = T_{n-1}(kz) + T_{n+1}(kz)$$

$$\frac{2n+1}{k}\frac{d}{dz}T_n(kz) = nT_{n-1}(kz) - (n+1)T_{n+1}(kz).$$

De plus, on a $j_0(kz) = \frac{sinkz}{kz}$ *et* $h_0(kz) = \frac{e^{ikz}}{ikz}$. Ceci suffit à déterminer entièrement les fonctions de Hankel et Bessel sphériques. Il est facile de vérifier que $j_n(kr)Y_{n,l}(\theta,\phi)$ vérifie l'équation de Helmholtz dans tout l'espace et est C^∞, c'est donc une solution d'un problème de Helmholtz intérieur à la sphère. Il est de même vrai que $h_n(kr)Y_{n,l}(\theta,\phi)$ vérifie l'équation de Helmholtz partout sauf en l'origine, mais réalise de plus la condition de radiation d'onde sortante de Sommerfeld, c'est donc une solution d'un problème de Helmholtz extérieur à la sphère. On a encore la relation suivante

$$h_n'(kz)j_n(kz) - h_n(kz)j_n'(kz) = \frac{i}{kz^2}.$$

On peut maintenant vérifier que $AY_{n,l} = h'_n(k)j'_n(k)\frac{k}{i}Y_{n,l}$.

En effet, donnons nous une condition de Neumann $g = Y_{n,l}$. Alors $\frac{h_n(kr)}{h'_n(k)}Y_{n,l}$ et $\frac{j_n(kr)}{j'_n(k)}Y_{n,l}$ sont des solutions respectivement extérieures et intérieures au problème de Helmholtz avec conditions de Neumann. Par conséquent

$$A^{-1}Y_{n,l} = -(\frac{h_n(k)}{h'_n(k)} - \frac{j_n(k)}{j'_n(k)})Y_{n,l},$$

soit

$$A^{-1}Y_{n,l} = \frac{j_n(k)h'_n(k) - j'_n(k)h_n(k)}{h'_n(k)j'_n(k)}Y_{n,l} = \frac{i}{kh'_n(k)j'_n(k)}Y_{n,l}.$$

La deuxième égalité résulte du calcul du wronskien pour l'équation de Bessel que l'on a considérée. On peut donc maintenant énoncer la proposition suivante.

Proposition 5 *A et $I + \frac{\Delta_S}{k^2}$ sont tous deux diagonalisables dans la même base, celle des harmoniques sphériques. On a*

$$\begin{aligned} AY_{n,l} &= h'_n(k)j'_n(k)\frac{k}{i}Y_{n,l} \\ (I + \frac{\Delta_S}{k^2})Y_{n,l} &= (1 - \frac{n(n+1)}{k^2})Y_{n,l} \end{aligned}$$

Calculons maintenant un équivalent de $h'_n(k)j'_n(k)$ lorsque n tend vers l'infini. On a $h'_n(k) \sim i\frac{(2n)!(n+1)}{2^n n! k^{n+1}}$ *et* $j'_n(k) \sim \frac{2^n n! n k^n}{(2n+1)}$. Alors $h'_n(k)j'_n(k)\frac{k}{i} \sim \frac{n(n+1)}{2n+1}$.

On peut remarquer que l'action de B sur le haut du spectre de A est de le ramener vers 0. En effet, $(I + \frac{\Delta_S}{k^2})^{-1}AY_{n,l} \sim \frac{k^2}{2n}Y_{n,l}$ pour n grand devant k. On peut alors se demander ce que l'on a gagné. En effet, du point de vue du conditionnement, avoir des valeurs propres très petites est aussi mauvais qu'avoir des valeurs propres très grandes. En fait le gain principal est que l'on aura beaucoup moins de vecteurs propres pour les valeurs propres élevées. En effet, les hautes valeurs propres de A correspondent à des n grands, pour lesquels

il y a beaucoup de vecteurs propres. Il y en a 2n+1 : ce sont les $Y_{n,l}$ pour l allant de -n à n. Ainsi B, s'il ne réduit pas beaucoup le conditionnement de A, a un rôle en fait aussi efficace car il rend le haut du spectre beaucoup plus clairsemé. Or, on sait que c'est le remplissage de cette partie du spectre qui conditionne pour une bonne partie la vitesse de convergence d'une méthode de résolution du type gradient conjugué, pourvu que l'on s'assure par ailleurs de la conjugaison exacte des directions de descente (cf. par exemple [VdSVdV86] ou [Rou91] dans le cas du gradient conjugué).

Dans ce qui vient d'être dit, on a toujours considéré n relativement à k. On peut donc penser que l'action de B sur le spectre de A se propage uniformément par rapport à k. Il faut noter que pour le bas du spectre de A, on ne peut pas dire grand-chose. En effet, A peut avoir une valeur propre aussi proche que l'on veut de 0. $I + \frac{\Delta_\Gamma}{k^2}$ quant à lui peut aussi, pour n au voisinage de k, avoir des valeurs propres petites. Cependant, on peut penser que globalement le spectre de A n'est pas trop modifié dans sa partie basse par le préconditionneur. En effet, pour n petit, B est proche de l'identité et donc ne modifie pas le spectre de A. Il n'y a que pour n dans un voisinage de k, ce qui ne concerne qu'un nombre faible de vecteurs propres, que l'on n'a pas d'information.

On peut voir aussi ce que deviendrait un tel préconditionneur pour le problème de Dirichlet. En prenant comme inconnue p le saut de la dérivée normale, la formulation variationnelle devient :
trouver $p \in H^{1/2}(\Gamma)$, $\forall q \in H^{1/2}(\Gamma)$,

$$\int_{\Gamma\times\Gamma} \frac{e^{ik|x-y|}}{4\pi|x-y|} p(x)q(y)dxdy = \int_\Gamma u_0(x)q(x)dx,$$

où u_0 est la donnée de Dirichlet sur la frontière. On peut se référer pour cela au chapitre d'introduction. En appliquant les résultats de ce chapitre et en particulier les lemmes 8 et 11 on voit alors que le préconditionneur obtenu par la même méthode (la troncature du noyau) tend en fait vers une constante fois l'identité. La constante est encore C_χ.

I.5 Comparaisons avec l'optique physique.

Une autre manière d'essayer de comprendre comment agit ce préconditionneur est de le comparer à l'approximation de l'optique physique dans le cas d'un objet convexe. Plus précisément, nous allons étudier des fonctions de la forme $\phi(x) = e^{i\vec{k}.x}\Phi(x)$ où $\vec{k}$ est un vecteur de norme k. Nous limiter à ces fonctions revient à supposer l'objet diffractant convexe. Cela suppose en effet qu'il n'y a pas de reflexions multiples des rayons sur la surface.Nous faisons cette restriction par souci de simplicité.

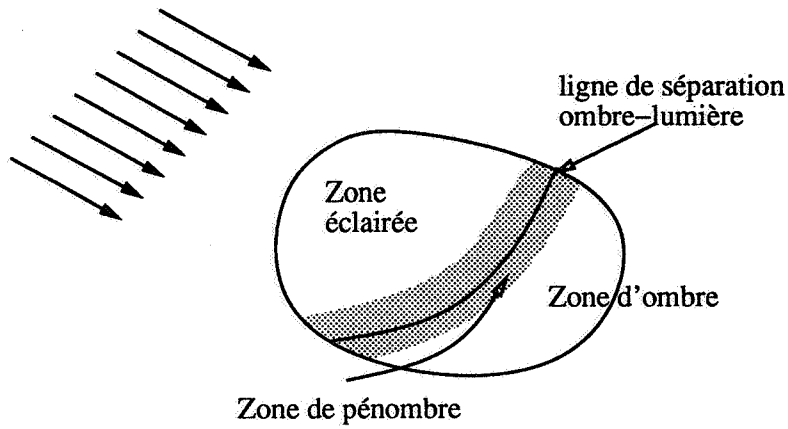


Figure I.5 :

Nous allons d'abord donner un certain nombre de définitions qui viennent de l'optique. Prenons comme onde incidente, l'onde plane $e^{i\vec{k}.x}$. Nous allons appeler dans ce cas *ligne de séparation ombre-lumière* la courbe placée sur Γ vérifiant $\vec{k}.n_x = 0$ pour x point de Γ. Nous allons appeler *zone de pénombre* un voisinage de la ligne de séparation ombre-lumière. Nous appelerons encore zone éclairée, l'ensemble des x de Γ tels que $\vec{k}.n_x < 0$ et zone d'ombre, l'ensemble des x de Γ tels que $\vec{k}.n_x > 0$. Tout ceci est illustré à la figure I.5. Par souci de simplicité, nous allons considérer le problème avec condition de Dirichlet. Dans ce cas l'inconnue est le saut de $\dfrac{\partial u}{\partial n}$. Rappelons dans ce cadre l'approximation de l'optique physique.

Si l'onde incidente est comme ci-dessus, alors $p = \left[\frac{\partial u}{\partial n}\right]$ *est approché par* $-2i\vec{k}.ne^{i\vec{k}.x}$ *dans la zone éclairée et par 0 dans la zone d'ombre.*

Cette approximation est valide sauf dans la zone de pénombre. Nous noterons $\Phi(x)$ une fonction qui vaut $-2i\vec{k}.n$ dans la zone éclairée et 0 dans la zone d'ombre. Regardons comment fonctionne cette approximation avec les équations intégrales. Nous allons évaluer l'intégrale $\int_\Gamma \frac{e^{ik|x-y|}}{4\pi|x-y|} e^{i\vec{k}.x}\Phi(x)dx.$ uniquement pour $y \in \Gamma$, en dehors de la zone de pénombre. Pour y fixé, nous allons décomposer Φ en $\Phi_1 + \Phi_2$ où Φ_1 est nulle en dehors d'un voisinage de y et Φ_2 est nulle au voisinage de y. On suppose de plus que le support de Φ_1 ne rencontre pas la ligne de séparation ombre-lumière. Nous allons d'abord calculer l'intégrale contenant Φ_1.

Proposition 6 *Pour y en dehors de la zone de pénombre*

$$\int_\Gamma \frac{e^{ik|x-y|}}{4\pi|x-y|} e^{i\vec{k}.x}\Phi_1(x)dx = e^{i\vec{k}.y}\frac{i\Phi(y)}{2|\vec{k}.n_y|} + \mathcal{O}(k^{-2})$$

Preuve :

Comme précedemment, nous paramétrisons la surface par son plan tangent. Nous prenons sur ce plan des coordonnées polaires (ρ, θ). Alors, pour avoir le terme principal, nous intégrons par parties en ρ. Ceci fonctionne comme pour la

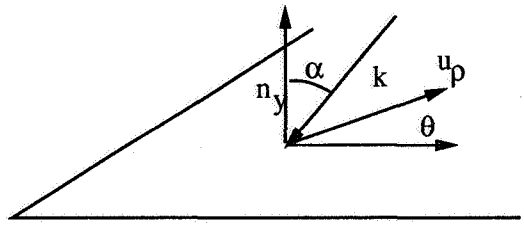


Figure I.6 :

phase stationnaire, lorsque la phase ne stationne pas, si bien qu'il ne reste que le terme tout intégré en ρ, pris en $\rho = 0$, puisque Φ_1 est à support compact. Il nous reste donc $\frac{-1}{4\pi}\int_0^{2\pi} \frac{1}{ik(1+\frac{\vec{k}}{k}.u_\rho)} d\theta.\Phi(y)e^{i\vec{k}.y}$ où u_ρ est le vecteur unitaire du plan

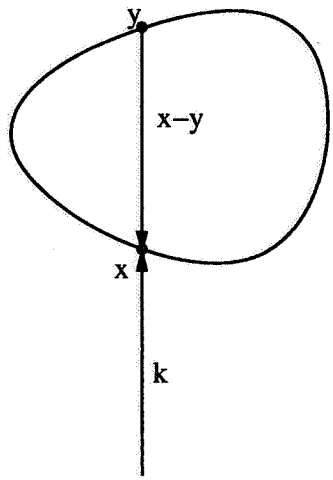


Figure I.7 : Transmission de l'onde à travers x, vers y.

tangent à Γ en y et d'angle polaire θ. Notons α, l'angle entre $\vec{k}$ et n_y (c.f. figure I.6). Ainsi, $\frac{\vec{k}}{k}.u_\rho = -\sin\alpha\cos\theta$. A l'aide du très classique changement de variable $t = \tan\frac{\theta}{2}$ on montre que $\int_0^\pi \frac{1}{1-\sin\alpha\cos\theta}d\theta = \frac{\pi}{|\cos\alpha|}$. Ainsi pour y dans la zone d'ombre l'intégale sur Φ_1 vaut 0, alors que dans la zone éclairée, elle vaut $-e^{i\vec{k}.x}$. Pour l'intégrale sur Φ_2, nous pouvons nous servir du théorème de la phase stationnaire(c.f. [CP81] ou [Dui73]). La phase vaut ici $k|x-y| + \vec{k}.x$. Elle stationne pour x vérifiant $\frac{x-y}{|x-y|} + \frac{\vec{k}}{k}$ parallèle à n_x. Ceci recouvre deux situations. La première est une situation de transmission : $\frac{x-y}{|x-y|} + \frac{\vec{k}}{k} = 0$ (c.f. figure I.7), la seconde une réflexion (c.f. figure I.8). Dans le cas convexe, que nous avons choisi, tous les points qui donnent des réflexions sont des points de la zone d'ombre. Ce qui fait que dans le cas des réflexions, la contribution donnée par la phase stationnaire est nulle. Pour le cas de la transmission, après calcul on constate que le déterminant de la Hessiènne de la phase vaut $\frac{|\vec{k}.n|^2}{2k|x-y|}$. Ceci fait, à l'aide du théorème de la phase stationnaire, que $\int_\Gamma \frac{e^{ik|x-y|}}{4\pi|x-y|}e^{i\vec{k}.x}\Phi_2(x)dx = \frac{ie^{i\vec{k}.y}\Phi(x_T(y))}{2|\vec{k}.n_{x_T}|}$ où $x_T(y)$ est le point qui réalise une transmission vers y. Ainsi, si y est dans la zone éclairée elle ne reçoit pas d'onde transmise et les réflexions ont une contri-

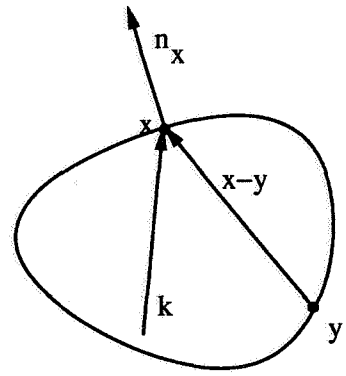


Figure I.8 : Réflexion de l'onde en x, vers y.

bution nulle, si bien que l'intégrale sur $\Phi 2$ à une contribution nulle au premier ordre. Pour y dans la zone d'ombre, seule l'onde transmise à une contribution. Celle-ci vaut $\frac{-ie^{i\vec{k}.y}2i\vec{k}.n_{x_T}}{-2\vec{k}.n_{x_T}} = -e^{i\vec{k}.y}$ au premier ordre. Ainsi, on voit que, mis à part dans la zone de pénombre, l'intégrale en Φ vaut $-e^{i\vec{k}.y}$ au premier ordre, ce qui est bien la condition de Dirichlet pour l'onde incidente étudiée. La zone où l'approximation est mauvaise est la zone de pénombre. Celle-ci correspond à la zone où la *théorie géométrique de la diffraction* fait intervenir des *rayons rampants.* En restant dans le cadre de l'optique physique c'est l'approximation de Fock (c.f. [Foc46]) qu'il faut utiliser dans cette région. Si nous avions voulu faire l'étude pour le problème avec condition de Neumann, il aurait suffit de remplacer le noyau par celui qui vient de la formulation de Hamdi.

Revenons maintenant à notre préconditionneur. Nous voulons calculer, pour y sur la surface Γ,

$$F(y) = \int_\Gamma \frac{e^{ik|x-y|}}{|x-y|} k\chi(k|x-y|)e^{i\vec{k}.x-y}\Phi(x)dx.$$

Comme d'habitude nous allons approcher la surface par son plan tangent. Au premier ordre près, nous avons

$$F(y) = \Phi(y) \int_0^{2\pi} \int_0^{\infty} \frac{e^{i\rho(1+\sin\alpha\cos\theta)}}{4\pi} \chi(\rho) d\rho d\theta.$$

Nous allons d'abord intégrer en θ. Nous reconnaissons la fonction de Bessel $J_0(x) = \frac{1}{2\pi} \int_0^{2\pi} e^{ix\cos\phi} d\phi$. Nous avons alors, qu'au premier ordre, $F(y)$ vaut

$$\Phi(y) \frac{1}{2} \int_0^{\infty} J_0(\rho \sin\alpha) e^{i\rho} \chi(\rho) d\rho.$$

Dans le cas du problème de Neumann le calcul fonctionne de manière similaire, en particulier pour ce qui concerne les calculs de phase stationnaire. Aussi l'explication qui suit est encore valable pour ce problème.

On peut maintenant tenter d'expliquer l'action de notre préconditionneur de manière différente. On voit ici que sa principale qualité est de ne pas déphaser le signal. Ainsi, on aura une bonne approximation des fréquences hautes du signal, qui correspondent à à la partie $e^{i\vec{k}.x}$. Par contre le préconditionneur sera assez mauvais, pour ce qui est des fréquences basses puisqu'il transforme complètement la fonction Φ. Autre manière de voir cela, de façon plus intuitive, est de constater que le préconditionneur ne prend en compte que des interactions entre points proches de moins d'une longueur d'onde. Il ne pourra donc approximer correctement que les fréquences qui correspondent à ce niveau. Il est difficile de tirer plus d'enseignements à partir de calculs de phase stationnaire car ceux-ci font intervenir la valeur des fonctions en des points particuliers, ce qui est difficilement compatible avec un contrôle des fonctions en norme de Sobolev.

Chapitre II

Résultats numériques

Nous allons, dans ce chapitre, essayer de vérifier que le préconditionneur exhibé précédemment est un bon préconditionneur. Nous commencerons par dire quels sont les algorithmes que nous utilisons pour la résolution, puis nous montrerons des résultats de calcul que nous commenterons. L'objet que nous considérons ici est une boule de rayon 1 et parfaitement réfléchissante. La surface Γ est donc une sphère de même rayon et les conditions aux limites sont celles de Neumann.

II.1 Algorithme de résolution

La lecture de [Rok83], ainsi que d'autres essais effectués à l'ONERA sur des matrices pleines provenant d'équations intégrales, nous ont conduits à utiliser comme algorithme de résolution du système linéaire l'algorithme du résidu conjugué généralisé (GCRA) aussi appelé *Orthomin*. Nous rappelons ici cet algorithme dans sa version non préconditionnée. On cherche à résoudre dans R^n, Ax = b où A est une matrice inversible de $M_n(R)$ ou $M_n(C)$ et b un vecteur de R^n ou C^n. ϵ est la valeur du résidu que l'on cherche à atteindre.

- Initialisation

  $x^0 \in R^n$ quelconque

  $r^0 = b - Ax^0$

$$d^0 = r^0$$

$$z^0 = Ad^0$$

$$k = 0$$

- Itérations sur k

$$\alpha^k = \frac{(r^k, z^k)}{(z^k, z^k)}$$

$$x^{k+1} = x^k + \alpha^k d^k$$

$$r^{k+1} = r^k - \alpha^k z^k$$

test d'arrêt : $\|\mathrm{r}^{\mathrm{k+1}}\| \leq \epsilon$

$$d^{k+1} = r^{k+1} + \sum_{l=0}^{k} \beta_l^{k+1} d^l$$

$$z^{k+1} = Ar^{k+1} + \sum_{l=0}^{k} \beta_l^{k+1} z^l$$

$$\beta_l^{k+1} = -\frac{(Ar^{k+1}, z^l)}{(z^l, z^l)} \; \forall l = 0, ..., k$$

Ici (,) représente le produit scalaire ou hermitien suivant le cas. Les r_k sont les résidus et les d_k les directions de descente. Il assure d'une part que $\forall\, k \neq l$, $(z^k, z^l) = 0$ et d'autre part, que $(r^k, Ar^l) = 0 \; \forall\, l,\; 0 \leq l < k$. On remarque, de plus, que dans le cas où A est une matrice hermitienne, les β_l^{k+1} sont nuls sauf pour l=k.

Les performances de l'algorithme non préconditionné nous ont conduits ensuite à rechercher une manière de le préconditionner. La première méthode fut de résoudre $B^{-1}Ax = B^{-1}b$ avec GCRA, où B est le préconditionneur. Cependant le système en B était inversé lui même par une méthode itérative et avec une précision assez faible. On cumulait alors deux inconvénients. D'une part la convergence de l'algorithme global était limitée par l'imprécision sur B^{-1}, ceci faisant que l'algorithme, qui avait une bonne vitesse de convergence dans les premières itérations stagnait très vite; d'autre part le résidu effectivement testé avait de moins en moins de sens. En effet le B^{-1} qui intervenait dans $B^{-1}b$

n'était pas le même que celui calculé quelques itérations plus tard, puisqu'il était recalculé à chaque fois avec une méthode itérative. On a alors essayé de préconditionner en s'inspirant du préconditionnement du gradient conjugué classique. L'algorithme s'écrit alors:

- Initialisation

$$x^0 \in R^n \text{quelconque}$$
$$r^0 = b - Ax^0$$
$$d^0 = B^{-1}r^0$$
$$z^0 = Ad^0$$
$$k = 0$$

- Itérations sur k

$$\alpha^k = \frac{(r^k, z^k)}{(z^k, z^k)}$$
$$x^{k+1} = x^k + \alpha^k d^k$$
$$r^{k+1} = r^k - \alpha^k z^k$$
$$\text{test d'arrêt} : \|r^{k+1}\| \leq \epsilon$$
$$d^{k+1} = B^{-1}r^{k+1} + \sum_{l=0}^{k} \beta_l^{k+1} d^l$$
$$z^{k+1} = AB^{-1}r^{k+1} + \sum_{l=0}^{k} \beta_l^{k+1} z^l$$
$$\beta_l^{k+1} = -\frac{(AB^{-1}r^{k+1}, z^l)}{(z^l, z^l)} \; \forall l = 0, ..., k$$

Avec cet algorithme on a encore $\forall k \neq l \; (z^k, z^l) = 0$. Ceci est une conséquence immédiate de la définition des β_l^k.
La relation de conjugaison s'exprime de la manière suivante,
$(r^k, B^{-1}Ar^l) = 0 \; \forall l, \; 0 \leq l < k$.
On a ensuite, par récurrence sur k que
$(r^k, AB^{-1}r^l) = (r^k, z^l) = 0 \; \forall l, \; 0 \leq l < k$.

De plus x^{k+1} minimise (r^{k+1}, r^{k+1}) sur $x^0 + K(B^{-1}A, B^{-1}r^0, k)$ où $K(A, u, n)$ est l'espace de Krylov engendré par les $A^i u$, $0 \leq i \leq n$.

Le préconditionneur que nous utilisons ici correspond à celui exposé dans le chapitre précédent en prenant comme fonction χ l'indicatrice du segment $[0, \alpha]$ où α est une constante à ajuster. Ceci revient à ne retenir de la matrice que les interactions correspondant à des points du maillage d'une distance inférieure à $\frac{\alpha}{k}$.

II.2 Tests numériques

Venons en maintenant aux résultats numériques. L'onde incidente utilisée est une harmonique sphérique, afin de pouvoir comparer la solution obtenue avec la solution analytique, mais nous avons aussi essayé avec une onde plane, qui est un second membre beaucoup plus réaliste et les qualités du préconditionneur demeurent dans ce cas. Le maillage qui est mis sur cette surface comporte 2048 facettes triangulaires et 1026 points. Si h est la longueur de la plus grande arête de la triangulation, alors on a ici $h = 0.165$. Expérimentalement la fréquence critique du maillage (celle au delà de laquelle on estime que la convergence vers la solution analytique n'a plus lieu) vaut $F_c = 780\ Hz$, ce qui correspond à une longueur d'onde λ_c de $0.427m$. Cette fréquence critique a été déterminée en comparant des solutions numériques et analytiques. La notion de convergence est ici regardée vis à vis du champ lointain. Si on s'était intéressé à la différence de pression sur la surface, il aurait fallu être beaucoup plus sévère : prendre h de l'ordre de $\frac{\lambda}{20}$. On a donc $h = \frac{\lambda_c}{2.59}$. On a d'abord exécuté l'algorithme de résidu conjugué généralisé sans préconditionnement, en balayant en fréquence l'intervalle [0Hz,780Hz]. Pour différentes valeurs du critère d'arrêt ϵ de l'algorithme on a noté le nombre d'itérations nécessaires pour converger. Le résultat est dans le tableau II.1. Dans ce tableau on note que le conditionnement se détériore avec l'augmentation de la fréquence. La non-convergence à 500Hz s'explique probablement par le fait que l'on se trouve au voisinage d'une fréquence propre du

ϵ^2	100Hz	200Hz	300Hz	400Hz	500Hz
10^{-4}	6	4	10	90	> 300
10^{-5}	10	7	14	> 300	> 300
10^{-6}	11	8	18	> 300	> 300

ϵ^2	600Hz	700Hz	750Hz	780Hz
10^{-4}	185	85	265	>300
10^{-5}	> 300	> 300	> 300	> 300
10^{-6}	> 300	> 300	> 300	>300

Tableau II.1 : nombre d'itérations pour GCRA non préconditionné.

problème de Dirichlet intérieur.

On a ensuite testé l'algorithme GCRA avec le préconditionnement décrit précédemment pour des fréquences proches de la fréquence maximale autorisée par le maillage, et aussi pour 500Hz. Ici B est inversé par une méthode itérative. On fixe pour cet algorithme le nombre d'itérations (dans notre cas, il a été fixé à 50, ce qui correspond à un résidu de l'ordre de 10^{-1} en général). On note δ la distance maximale entre les centres de deux facettes pour que leur interaction soit prise en compte dans le préconditionneur. Dans le tableau II.2 on montre le nombre de coefficients de la matrice retenus pour le préconditionneur pour différentes valeurs de δ. Ce nombre est à comparer avec les 1 052 676 coefficients de la matrice complète A. On voit que dans le pire des cas le préconditionneur

δ	nombre de coefficients
0.44m	85490
0.40m	74962
0.36m	63618
0.32m	53426
0.28m	45090
0.24m	36010

Tableau II.2 : nombre de coefficients du préconditionneur.

fait moins du dixième de la matrice pleine. On va maintenant donner, dans les tableaux II.3 et II.4, les nombres d'itérations qu'il a fallu pour atteindre un résidu donné. On peut ici remarquer plusieurs choses. D'une part la méthode de préconditionnement donne globalement d'excellents résultats pour des fré-

δ	780Hz			750Hz			700Hz		
	10^{-4}	10^{-5}	10^{-6}	10^{-4}	10^{-5}	10^{-6}	10^{-4}	10^{-5}	10^{-6}
0.40m	3	5	22	2	4	26	3	4	10
0.36m	2	5	35	2	4	25	3	4	9
0.32m	3	5	21	2	4	12	3	4	10
0.28m	3	8	20	2	7	17	3	11	24
0.24m	2	30	> 50	2	17	> 50	4	15	49

Tableau II.3 : nombre d'itérations pour GCRA préconditionné.

δ	600Hz			500Hz		
	10^{-4}	10^{-5}	10^{-6}	10^{-4}	10^{-5}	10^{-6}
0.44m	2	5	14	12	25	> 50
0.40m	2	4	10	12	> 50	> 50
0.36m	3	7	11	> 50	> 50	> 50
0.32m	3	10	18	> 50	> 50	> 50
0.28m	5	27	> 50	17	> 50	> 50
0.24m	4	17	45	24	37	> 50

Tableau II.4 : nombre d'itérations pour GCRA préconditionné.

quences élevées, puisque l'on passe de cas de convergences en quelques centaines d'itérations à une dizaine d'itérations pour $\epsilon^2 = 10^{-5}$ par exemple. D'autre part on voit que la distance δ optimale pour le préconditionneur est effectivement inversement proportionnelle à la fréquence, ce que montraient par ailleurs les résultats théoriques obtenus plus haut. Les irrégularités observées pour un résidu de 10^{-6} à 780 Hz viennent d'une part du fait que cette fréquence est un peu élevée pour le maillage et d'autre part du fait que le système du préconditionneur n'est pas résolu avec suffisamment de précision. Le cas de la fréquence de 500Hz est plus complexe à analyser. On est proche d'une fréquence propre du problème de Dirichlet intérieur et on a donc un très mauvais conditionnement. Ceci apparait à deux niveaux. Non seulement on met un très grand nombre d'itérations pour converger si on ne préconditionne pas correctement, mais aussi, même dans le cas d'une bonne convergence, le champ lointain calculé est encore à un peu plus de 10 % du champ lointain théorique. Ceci montre que la source du mauvais conditionnement ne vient plus uniquement de l'augmentation de la fréquence. On

δ	temps d'inversion
0.44m	2.73s
0.40m	2.48s
0.36m	2.25s
0.32m	2.01s
0.28m	1.70s
0.24m	1.54s

Tableau II.5 : Temps de résolution d'un système pour lepréconditionneur

remarque cependant que, même dans ce cas, le nombre d'itérations est fortement diminué, tout au moins dans le cas d'un résidu pas trop petit. On va enfin donner des résultats en vitesse de calcul. Le programme a été testé sur le Cray-II du C.C.V.R. utilisé en monoprocesseur. A titre de comparaison on donne le temps de calcul du produit Matrice-Vecteur, pour la matrice pleine qui est de 0.13s. Pour le préconditionneur, le temps d'inversion est donné dans le tableau II.5. Ceci fait que, pour la fréquence de 700Hz, le temps de résolution passe de 11s à 6.5s et pour 750Hz de 34s à 4.3s, en préconditionnant avec $\delta = 0.32m$ et pour un résidu de 10^{-4}. Ces résultats montrent quel peut être l'intérêt de ce préconditionneur même sur un calculateur vectoriel qui défavorise pourtant les calculs avec adressage indirect, puisque d'une part on voit que lorsque l'algorithme ne converge pas, une fois préconditionné il converge, et que d'autre part cette convergence n'est pas d'un coût prohibitif car dans les cas où la comparaison est possible, on gagne beaucoup de temps à préconditionner.

Chapitre III

Parallélisation de l'algorithme

Introduction

Nous allons, dans ce chapitre, présenter une implémentation sur une machine parallèle du programme dont nous avons montré des résultats dans le chapitre précèdent. Nous allons commencer par un rappel sur les deux grandes familles d'ordinateurs parallèles MIMD : ceux à mémoire partagée et ceux à mémoire distribuée. Nous montrerons ensuite quels modèles de programmation sont utilisés pour chacune des deux familles. Par la suite nous rentrerons plus dans le détail pour la machine que nous avons effectivement utilisée avant de montrer comment le programme a été parallélisé. Nous donnerons enfin des résultats en termes de vitesse de calcul et d'efficacité du parallélisme.

Lorsque l'on parle d'ordinateurs parallèles ou multiprocesseurs, on englobe une vaste catégorie de machines qui est très hétérogène. En effet, leur seul point commun à tous est d'avoir au moins deux processeurs. Rien n'est dit de la manière dont ils fonctionnent ensemble ou de la manière dont ils gèrent les données, communes ou non, par exemple. Néanmoins, on arrive à dégager deux grandes classes de machines qui sont plus ou moins fortement liées à deux modèles de programmation distincts.

La première classe à être apparue historiquement de manière importante et commerciale est celle des ordinateurs à mémoire partagée. C'est aussi celle qui

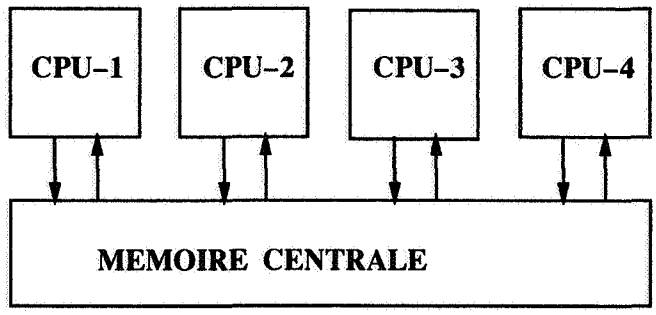


Figure III.1 : Architecture à mémoire partagée

est la plus utilisée de nos jours dans le domaine du calcul scientifique. Elle contient des machines comme les Cray, les Convex ou les Alliant FX par exemple. Leur principale caractéristique est d'avoir une seule mémoire en commun pour tous les processeurs. Cette mémoire est vue et accessible par tous les processeurs. Ce procédé à l'avantage de conduire à une manière de programmer qui ressemble fort à celle utilisée sur les monoprocesseurs. En effet, les processeurs ne communiquent pas à proprement parlé. La plupart du temps, ils travaillent de manière complètement indépendante sur des données différentes a priori. La synchronisation est gérée par le système d'exploitation soit sur requête du programmeur, soit parce que le compilateur a détecté la nécessité d'une synchronisation. Elle repose naturellement dans la plupart des cas sur des mécanismes hardware afin d'être efficace. La plupart du temps ces machines sont accompagnées d'un précompilateur qui détecte un minimum de parallélisme au niveau des boucles et qui comprend des directives de compilation données dans le code source ayant trait à la vectorisation ou à la parallélisation. Ce genre de machines est souvent composé d'un petit nombre de processeurs très puissants. De plus il n'est pas possible d'espérer raisonnablement voir rapidement et même à plus longue échéance la puissance de ces machines augmenter notablement que ce soit grâce à des améliorations technologiques ou par une forte augmentation du nombre de processeurs. En effet, il existe une contrainte technique qui interdit cette croissance : il s'agit de la complexité croissante du réseau d'interconnexion entre les processeurs et la mémoire au fur et à mesure que l'on augmente le nombre de

ceux-ci et qui est nécessaire pour pouvoir les alimenter en parallèle. Ceci fait que c'est réellement la mémoire partagée qui est la caractéristique de cette classe de machines .

Face à cet état de ralentissement de la croissance des performances et devant l'appétit toujours plus grand des numériciens se développent donc des machines à beaucoup plus grand nombre de processeurs et qui se sont affranchies de la contrainte de la mémoire partagée. L'extrême dans ce domaine est atteint par les machines à mémoire distribuée. On compte parmi celles-ci les machines du genre

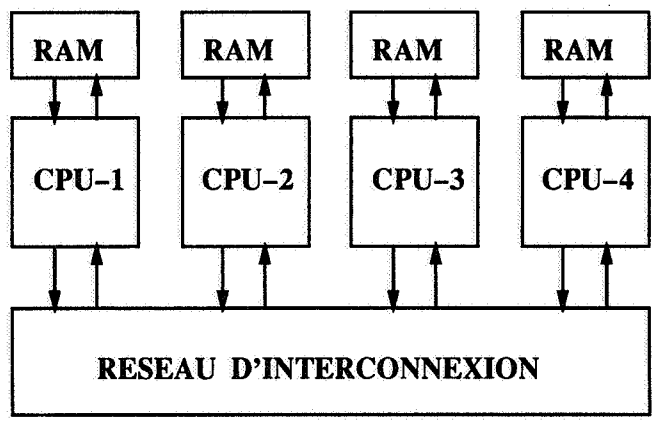


Figure III.2 : Architecture à mémoire distribuée

hypercube comme le Ncube ou l'iPSC ou des machines à base de transputers par exemple. Chaque processeur est ici muni de sa propre mémoire. Il exécute son propre jeu d'instructions sur ses propres données sans se préoccuper a priori de ce qui se fait par ailleurs. Néanmoins il est possible de les faire travailler ensemble. Pour ceci le mécanisme de synchronisation utilisé est l'échange de messages. Les messages peuvent, soit être échangés uniquement dans un but de synchronisation, leur contenu importe alors peu, soit contenir des données que l'on souhaite transférer à d'autres processeurs. Pour le programmeur la vision est alors exactement l'inverse de celle qu'il avait pour les machines à mémoire partagée. En effet, au lieu de considérer toutes les données comme a priori communes, ce qui nécessite de s'assurer que deux processeurs n'écrivent

pas en même temps sur la même variable, on doit regarder l'espace mémoire d'un processeur comme étant privé et les données qui y sont contenues comme connues uniquement du processeur détenteur de la mémoire. On doit, si l'on veut qu'une donnée soit connue de plusieurs processeurs, la faire migrer de manière explicite du processeur d'origine vers les processeurs destinataires. L'avenir de ces machines est a priori prometteur, puisque l'on peut plus facilement imaginer une forte montée en puissance de celles-ci; en effet le nombre de processeurs peut être très grand, par exemple de l'ordre du millier. De plus, il y a de grands gains à espérer du coté des composants puisque ceux-ci sont plutôt lents en comparaison des processeurs des machines à mémoire partagée. On commence seulement à voir apparaitre des processeurs pouvant atteindre une à quelques dizaines de Mflops en scalaire. Cependant il existe un frein à leur diffusion: le modèle de programmation de l'échange de messages, qui est le plus utilisé sur ces machines du fait qu'il est celui qui colle le plus à leur nature physique, fait qu'on est obligé de reprogrammer ses applications, ce que les industriels n'aiment pas. De plus c'est une manière de concevoir les programmes qui est inhabituelle pour les numériciens. Pour pallier ceci on voit se développer le concept de mémoire partagée virtuelle. Celui-ci consiste à considérer l'ensemble des mémoires comme une seule mémoire et chaque mémoire locale comme un cache. Il serait alors possible de programmer les machines à mémoire distribuée comme des machines à mémoire partagée à la nuance près que le coût d'accès à une page de mémoire sera fortement non uniforme.

III.1 Description de la machine cible

Nous allons maintenant présenter la machine sur laquelle a été réalisée l'implémentation du code d'équations intégrales. Il est intéressant de rentrer un peu dans le détail de l'architecture de celle-ci pour comprendre les choix de programmation qui ont été faits.

III.1.1 Architecture de l'iPSC

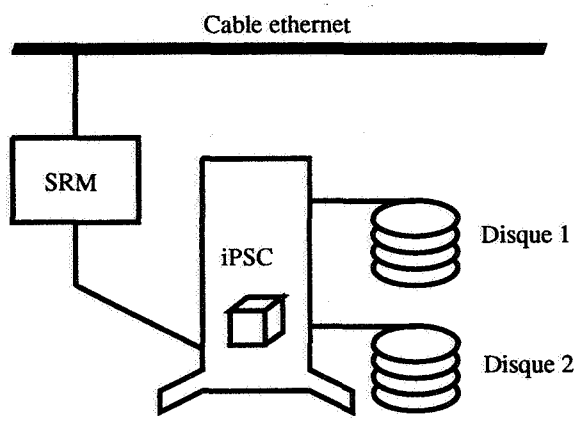


Figure III.3 : Le système iPSC-II

L'iPSC est un ordinateur multiprocesseur à mémoire distribuée. Ceci signifie donc qu'il est composé de "noeuds" comportant chacun un processeur et de la mémoire, plus un réseau de communication entre ces noeuds. Le réseau a une topologie du type "hypercube", c'est à dire que chaque noeud est situé sur un sommet de l'hypercube et a comme voisins ceux qui sont situés sur une même arête. Dans la figure III.4 sont dessinés quelques hypercubes pour des dimensions allant de 1 à 4. Les sommets sont numérotés en binaire. Les arêtes hachurées correspondent à la dimension qui vient d'être rajoutée. Notons que pour le 4-cube on peut aussi le représenter comme à la figure III.5 sous la forme d'une grille torique à deux dimensions (les noeuds y sont numérotés en base 10). L'iPSC-II actuellement utilisé a un réseau à 32 noeuds connecté avec l'extérieur avec une machine hôte appelée S.R.M. (System Resource Manager) et qui sert de frontal. Il faut noter de plus que deux de ces noeuds ont des disques qui sont vus de manière transparente par tout le réseau. Ceci est possible grâce au C.F.S. (Concurrent File System) qui est un ajout fait au système d'exploitation des noeuds.

Nous allons maintenant rappeler quelques propriétés de l'hypercube. Pour plus de détails on pourra se reporter à [SS85].

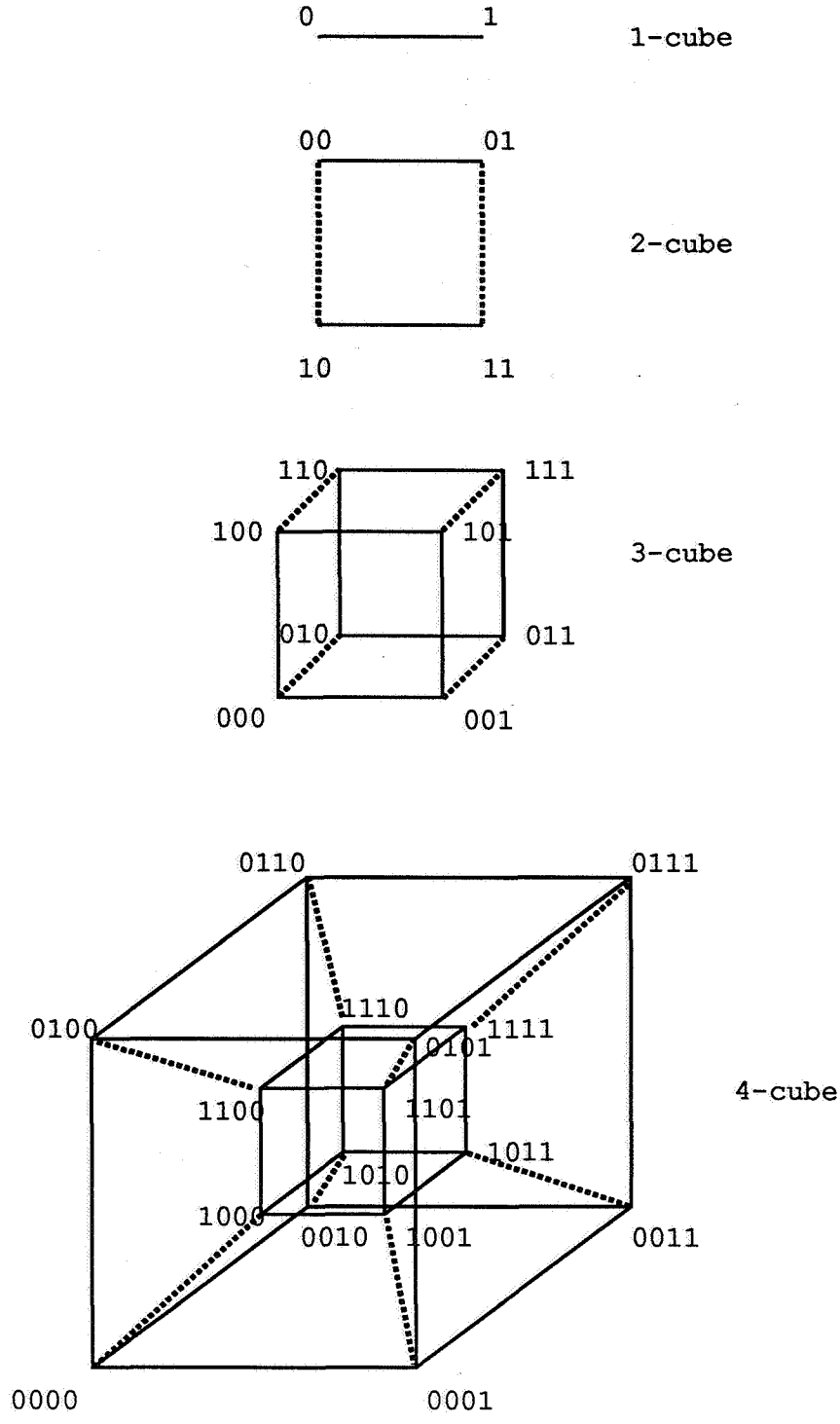


Figure III.4 : n-cubes

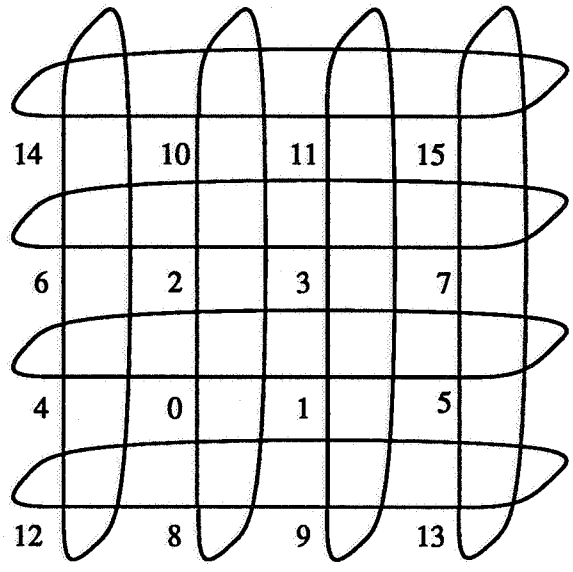


Figure III.5 : 4-cube représenté en grille

- Si n est la dimension de l'hypercube, alors il a 2^n sommets et son diamètre est n.

- On peut facilement déterminer les voisins d'un noeud à partir de la décomposition de son numéro en base 2. En effet deux noeuds sont voisins si et seulement si la décomposition en base 2 ne diffère que de 1 bit.

- Il existe de plus un moyen simple de déterminer un chemin allant du noeud A au noeud B ayant pour longueur la distance entre A et B. On va le présenter sur un exemple. Prenons A=3 et B=14.
  A s'écrit en base 2 : 0011 et B : 1110. On va alors de A à A1 = 0010, puis de A1 à A2 = 0110 et enfin de A2 à A3 = B = 1110. On a donc un algorithme de routage statique très simple pour décrire un chemin d'un point à un autre. Cela dit, ce système n'a pas que des avantages car il n'optimise pas l'utilisation du réseau. Il peut alors se produire des problèmes de contention sur le réseau.

III.1.2 Architecture d'un noeud

Venons en maintenant à la description plus précise d'un noeud de l'iPSC. Il est

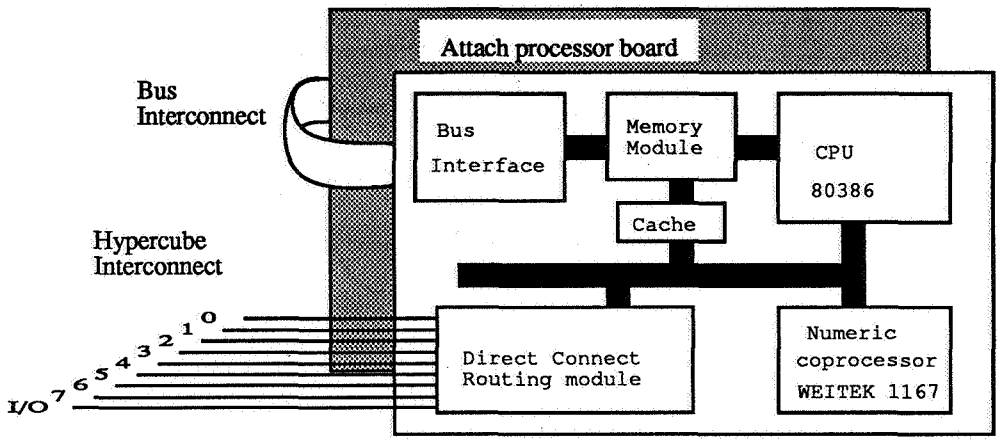


Figure III.6 : Schéma d'un noeud

composé

- d'un processeur,
- d'un coprocesseur,
- d'une mémoire,
- d'un module de routage des communications,
- d'une M.M.U. (Memory Management Unit),
- et d'un bus.

Le processeur est un INTEL 30386 avec comme coprocesseur un WEITEK 1167. Il permet d'obtenir une performance utilisable de 0,5 MFlops par noeud. La mémoire est dans cette configuration de 12 Moctets par noeuds. Le module de

communication, appelé DCM (Direct Connect Module), sert à gérer les échanges de messages avec les autres processeurs d'une part, et d'autre part à rediriger les messages qui transitent par le processeur considéré. Ceci a l'avantage de permettre qu'un message qui transite par un noeud n'affecte pas la vitesse des calculs exécutés localement. Il a de plus la capacité d'émettre sur tous les liens en même temps. Ceci fait qu'un message qui transite sur un noeud n'affecte une émission ou une réception locale que si c'est le même lien qui est concerné. La seule restriction sur le parallélisme au niveau de l'émission de message par un processeur est que le bus mémoire n'est effectivement capable d'alimenter qu'un seul lien. La présence de ce DCM permet de considérer en fait que tous les noeuds sont à la même distance. En effet des expériences réalisées à l'ONERA par exemple montrent que le volume de données transférées et le nombre de messages ont beaucoup plus d'influence que la distance parcourue par ces derniers.

III.1.3 Communications : protocoles et performances

Nous allons ici parler des différents protocoles de communication ainsi que de la manière de les programmer. Sur le plan du système il existe 2 protocoles, l'un dit long pour les messages de taille supérieure à 100 octets et l'autre dit court. Le protocole court envoie directement le message précédé d'une entête. Le long envoie d'abord un message court pour prévenir qu'il va envoyer un message d'une longueur donnée, le processeur récipiendaire alloue un buffer de la taille correspondante et dit qu'il est prêt (ou qu'il n'a pas pu allouer) puis le processeur émetteur envoie son message. On voit donc que dans le cas d'un message long le coût d'initialisation va être élevé et que le coût de communication n'est pas linéaire.

Sur le plan de la programmation on a le choix entre deux modes d'échanges de messages : la communication asynchrone et la communication synchrone. La dernière attend pour passer à la suite d'avoir reçu l'accusé de réception du message envoyé, ou que le message attendu soit arrivé, la première continue

dès le message posté. Dans ce cas on a la possibilité de reporter à plus tard la vérification de l'envoi ou de la réception de message.

III.2 Etude du programme

Nous allons maintenant présenter et étudier le programme que nous avons réalisé du point de vue de sa parallélisation. Le programme est constitué de 7 parties principales. Nous notons ici n pour le nombre de degrés de liberté.

- Lire les données. Le nombre d'I.O. est en $\mathcal{O}(n)$.
- Calculer la matrice. Le nombre d'opérations est en $\mathcal{O}(n^2)$. En fait la constante qui intervient ici est très grande et c'est la partie la plus coûteuse en temps de calcul.
- Calculer le second membre. Le nombre d'opérations est en $\mathcal{O}(n)$.
- Résoudre le système linéaire. Grâce aux efforts réalisés sur le préconditionnement, le nombre d'opérations est en $\mathcal{O}(n^2)$.
- Calculer le champ lointain. Le nombre d'opérations est en $\mathcal{O}(nm)$ où m est le nombre de directions dans lesquelles on calcule le champ lointain.
- Stocker les résultats sur disque.Le nombre d'I.O. est en $\mathcal{O}(n)$.

Pour utiliser de manière efficace un ordinateur multiprocesseur il faut exhiber des tâches qui s'exécutent en parallèle et qui réalisent le minimum de calculs redondants. Mais ici du fait de la mémoire distribuée, il faut aussi se poser la question de la répartition des données. Nous allons donc d'abord parler de cette répartition et des choix qui y ont présidé. Nous présenterons ensuite dans le détail les algorithmes effectivement utilisés pour certaines parties spécifiques du code.

III.2.1 Description du programme

Deux raisons principales nous ont conduit à étudier d'abord la phase de calcul de la matrice. D'une part c'est la partie qui coûte le plus cher en temps de calcul, comme on l'a vu précédemment. Il importe donc, si l'on veut tirer parti d'un multiprocesseur, de s'assurer que cette phase est bien parallélisée. D'autre part c'est la partie qui génère le plus gros volume de résultats, et à ce titre il faut s'assurer que ceux-ci sont répartis sur les mémoires des différents processeurs de manière équilibrée si on veut pouvoir traiter de grosses applications sans trop générer de migrations de données entre processeurs.

Cette partie du programme se décompose en deux phases. D'une part le calcul des matrices élémentaires, et l'assemblage de celles-ci dans la matrice globale d'autre part. Ces deux phases se mélangent de la manière suivante. Pour un numéro de facette K fixé on calcule l'intéraction de K avec toutes les facettes. Ceci est complétement parallélisable. Puis on introduit cette bande de matrice dans la matrice globale. Cette phase demande de prendre des précautions afin d'être parallélisée. On réitère ensuite en prenant la facette suivante. Ici la structure qui génère le moins de calculs redondants est celle qui consiste à répartir de manière équilibrée les facettes K sur les différents processeurs. La matrice globale est pour sa part répartie de façon que les produits matrice-vecteur soient les plus rapides possible. Pour cela elle est distribuée par bandes horizontales de degrés de liberté sur les processeurs. En effet, cela permet de réaliser les produits hermitiens localement puis de n'assembler sur le réseau qu'un scalaire. Si nous avions eu des degrés de liberté répartis sur plusieurs processeurs à la fois (c'est ce qui arrive dans le cas d'une partition par facettes) nous aurions été obligés de d'abord assembler le vecteur puis de faire le calcul global sur tous les processeurs. Dans ce cas, il y aurait eu à la fois plus de données échangées et plus de calculs réalisés. Sur chacun de ceux-ci coexiste donc deux listes, celle des facettes pour la phase de calcul des matrices élémentaires et celle des degrés de liberté qui seront gardés localement. On a fait le choix que la

deuxième est une sous-liste de la liste des degrés de liberté contenus dans les éléments de la première, ceci afin de ne pas avoir une phase de transferts trop importante.

Nous allons maintenant brièvement parler de la phase de lecture des données et du stockage des résultats sur disque. En fait ces deux phases sont similaires. La première comporte une lecture de l'ensemble des données par le noeud 0 (celui par qui passent tous les I/O) puis la distribution de celles-ci sur tous les noeuds, la seconde commence par rassembler les résultats sur le noeud 0 puis les écrit sur disque.
Il reste enfin à parler de la phase de résolution qui, après la phase de calcul de la matrice, est celle qui prend le plus de temps. Dans cette partie du programme la matrice est donc divisée en bandes de degrés de liberté et les vecteurs sont décomposés en sous-vecteurs locaux. Ces répartitions correspondent à une même partition de l'ensemble des degrés de liberté. L'algorithme étant celui exposé précédemment, on voit clairement que tous les calculs sont locaux sauf les produits hermitiens et les produits matrice-vecteur.

III.2.2 Parallélisation de produit hermitien

Nous allons maintenant regarder la parallélisation du produit hermitien de deux vecteurs et les différentes options envisageables. Nous allons commencer par envisager que les vecteurs sont répartis par processeur en une partition; c'est à dire qu'un degré de liberté est sur un processeur et un seul. Ceci n'est pas un choix obligatoire mais on verra tout à l'heure qu'il est de loin le plus intéressant. Pour réaliser un produit hermitien on effectue d'abord celui-ci localement sur chaque processeur, puis il s'agit d'assembler ce résultat sur chacun des processeurs, afin que le produit global soit connu partout. Nous allons donc présenter l'algorithme ADEA (Alternate Directions Exchange Algorithme). Celui-ci est assez classique et est illustré sur la figure III.7 dans le cadre d'un cube à 3 dimensions. Nous allons écrire l'algorithme en pseudo-code. Il s'exécute sur tous les processeurs.

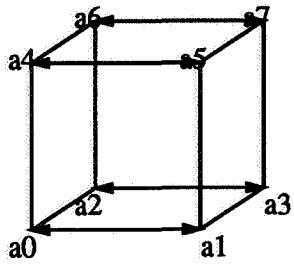
La valeur à assembler se nomme ici s.

```
      do 1 i=0,nombre_de_dimensions-1
        ivois=le processeur voisin dans la direction i
        Envoyer vers ivois la valeur de s de maniere asynchrone
        attendre de ivois une valeur qui sera mise dans s_tampon
            de maniere synchrone
        s'assurer que le message envoye est parti
        s=s+s_tampon
1     continue
```

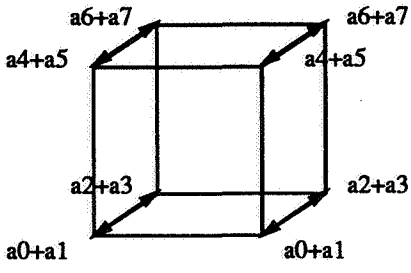
Cette présentation appelle quelques commentaires.

- l'envoi de s en mode asynchrone permet de masquer cette communication par la suivante. On ne paye alors que le temps d'une communication.
- Il n'est pas nécéssaire de se mettre en attente avant d'envoyer car si le message que l'on n'attend pas encore arrive, la machine le met en attente. On ne risque donc pas de le perdre.
- la vérification du départ est impérative avant d'accumuler dans s, sinon on risque d'envoyer une valeur déjà modifiée.
- Ici il n'est pas bénéfique d'attendre de manière asynchrone. En effet le gain obtenu par le masque d'une partie de l'attente par la première vérification est perdu par le débranchement dû à la seconde vérification.

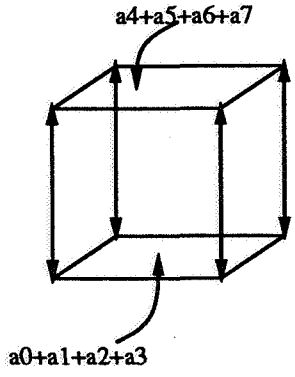
Evaluons maintenant la vitesse de calcul qui résulte de cet algorithme. Nous allons noter *ndim* la dimension de l'hypercube, *nlong* la longueur globale des vecteurs à multiplier. On supposera que celle-ci est un multiple du nombre de processeurs *nproc*. Pour les temps de communication on a un coût d'initialisation τ_s et un vitesse de transmission V_{comm} en octets par seconde. Pour les temps de



Echange dans la direction 0



Echange dans la direction 1



Echange dans la direction 2

Figure III.7 : ADEA

calcul on a une vitesse de calcul de V_{cal} en nombre d'opérations flottantes par seconde.

On va d'abord évaluer le temps de calcul. Le nombre d'opérations à effectuer sur chaque processeur est $8 * \frac{nlong}{nproc}$, alors $T_{cal} = \frac{8 * nlong}{V_{cal} * nproc}$. Le 8 est là car un + et un x complexes font huit opérations flottantes, le 16 parce qu'un complexe en double précision occupe 16 octets. De même calculons le temps de communications. $T_{comm} = ndim * (\tau_s + \frac{16}{V_{comm}})$. Le temps global est donc $T = T_{comm} + T_{cal}$ soit $T = ndim * (\tau_s + \frac{16}{V_{comm}}) + \frac{8 * nlong}{V_{cal} * nproc}$. La vitesse obtenue est alors

$$V_{glob} = 8 * \frac{nlong}{T} = \frac{8 * nlong}{ndim * (\tau_s + \frac{16}{V_{comm}}) + \frac{8 * nlong}{V_{cal} * nproc}}$$
$$= V_{cal} * nproc * \frac{1}{1 + ndim * \frac{nproc}{nlong} \frac{V_{cal}}{8} (\tau_s + \frac{16}{V_{comm}})}$$

On va maintenant calculer l'efficacité du parallélisme. Pour ceci on va comparer la vitesse de calcul obtenue avec celle qui consiste à multiplier la vitesse de calcul d'un processeur par le nombre de processeurs. Nous allons noter $C = \frac{V_{cal}}{8}(\tau_s + \frac{16}{V_{comm}})$. C'est une constante qui ne dépend que du hardware de la machine. Nous notons aussi $nloc = \frac{nlong}{nproc}$ la longueur des vecteurs par processeur. Alors le coefficient E d'efficacité vaut

$$E = \frac{V_{glob}}{V_{cal} * nproc} = \frac{1}{1 + C * \frac{ndim}{nloc}}$$

On définit maintenant la longueur locale de demi-efficacité $nloc_{1/2}$ comme étant la valeur de nloc pour laquelle l'efficacité vaut 1/2. Ici $nloc_{1/2} = ndim * C$. Ceci signifie que la taille par processeur nécessaire pour obtenir une efficacité donnée varie linéairement en fonction de la dimension de l'hypercube.

Si on avait voulu maintenir la possibilité que certains degrés de liberté soient partagés pour plusieurs noeuds alors les choses se seraient passées différemment.

En effet faire les produits localement n'aurait pas eu de sens. Il aurait donc fallu d'abord assembler le vecteur global puis réaliser dans chaque processeur le produit hermitien global, cela aurait coûté très cher en temps de calcul et en temps de communication. Cependant cette manière de voir aurait eu un intérêt: en effet lors du calcul de la matrice il n'aurait plus été nécessaire de redistribuer les lignes et on aurait économisé cette phase de communication. Néanmoins les expériences numériques nous ont conduit à rejeter cette idée.

III.2.3 Parallélisation du produit matrice-vecteur

Nous allons maintenant montrer comment on peut paralléliser le produit matrice-vecteur. Ceci se fait de manière similaire au produit hermitien. En effet c'est encore un schéma ADEA qui est utilisé. Dans un premier temps on assemble les vecteurs locaux dont un vecteur global tampon puis on réalise localement les produits des matrices bande par ce vecteur. On obtient alors le vecteur résultat local. Précisons cela en pseudo-code. Z1_loc sera le vecteur local initial, Z2_loc le vecteur résultat local, A_loc la matrice locale, Z1_tampon et Z2_tampon deux vecteurs tampons, et N_loc le tableau des numéros globaux des degrés de liberté locaux.

```
assembler Z1_loc dans Z1_tampon :
          Z1_tampon(N_loc(i))=Z1_loc(i)
do 1 i=0,nombre_de_dimensions-1
  ivois=le processeur voisin dans la direction i
  attendre de ivois son Z1_tampon  de maniere asynchrone
        pour les mettre dans Z2_tampon
  Envoyer vers ivois Z1_tampon de maniere
        asynchrone
  s'assurer que le message attendu est arrive
  s'assurer que le message envoye est parti
  Z1_tampon=Z1_tampon+Z2_tampon
```

```
1       continue
        Z2_loc=A_loc*Z1_tampon
```

On voit que l'algorithme est fortement semblable au précédent. Faisons une évaluation de la vitesse de calcul. On garde les mêmes notations que pour le produit hermitien. Le nombre d'opérations par processeur est ici de $8 * nlong * \frac{nlong}{nproc}$. Alors $T_{cal} = \frac{8 * nlong^2}{V_{cal} * nproc}$.
De même le temps de communication $T_{comm} = ndim * (\tau_s + 16 * \frac{nlong}{V_{comm}})$. Alors le temps total vaut $T = \frac{8 * nlong^2}{V_{cal} * nproc} + ndim * (\tau_s + 16 * \frac{nlong}{V_{comm}})$, et la vitesse obtenue vérifie

$$\begin{aligned} V_{glob} &= \frac{8 * nlong^2}{\frac{8 * nlong^2}{V_{cal} * nproc} + ndim * (\tau_s + 16 * \frac{nlong}{V_{comm}})} \\ &= V_{cal} * nproc * \frac{1}{1 + ndim * (\tau_s + \frac{16 * nlong}{V_{comm}}) * \frac{V_{cal} * nproc}{8 * nlong^2}} \\ &= V_{cal} * nproc * \frac{1}{1 + \frac{ndim}{nproc} * \frac{1}{nloc^2} * \frac{\tau_s * V_{cal}}{8} + ndim * \frac{1}{nloc} * \frac{2 * V_{cal}}{V_{comm}}} \end{aligned}$$

Le coefficient d'efficacité vaut alors

$$E = \frac{1}{1 + \frac{ndim}{nproc} * \frac{1}{nloc^2} * C_1 + ndim * \frac{1}{nloc} * C_2}$$

où les coefficients C_1 et C_2 sont des constantes qui ne dépendent que de la machine. Alors pour un grand nombre de processeurs $nloc_{1/2} \sim C_2 * ndim$. C'est le même ordre de grandeur que pour le produit hermitien sauf que la constante est dans ce cas plus petite.

Nous allons montrer maintenant un autre algorithme qui est bien plus efficace asymptotiquement. Dans l'algorithme précédent on communiquait à chaque fois une vecteur global alors que ça n'était pas forcément nécessaire. En effet cela

permettait d'éviter au processeur récepteur de se demander où ranger les degrés de liberté qui arrivaient puisqu'ils étaient déjà prérangés dans un vecteur global. Une autre solution serait de passer avec les données le tableau de leurs adresses. C'est cette solution que nous allons présenter ici avec les mêmes notations que ci-dessus. On rajoute cependant deux tableaux tampons pour les entiers, N1_tampon et N2_tampon.

```
      mettre Z1_loc dans Z1_tampon :
                Z1_tampon(i)=Z1_loc(i)
      mettre N_loc dans N1_tampon :
                N1_tampon(i)=N_loc(i)
       do 1 i=0,nombre_de_dimensions-1
         ivois=le processeur voisin dans la direction i
         attendre de ivois de maniere asynchrone des donnees
               pour les mettre dans Z2_tampon et N2_tampon
         Envoyer vers ivois Z1_tampon et N1_tampon de maniere
               asynchrone
         s'assurer que les messages attendus sont arrives
         mettre Z2_tampon a la fin de Z1_tampon
         mettre N2_tampon a la fin de N1_tampon
1      continue
       assembler Z1_tampon dans Z2_tampon :
                Z2_tampon(N1_tampon(i))=Z1_tampon(i)
       Z2_loc=A_loc*Z2_tampon
```

Nous notons qu'il n'est pas nécessaire de s'assurer que les messages envoyés sont partis puisqu'en mettant les données qui arrivent après les données envoyées on ne modifie pas celles-ci. Ici le volume de données transférées dépend de la dimension dans laquelle on transmet. En effet ce volume double à chaque fois si tous les processeurs ont le même nombre de degrés de liberté, ce que nous supposons. Le temps dédié au calcul est le même qu'au dessus. Pour le temps

de communications on a

$$T_{comm} = \sum_{i=0}^{ndim-1} (\tau_s + \frac{nlong}{nproc * V_{comm}} * 2^i * 24) \leq ndim * \tau_s + nlong * \frac{24}{V_{comm}}.$$

Alors comme précédemment on calcule la vitesse globale.

$$V_{glob} = \frac{V_{cal} * nproc}{1 + \frac{\tau_s * V_{cal}}{8} * \frac{ndim}{nproc * nloc^2} + \frac{3 * V_{val}}{V_{comm}} * \frac{1}{nloc}}.$$

de même l'efficacité vaut

$$E = \frac{1}{1 + C1 * \frac{ndim}{nproc * nloc^2} + C2 * \frac{1}{nloc}}.$$

Ici encore C1 et C2 sont des constantes qui ne dépendent que de la nature de la machine sur laquelle on passe. Regardons maintenant l'interprétation que l'on peut faire des constantes C1 et C2. La dernière est le rapport de la vitesse de calcul sur la vitesse de communication. Elle est d'autant plus grande que les processeurs sont rapides ou les communications lentes. De même C1, qui est le produit du startup des communications par la vitesse des processeurs, est d'autant plus grande que les processeurs sont rapides ou le coût d'initialisation des communications fort. On a alors la longueur locale de demi-efficacité qui vérifie $nloc_{1/2} \sim C2$ lorsque la dimension tend vers l'infini. Ce résultat signifie que l'efficacité est maintenue pour un volume de données constant par processeur. Ceci est d'une performance nettement plus grande que l'algorithme précédent où il fallait augmenter la charge des processeurs lorsque l'on voulait maintenir l'efficacité en augmentant la dimension de l'hypercube. Ceci ne pouvant évidemment pas se faire indéfiniment à cause de la taille finie de la mémoire de chacun des processeurs.

III.2.4 Parallélisation du calcul de la matrice

Ici aussi on décompose l'algorithme en une phase de calcul où sont évaluées les matrices élémentaires d'interaction entre triangles et une phase de commu-

nication où sont échangées des lignes de matrices. Avant de préciser le détail de l'algorithme de communication donnons quelques notations. Sur chaque processeur sont présentes plusieurs listes.

- liste 1: La liste des degrés de liberté correspondant aux lignes de la matrice qui sont calculées localement
- liste 2: liste des degrés de liberté qui doivent rester présents sur le processeur une fois la phase de transfert terminée. On a décidé pour réduire cette phase que la liste 2 est une sous-liste de la liste 1.
- Corres: Tableau de correspondance entre numéros locaux et globaux pour les degrés de liberté. Corres(indice_local)=indice_global.
- liste 3: la liste, pour chaque degré de liberté de la liste 2, des processeurs qui contiennent ce degré de liberté dans leur liste 1.
- liste 4: La liste contenant pour chaque degré de liberté de (liste 1)-(liste 2) le numéro du processeur l'ayant dans sa liste 2.

Voyons maintenant en pseudo-code l'algorithme utilisé. Nous ne mentionnons pas la phase de calcul.

```
C$$$ Envoi des lignes de liste 1 - liste 2,
      Pour tout i dans (liste 1)-(liste 2)
        envoyer la ligne i en mode asynchrone
        au processeur liste 4(i).
C$$$ Attente de reception
      Pour tout i de liste 2,
        Pour chaque processeur de liste 3 ayant i dans sa liste 1
           attendre en mode synchrone la ligne correspondante
      Fin
```

Cet algorithme n'est pas optimal. En effet, on pourrait penser que faire les attentes en mode asynchrone et s'assurer qu'elles sont arrivées à la fin permet de gagner un peu de temps. Cependant, les lignes sont envoyées avant la mise en attente et si un processeur est bloqué par l'attente d'une ligne qui n'est pas encore là, néanmoins les autres lignes continuent à arriver et sont mises dans des buffers temporaires par le système d'exploitation du processeur. Ceci fait que l'espérance de gain n'est pas très grande. Notons qu'il ne sert à rien de s'assurer que le message envoyé est parti puisque chaque processeur vérifie que ce qui lui est envoyé arrive. Cette manière a de plus l'avantage d'être une des plus simples qui puissent être écrites.

Evaluons maintenant la performance de cet algorithme. Le temps de calcul $T_{cal} = C_1 \frac{nloc * nlong}{V_{cal}}$. Regardons maintenant le temps de communication. Nous allons d'abord chercher à évaluer le nombre de degrés de liberté qui donnent lieu à un transfert. En fait ce sont des degrés de liberté qui sont partagés par plusieurs processeurs. Sur chaque processeur un majorant de ce nombre peut donc être le nombre de points de la frontière du domaine. Il est asymptotiquement proportionnel à $\sqrt{nloc}$. Nous affirmons maintenant qu'il est licite de dire que le nombre de phases de transfert est majoré par le plus gros temps de réception d'un processeur. En effet comme tous les envois sont faits en première phase et en asynchrone, dans la phase d'attente les processeurs ne se gênent pas. Je néglige ici les problèmes de contention sur le réseau de communication. On a alors $T_{comm} \leq \sqrt{nloc}(\tau_s + \frac{16}{V_comm} * nloc * nproc)$. On peut maintenant donner un minorant de l'efficacité.

$$E > \frac{1}{1 + \frac{1}{\sqrt{nloc}}(\frac{\tau_s}{nproc * nloc} + \frac{16}{V_{comm}})}$$

Pour des systèmes asymptotiquement grands (i.e. avec nproc tendant vers l'infini), on a donc une longueur de demi-efficacité $nloc_{1/2}$ qui est constante en fonction du nombre de processeurs ce qui est un bon résultat. En effet cette phase de calcul étant de loin la plus vorace en temps CPU, il eût été dommage qu'elle

présentât un caractère réfractaire vis à vis de sa parallélisation. Finalement nous voyons que seul le produit hermitien présente cette regrettable caractéristique. C'est dommage, mais d'une part le temps CPU observé pour ce calcul reste convenable et d'autre part nous considérons que c'est le pris à payer pour tous les gains réalisés par ailleurs.

III.3 Résultats d'implémentation

Nous allons donner maintenant quelques résultats qui montrent la vitesse effective des algorithmes que nous avons regardé plus haut. Nous commençons par le produit hermitien. La vitesse de calcul que nous avons observée est de 0,4 Mflops sur un processeur (en 64 bits). Nous avons de plus mesuré que le temps pour transférer un complexe double précision (16 octets) dans une direction est de 0,55 ms. Cette mesure a été faite avec différentes tailles de cube et ce résultat varie très peu. Il passe de 0,55 pour 1 à 3 dimensions à 0,562 pour 4 dimensions et 0,56 pour 5 dimensions, ce qui, compte tenu de la précision de la mesure peut être considéré comme donnant un résultat constant. On a donc comme expression numérique de l'efficacité $E = \frac{1}{1 + 27,5\frac{ndim}{nloc}}$. Ceci fait que pour un 5-cube le nombre de degrés de liberté locaux doit être au moins égal à 138 pour que l'efficacité soit supérieure à 1/2.

Venons maintenant au produit matrice*vecteur. Nous allons présenter dans les tableaux III.1 et III.2 les temps obtenus pour la phase de calcul T_{cal} et pour les deux algorithmes de communication T_{good} pour le plus rapide et T_{bad} pour le plus lent. L'unité est ici la milliseconde. Dans le tableau III.1 * signifie que le test n'a pas pu être réalisé par manque de mémoire sur les processeurs. En effet sur un 5-cube 256 degrés de liberté locaux nécessitent $32 * 256 * 256 * 16$ octets soit plus de 33 Moctets. Remarquons que plus la dimension du cube croît et plus il est nécessaire d'utiliser l'algorithme rapide et de même si la taille locale croît. Regardons maintenant dans le tableau III.3 les efficacités obtenues. On note

nloc	256			128		
	T_{cal}	T_{bad}	T_{good}	T_{cal}	T_{bad}	T_{good}
5-cube	*	*	*	10072	410	53
4-cube	*	*	*	5032	157	28
3-cube	10063	125	26	2516	62	16
2-cube	5039	46	13	1259	23	9

Tableau III.1 : Produit Matrice*Vecteur

nloc	64			32			16		
	T_{cal}	T_{bad}	T_{good}	T_{cal}	T_{bad}	T_{good}	T_{cal}	T_{bad}	T_{good}
5-cube	2516	205	31	630	104	20	160	53	14
4-cube	1260	83	17	319	36	13	79	25	9
3-cube	634	31	11	157	15	8	39	10	7
2-cube	315	11	6	78	6	5	19	5	5

Tableau III.2 : Produit Matrice*Vecteur

E_{good} l'efficacité obtenue par l'algorithme le plus rapide et E_{bad} celle obtenue par l'algorithme le plus lent. On remarque que pour le plus rapide l'efficacité est à peu près constante lorsque la dimension croît avec nloc constant, ce qui avait été prédit dans notre étude. On observe de plus que l'efficacité s'améliore lorsque le nombre de degrés de liberté locaux grandit. Il faut remarquer que l'algorithme le plus lent, quant à lui, se détériore au fur et à mesure que la dimension augmente. On note enfin que malgré tout pour des tailles de matrices acceptables on a une efficacité qui reste proche de 1. Nous allons donner maintenant quelques résultats

nloc	256		128		64		32		16	
	E_{good}	E_{bad}	E_{good}	E_{bad}	E_{good}	E_{bad}	E_{good}	E_{bad}	E_{good}	E_{bad}
5-cube	*	*	0,99	0,96	0,99	0,92	0,97	0,86	0,92	0,75
4-cube	*	*	0,99	0,97	0,99	0,94	0,96	0,90	0,90	0,76
3-cube	1,00	0,99	0,99	0,98	0,98	0,95	0,95	0,91	0,85	0,80
2-cube	1,00	1,00	0,99	0,98	0,98	0,97	0,94	0,93	0,79	0,79

Tableau III.3 : Efficacité

pour le calcul complet. Nous distinguerons d'une part le temps d'assemblage noté T_{ass} et d'autre part le temps de résolution noté T_{resol}. Nit est ici le nombre d'itérations nécessaire pour atteindre la convergence. Ici le critère est fixé à

$\epsilon = 10^{-2}$ et le second membre est, cette fois-ci, une onde plane. Il sera donné à titre de comparaison le temps de calcul obtenu sur un processeur de Cray-2 pour un petit cas de calcul. La fréquence est en Hertz et les temps sont en

Nombre de points	Fréquence	Nit	T_{ass}	T_{resol}
1026	700	16	1011	190
1602	850	14	2491	248
2502	1100	12	6349	470
3602	1200	18	12432	1110

Tableau III.4 : Temps de calcul sur iPSC-2

secondes. A la figure III.8 nous présentons ces résultats sous une autre forme.

Nombre de points	T_{ass}		T_{resol}	
	C-II	5-Cube	C-II	5-Cube
1026	263	1011	5	190

Tableau III.5 : Temps comparés Cray-II vs n-Cube en secondes.

Nous montrons comment évolue le temps de calcul suivant l'augmentation de la taille de maillage. En abcisse nous avons le nombre de points du maillage et en ordonnée le temps de calcul pour l'assemblage et pour une itération divisé par ce nombre de points au carré. De plus, les deux courbes sont normalisées de manière à obtenir 1 pour 1026 points. On peut voir, sur cette figure, que pour l'assemblage la vitesse de calcul reste à peu près constante. Ceci est dû au fait que les triangles sont répartis de manière équilibrée sur les processeurs et que les communications sont bien moins coûteuses que le calcul. Pour une itération, on s'aperçoit au contraire que l'efficacité croît beaucoup lorsque la taille du problème croît. Ceci est dû au fait que dans ce cas les communications ont un poids plus fort, et diminuent d'importance lorsque l'on augmente la taille du problème.

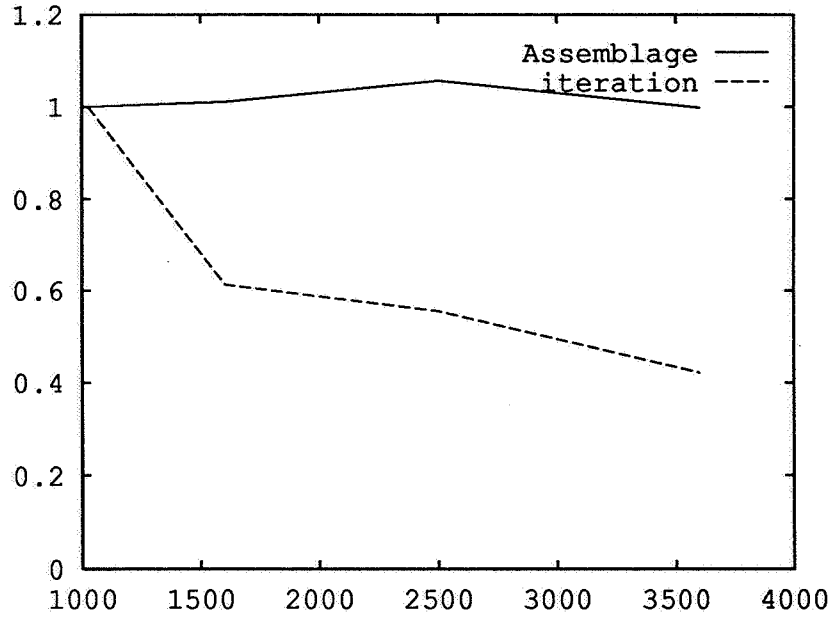

1.2
1
0.8
0.6
0.4
0.2
0
1000
1500
2000
2500
3000
3500
4000
Assemblage
iteration

Figure III.8 :

Deuxième Partie

Amélioration de la méthode par couplage

Introduction

Nous avons pu nous apercevoir dans le chapitre précédent que la principale limitation à la méthode des équations intégrales est la grande place prise en mémoire par la matrice et conjointement le temps important qui est nécessaire pour la calculer. Nous allons développer dans cette partie une amélioration qui va permettre, pensons-nous, d'atteindre des fréquences notablement plus élevées. Pour ceci, nous allons mettre notre objet diffractant à l'intérieur d'une surface de révolution (cf. fig I.1). Nous allons alors résoudre un problème qui couple une formulation d'éléments finis classiques entre l'objet et la surface de révolution avec des équations intégrales sur cette surface qui résument le problème extérieur.

Ce que l'on gagne ainsi est que les équations intégrales sur la surface de révolution donnent lieu à une matrice circulante par blocs. De la sorte, cette matrice peut être mise sous une forme diagonale par blocs par un algorithme de F.F.T.. Ainsi, et la taille de la mémoire occupée par les données, et le temps de calcul se trouvent fortement diminués. Nous allons, dans le premier chapitre, donner une formulation du couplage qui présentera de bonnes propriétés. Dans le chapitre suivant nous montrerons comment on accélère le calcul sur la surface de révolution. Dans le chapitre 3, nous montrerons comment à la fois diminuer encore le nombre de degrés de liberté et résoudre le problème de la singularité qui se trouve aux deux pôles de la surface de révolution.

Chapitre Premier

Formulation du couplage

Nous allons montrer maintenant une formulation du couplage qui présente plusieurs intérêts. Tout d'abord précisons quelques notations. Comme on le voit dans la figure I.1, Ω_0 est le corps diffractant et mathématiquement un ouvert

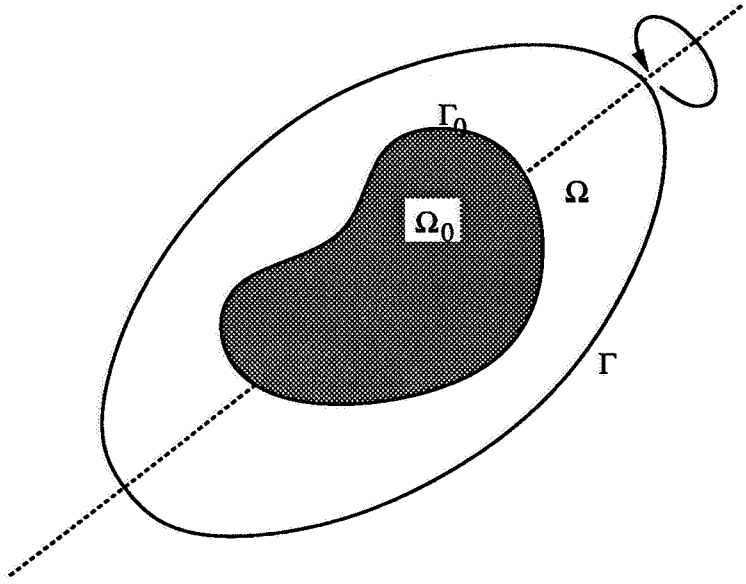


Figure I.1 : Problème couplé

borné régulier de $\mathbb{R}^3$, Γ_0 est son bord, Γ est la surface de révolution qui entoure Ω_0 et Ω est le volume compris entre Γ et Γ_0. Le problème que l'on veut résoudre est : pour g $\in H^{-1/2}(\Gamma_0)$, trouver u dans $H^1_{loc}(\mathbb{R}^3 - \Omega_0)$ vérifiant

$$\begin{cases} \Delta u + k^2 u & = 0 \\ \dfrac{\partial u}{\partial n} & = g \\ \lim\limits_{r\to\infty} r(\dfrac{\partial u}{\partial r} + iku) & = 0 \end{cases}$$

On va le décomposer en un problème intérieur dans Ω et un problème extérieur. Dans Ω on résout $\forall v \in H^1(\Omega)$, $\int_\Omega -\nabla u.\nabla v + k^2 uv + \int_\Gamma \dfrac{\partial u}{\partial n} v - \int_{\Gamma_0} gv = 0$. Pour l'extérieur on va considérer deux inconnues $\alpha \in H^{1/2}(\Gamma)$ et $\lambda \in H^{-1/2}(\Gamma)$. Ces deux inconnues sont la trace et la trace normale de la fonction sur Γ. Alors elles sont liées par les relations suivantes (cf. Chapitre de Rappels).

$$\begin{cases} \dfrac{\lambda}{2} & = D\alpha - K\lambda \\ \dfrac{\alpha}{2} & = K'\alpha - S\lambda \end{cases}$$

Le couplage s'exprime par $\alpha = u_{|\Gamma}$ et $\lambda = \dfrac{\partial u}{\partial n_{|\Gamma}}$. Une façon simple de réaliser ce couplage consiste à substituer λ à $\dfrac{\partial u}{\partial n}$ dans l'équation sur Ω et u à α dans une des deux équations de couplage. Ça n'est pas celle que nous avons retenue. En effet nous effectuons la substitution $\int_\Gamma \dfrac{\partial u}{\partial n} v = \int_\Gamma \dfrac{\lambda}{2} v + \int_\Gamma (Du - K\lambda)v$ dans l'équation dans Ω et nous ajoutons par ailleurs
$\forall \mu \in H^{-1/2}(\Gamma), \int_\Gamma (K' - \dfrac{I}{2})u.\mu - S\lambda\mu = 0.$

Si bien que le sytème obtenu s'écrit :
trouver $(u,\lambda) \in H^1(\Omega) \times H^{-1/2}(\Gamma), \forall (v,\mu) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$,
$$\int_\Omega -\nabla u.\nabla v + k^2 uv + \int_\Gamma (\frac{I}{2} - K)\lambda v + \int_\Gamma Duv + \int_\Gamma (\frac{I}{2} - K')u\mu + \int_\Gamma S\lambda\mu = \int_{\Gamma_0} gv$$

L'intérêt de cette formulation est d'abord d'être symétrique. On peut en effet énoncer la proposition suivante.

Proposition 7 *Notons* $b((u,\lambda),(v,\mu)) = \int_\Omega -\nabla u.\nabla v + k^2 uv + \int_\Gamma (\frac{I}{2} - K)\lambda v + \int_\Gamma Duv + \int_\Gamma (\frac{I}{2} - K')u\mu + \int_\Gamma S\lambda\mu$, *alors* $b((u,\lambda),(v,\mu))$ *est symétrique en (u,λ) et (v,μ).*

Preuve :

Il est en effet notoire, et qui plus est facile à vérifier, que les opérateurs $\frac{I}{2} - K$ de $H^{-1/2}$ dans $H^{-1/2}$ et $\frac{I}{2} - K'$ de $H^{1/2}$ dans $H^{1/2}$ sont les transposés l'un de l'autre. De même, D de $H^{1/2}$ dans $H^{-1/2}$ et S de $H^{-1/2}$ dans $H^{1/2}$ sont autoadjoints. Ceci termine la preuve de la proposition.

De plus, si on note $\mathcal{L}$ l'opérateur de $H^1(\Omega) \times H^{-1/2}(\Gamma)$ associé à la forme bilinéaire symétrique b, on a

Proposition 8 *La partie imaginaire de l'opérateur $\mathcal{L}$ est hermitienne.*
Si $(u, \lambda) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$ et vérifie u est prolongeable dans $H^1_{loc}(\mathbb{R}^3 - \Omega_0)$ hors de Ω en une solution de l'équation de Helmholtz hors de Ω_0 qui satisfasse la condition de radiation, et $\lambda = \frac{\partial u}{\partial n}_{|\Gamma}$
alors $(\mathcal{I}m(\mathcal{L})(u, \lambda), (\bar{u}, \bar{\lambda})) \geq 0$.
Si, de plus, $\mathcal{L}(u, \lambda) = 0$, alors u=0.

Preuve :

la partie imaginaire de $\mathcal{L}$ étant réelle et symétrique, elle est forcément hermitienne.

Pour la positivité, il suffit de remarquer que, lorsque les conditions de la proposition sont satisfaites, $(\mathcal{L}(u, \lambda), (\bar{u}, \bar{\lambda})) = \int_{\Gamma_0} \frac{\partial u}{\partial n} \bar{u}$. Or, la partie imaginaire de cette quantité est exactement le module de l'amplitude limite de u comme nous l'avons vu au chapitre des rappels. Ceci montre à la fois la positivité et, compte tenu du fait que si l'amplitude limite est nulle alors le champ est nul dans tout l'espace, aussi la dernière partie de la proposition.

Chapitre II

Accélération sur un bord axisymétrique

Nous allons, dans ce chapitre, nous attacher à montrer que le calcul des matrices provenant d'équations intégrales et les produits matrices-vecteurs peuvent être accélérés sur une surface de révolution, si l'on utilise un maillage en "quartiers d'orange" (c'est à dire en méridiens et parallèles) comme dans la figure II.1. Nous allons utiliser ici le fait qu'avec un tel maillage, la matrice est circulante

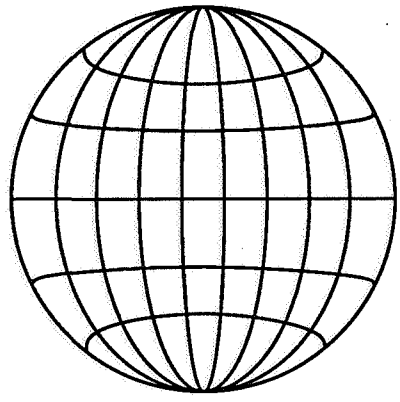


Figure II.1 : Maillage en "quartiers d'orange"

par bloc. Dans un premier temps nous allons donc donner des algorithmes de calcul de la matrice en $\mathcal{O}(N_p^{3/2} \log_2 N_p)$, de produit matrice-vecteur en $\mathcal{O}(N_p^{3/2})$ et d'inversion par méthode directe en $\mathcal{O}(N_p^2)$ où N_p est le nombre de degrés de liberté du maillage de la surface. Puis, nous montrerons que l'on peut très simple-

ment réaliser une implémentation parallèle de ces algorithmes sur un hypercube. En particulier, nous aurons un solveur direct très efficace, ce qui n'est pas toujours le cas sur des systèmes distribués. Nous donnerons enfin des applications numériques sur l'iPSC-II.

II.1 L'algorithme rapide

Sur notre maillage, nous allons distinguer deux types de points. D'une part, les pôles et, d'autre part, les autres points. Ce dernier groupe est divisé en tranches verticales $T_0, T_1, ..., T_{n-1}$ identiques à une rotation près. On se limite ici au cas où n est une puissance de 2. Nous noterons m le nombre de points par tranche. Puisque, dans le cas des équations intégrales, le coefficient d'influence entre deux points x et y ne dépend que du vecteur x-y et est invariant par rotation, la matrice d'équations intégrales sur la surface de révolution est de la forme :

$$\mathcal{M} = \begin{array}{|c|c|c|c|c|} \hline B & A & A & ... & A \\ \hline C & A_0 & A_1 & ... & A_{n-1} \\ \hline C & A_{n-1} & A_0 & ... & A_{n-2} \\ \hline ... & ... & ... & ... & ... \\ \hline C & A_1 & A_2 & ... & A_0 \\ \hline \end{array}$$

On a supposé que les degrés de liberté étaient numérotés en commençant par les deux pôles et en continuant par la tranche T_0 puis T_1, jusqu'à la dernière. Notons que les divers blocs matriciels qui apparaissent sont pleins. On va décomposer $\mathcal{M}$ en 2 matrices. Avant cela nous pouvons déjà remarquer que le calcul de la matrice $\mathcal{M}$ se limite à celui des blocs A, B, C et A_i. Nous pouvons écrire que $\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2$ où

$$\mathcal{M}_1 = \begin{array}{|c|c|c|c|c|} \hline B & A & A & ... & A \\ \hline C & 0 & 0 & ... & 0 \\ \hline C & 0 & 0 & ... & 0 \\ \hline ... & ... & ... & ... & ... \\ \hline C & 0 & 0 & ... & 0 \\ \hline \end{array} \quad \text{et } \mathcal{M}_2 = \begin{array}{|c|c|c|c|c|} \hline 0 & 0 & 0 & ... & 0 \\ \hline 0 & A_0 & A_1 & ... & A_{n-1} \\ \hline 0 & A_{n-1} & A_0 & ... & A_{n-2} \\ \hline ... & ... & ... & ... & ... \\ \hline 0 & A_1 & A_2 & ... & A_0 \\ \hline \end{array}.$$

On va noter $\bar{\mathcal{M}}_2$ la sous matrice de $\mathcal{M}_2$ obtenue en ne prenant pas les degrés de liberté associés aux pôles. La première chose à remarquer est que $\bar{\mathcal{M}}_2$

est circulante par blocs, on notera alors $\bar{\mathcal{M}}_2 = circ(A_0, A_1, ..., A_{n-1})$. Cette constatation nous conduit à vouloir utiliser une méthode F.F.T. par blocs pour accélérer les calculs.

Précisons tout d'abord quelques notations. Soit $(V_i)_{i=0,n-1}$ une suite de vecteurs de dimension m. Nous noterons $\mathcal{F}_n^m(V)_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} V_j e^{\frac{ijk}{2n\pi}}$ la suite transformée de Fourier par blocs de V.

De même, nous définissons $\bar{\mathcal{F}}_n^m(V)_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} V_j e^{\frac{-ijk}{2n\pi}}$ la suite transformée de Fourier inverse par blocs de V.

Notons encore, si $\mathcal{N} = circ(B_0, B_1, ..., B_{n-1})$ où les blocs sont de taille $m \times l$, $F_n^{ml}(\mathcal{N})$ la matrice diagonale par blocs $diag(C_0, C_1, ..., C_{n-1})$, où $C = \sqrt{n}.\mathcal{F}_n^{m.l}(B)$. Nous avons alors le théorème suivant qui est classique.

Théorème 5 *Soit* $\mathcal{N} = circ(B_0, B_1, ..., B_{n-1})$ *et* $V = (V_0, V_1, ..., V_{n-1})$. *Alors,* $\mathcal{N}.V = \mathcal{F}_n^m(F_n^{ml}(\mathcal{N}).\bar{\mathcal{F}}_n^m(V))$.

Nous avons donc un moyen de calculer $\bar{\mathcal{M}}_2.V$.Regardons comment se passent les choses pour $\mathcal{M}_1$. Nous pourrions calculer $\mathcal{M}_1.V$ de manière tout à fait standard, ce qui, compte tenu de son faible remplissage, nous donnerait des temps de calcul tout à fait acceptables. Toutefois, nous allons voir que l'on peut facilement tirer parti du travail fait pour la partie circulante de la matrice.

Proposition 9 *Si A est le bloc qui intervient dans la ligne supérieure de* $\mathcal{M}$ *et de* $\mathcal{M}_1$, *alors* $A(V_0 + V_1 + ... + V_{n-1}) = \sqrt{n}A.\left[\bar{\mathcal{F}}_n^m(V)_0\right]$. *De même, si B est le bloc de la colonne de gauche de* $\mathcal{M}$ *et U un vecteur dont les degrés de liberté sont les deux pôles, alors* $\sqrt{n}\mathcal{F}_n^m \begin{pmatrix} B.U \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix} = \begin{pmatrix} B.U \\ B.U \\ . \\ . \\ . \\ B.U \end{pmatrix}$.

Preuve :

Il est facile de constater que $\sqrt{n}\mathcal{F}_n^m(V)_0 = V_0 + V_1 + ... + V_{n-1}$. De même

$\sqrt{n}\bar{\mathcal{F}}_n^m(V)_0 = V_0 + V_1 + ... + V_{n-1}$. Ceci prouve finalement la proposition.

Ainsi, après transformations, la nouvelle matrice à la structure suivante.

$$\mathcal{M}' = \begin{array}{|c|c|c|c|c|} \hline B & \sqrt{n}A & 0 & 0...0 & 0 \\ \hline \sqrt{n}C & \mathcal{F}(A_i)_0 & 0 & 0...0 & 0 \\ \hline 0 & 0 & \mathcal{F}(A_i)_1 & 0...0 & 0 \\ \hline ... & ... & ... & ... & ... \\ \hline 0 & 0 & 0 & 0...0 & \mathcal{F}(A_i)_{n-1} \\ \hline \end{array}$$

Nous pouvons maintenant donner un algorithme pour calculer un produit matrice-vecteur avec $\mathcal{M}$. Dans le cas d'équations intégrales, les blocs A_i sont des blocs carrés et, de plus, lorsque la fréquence croît on a $m \sim n$. On suppose que la partie circulante de $\mathcal{M}$ à déjà subi une transformation de Fourier. Ceci fait que le coût de calcul de la matrice est en $\mathcal{O}(m^2 n \log n) = \mathcal{O}(n^3 \log n) = \mathcal{O}(N_p^{3/2} \log N_p)$.

Algorithme 1 *On écrit le vecteur à multiplier comme* $(U, V_0, V_1, ..., V_{n-1})$ *et le vecteur résultat comme* $(U', V_0', V_1', ..., V_{n-1}')$. *Alors l'algorithme du produit de* $\mathcal{M}$ *et* $(U, V_0, V_1, ..., V_{n-1})$ *se décompose en trois étapes.*

- *Transformée de Fourier inverse du vecteur initial :* $V = \bar{\mathcal{F}}_n^m(V)$
- *Réalisation des divers produits :*
  - $U' = B.U$
  - $U' = U' + \sqrt{n}A.V_0$
  - $V_j' = \mathcal{F}_n^{m^2}(A_i)_j.V_j$ *pour* $j = 0, ..., n-1$
  - $V_0' = V_0' + \sqrt{n}C.V_0$
- *Transformée de Fourier du vecteur résultat :* $V' = \mathcal{F}_n^m(V')$

Proposition 10 *la complexité de l'algorithme 1 est en* $\mathcal{O}(N_p^{3/2})$ *où* N_p *est le nombre de degrés de liberté du maillage.*

Preuve :
Pour la première et la troisième phase, il s'agit de réaliser m transformées de Fourier. Comme celles-ci sont executées par un algorithme de F.F.T., le coût de ces deux phases est en $\mathcal{O}(mn \log n) = \mathcal{O}(n^2 \log n)$. Pour la deuxième phase, il s'agit juste de calculer des produits matrice-vecteur et le coût est en $nm^2 = \mathcal{O}(n^3)$. Comme $N_p = nm + 2 = \mathcal{O}(n^2)$, on a bien le résultat annoncé.

On peut de même donner un algorithme d'inversion de $\mathcal{M}$ par méthode directe. Il s'agit d'utiliser la décomposition LU par blocs de la transformée de $\mathcal{M}$ après F.F.T.. Le calcul de cette transformée est en $nm^3 = \mathcal{O}(N_p^2)$.

Algorithme 2 *les hypothèses et notations restent les mêmes que pour l'algorithme précédent. Comme précédemment nous avons trois étapes.*

- *Transformée de Fourier inverse du vecteur initial.*
- *Réalisation des diverses inversions des blocs.*
- *Transformée de Fourier du vecteur résultat.*

Ici tous les blocs sont de taille m sauf le premier, qui est de taille m+2, puisqu'il incorpore le couplage avec les pôles.

Proposition 11 *L'algorithme 2 a une complexité en* $\mathcal{O}(N_p^{3/2})$.

Preuve :
le coût des première et troisième phases est le même que pour l'algorithme 1. Pour la deuxième phase nous avons n descentes-remontées à effectuer, son coût est en $\mathcal{O}(nm^2)$. On conclue comme pour la proposition précédente.

On a, de plus, un volume de données à stocker en $\mathcal{O}(N_p^{3/2})$.

II.2 Parallélisation de l'algorithme

II.2.1 Divers algorithmes de F.F.T.

Nous allons ici montrer que ces algorithmes se parallélisent très bien. Commençons par quelques rappels sur les différents algorithmes de F.F.T. qui existent. On pourra se référer pour plus de détails à [Tem76]. Nous allons en détailler deux. Le premier est l'algorithme de Cooley-Tukey (c.f. [CT65]).

La première phase de cet algorithme consiste à réordonner le vecteur entrant suivant l'ordre du Bit Reverse Ordering (B.R.O.). Si $n = 2^l$ est la longueur de la transformée de Fourier à éxecuter, alors tout indice de la suite est compris entre 0 et n-1. Soit i un de ces indices, $i = a_0 a_1 ..... a_{l-1}$ en base 2, alors l'image de i par le B.R.O. est $j = a_{l-1} ..... a_1 a_0$. Nous donnons ci-dessous un exemple pour une F.F.T. de longueur 8.

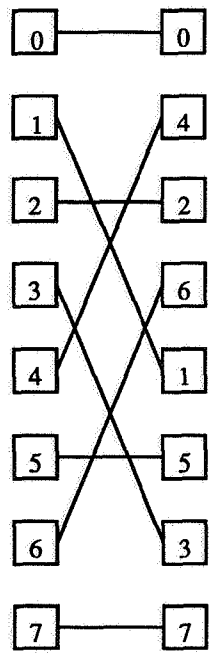


Figure II.2 : Bit Reverse Ordering

Ensuite vient la phase de combinaisons. Plutôt que d'exposer de manière

théorique la façon dont ce comporte l'algorithme, nous allons montrer comment sont réalisés les échanges dans le cas d'une F.F.T. de longueur 8. On pourra, si l'on veut en savoir plus, se reporter à [Tem76]. La signification de l'opérateur

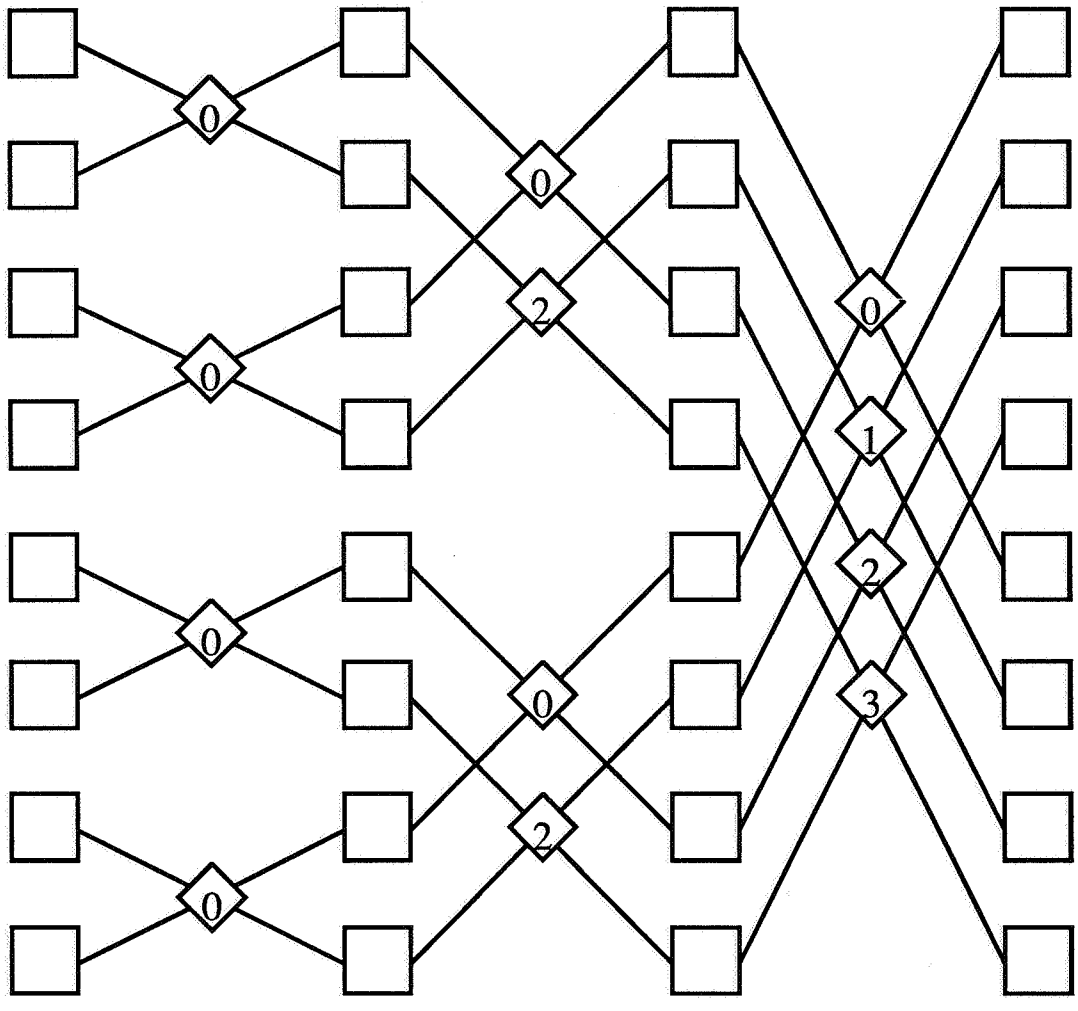


Figure II.3 : Phase de combinaisons de l'algorithme de Cooley-Tukey.

dans la boîte losange de la figure II.3 est donnée à la figure II.4. Le complexe ω vaut ici $e^{\frac{2i\pi}{n}}$ s'il s'agit d'une transformée de Fourier directe et $e^{-\frac{2i\pi}{n}}$ si c'est pour une transformée de Fourier inverse.

Le second algorithme que nous allons regarder est celui de Gentleman-Sande (c.f. [GS66]). C'est en quelque sorte le transposé du précédent. Il commence par une phase de combinaisons et se termine par le Bit Reverse Ordering. Ici

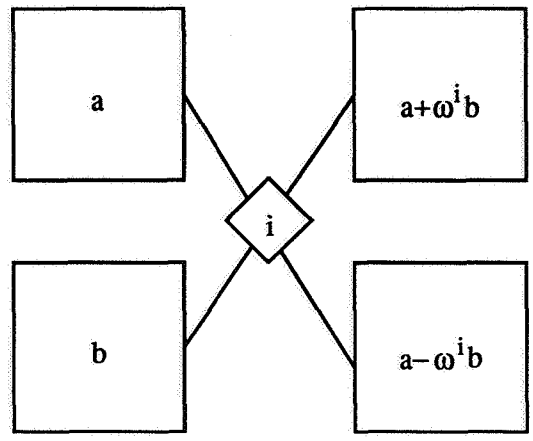


Figure II.4 : Opérateur élémentaire de la F.F.T.

encore, nous allons nous contenter d'illustrer la méthode sur une transformée de longueur 8.

II.2.2 Parallélisation

Avant de paralléliser, faisons la remarque suivante. Aussi bien pour l'inversion par méthode directe que pour le produit matrice-vecteur on peut faire l'économie des phases de B.R.O.. En effet, pour la diagonalisation par blocs de la matrice, Nous allons utiliser uniquement la partie combinaisons de l'algorithme de Gentleman-Sande. Les blocs, au lieu d'être rangés dans le bon ordre seront ordonnés suivant le B.R.O. puisque c'est une permutation involutive. On peut le voir, soit directement sur la formule, soit en remarquant que c'est un produit de transpositions à supports disjoints. Ensuite, pour réaliser un produit Matrice-Vecteur, on applique encore uniquement la partie combinaisons de l'algorithme de Gentleman-Sande. Nous obtenons alors la transformée de Fourier par blocs du vecteur dans l'ordre B.R.O.. On réalise alors les divers produits par bloc. On a alors dans l'ordre B.R.O. la transformée de Fourier du résultat. On va donc appliquer l'algorithme de Cooley-Tukey uniquement dans sa phase de combinaisons puisque le résultat est déjà dans l'ordre B.R.O.. On a donc pu ainsi échapper à la phase de permutation. Pour une inversion directe, le même procédé s'applique.

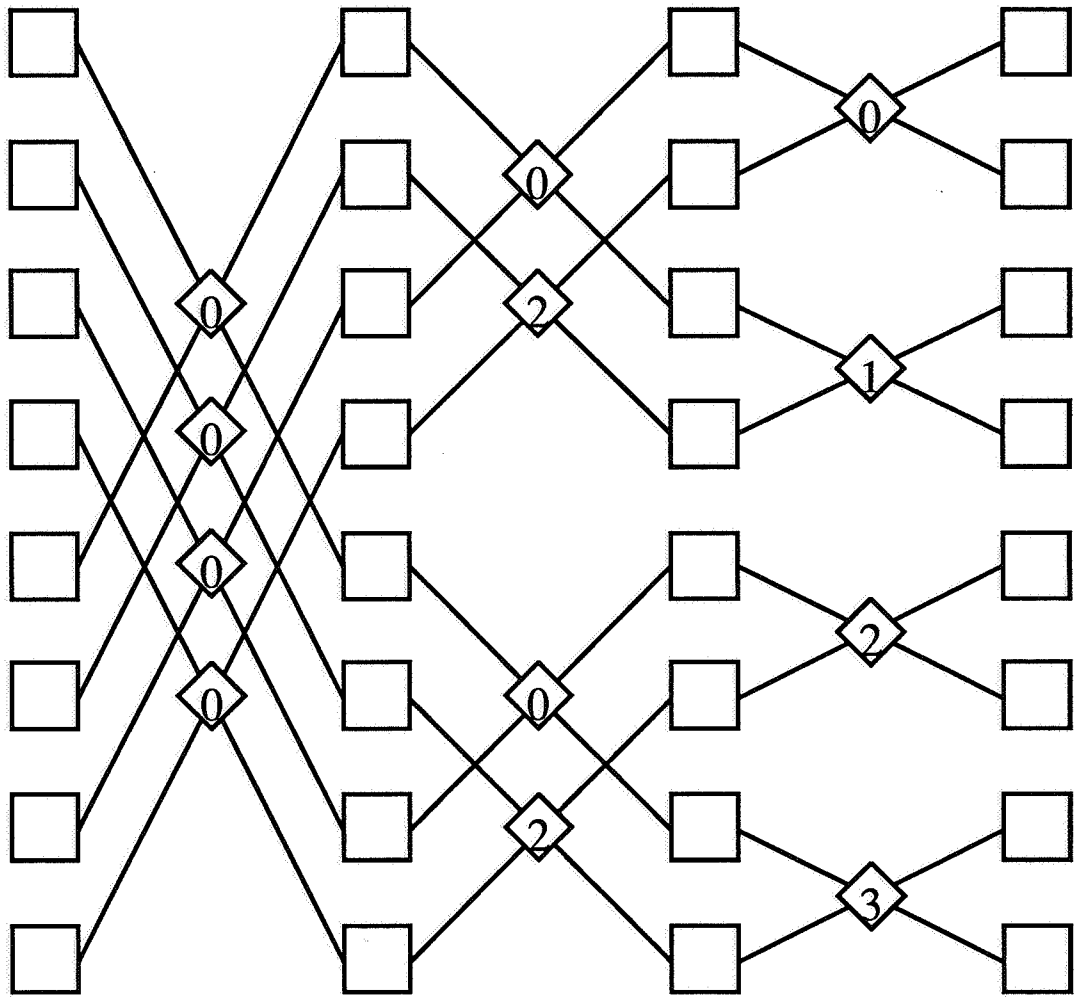


Figure II.5 : Phase de combinaisons de l'algorithme de Gentleman-Sande

En effet, la seule différence est que l'on fait des inversions par blocs à la place des produits par blocs.

Voyons maintenant comment cela peut s'implémenter sur un réseau hypercube. Nous allons d'abord supposer qu'il y a autant de noeuds que de termes pour la transformée de Fourier. Alors on répartit un bloc par processeur, le bloc i allant sur le noeud i. On s'aperçoit que, dans ce cas, le schéma de combinaisons est exactement celui de l'algorithme A.D.E.A. présenté dans la partie précédente (c.f. p. 74). On peut voir à la figure II.6 ce schéma, pour un 3-cube pour

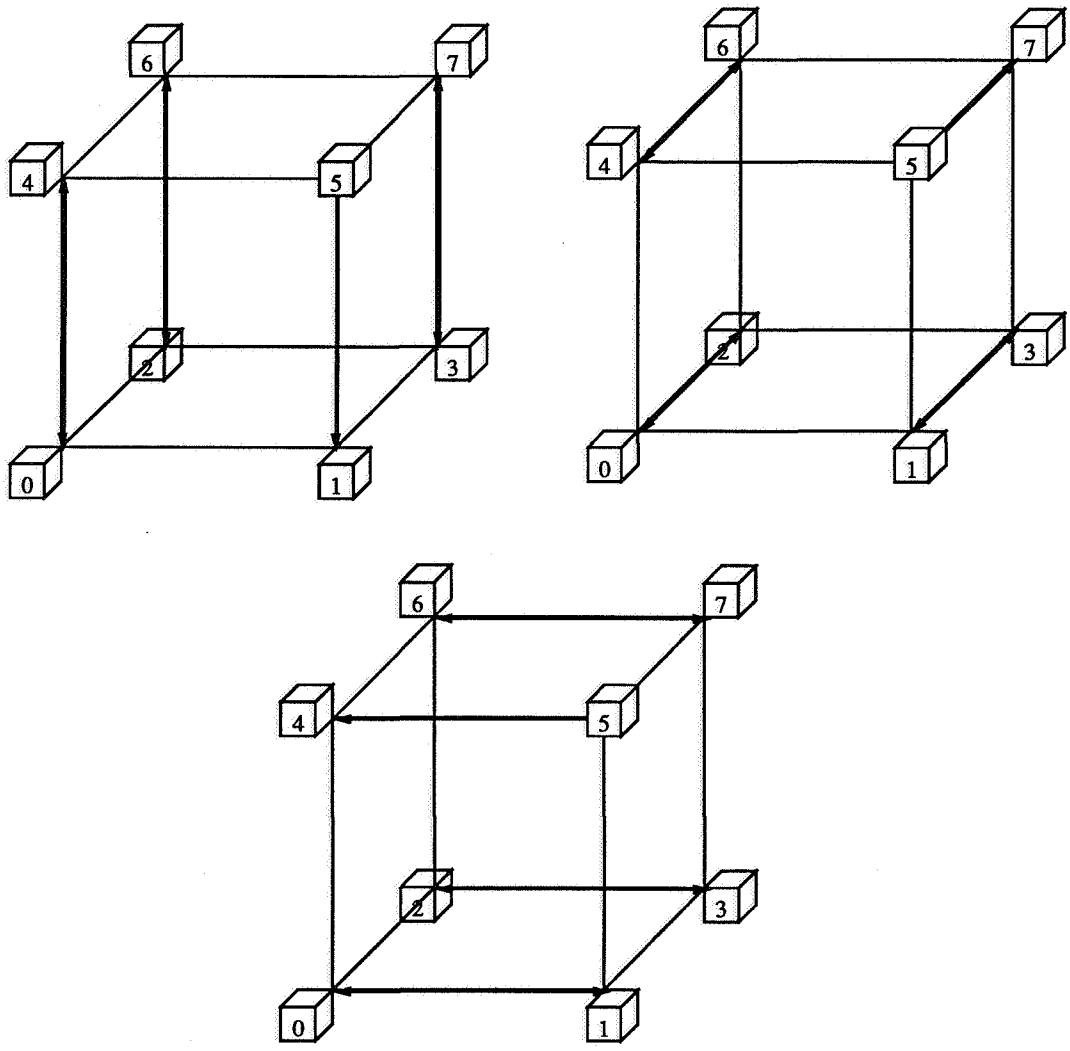


Figure II.6 : Application de l'algorithme de Gentleman-Sande sur un 3-cube

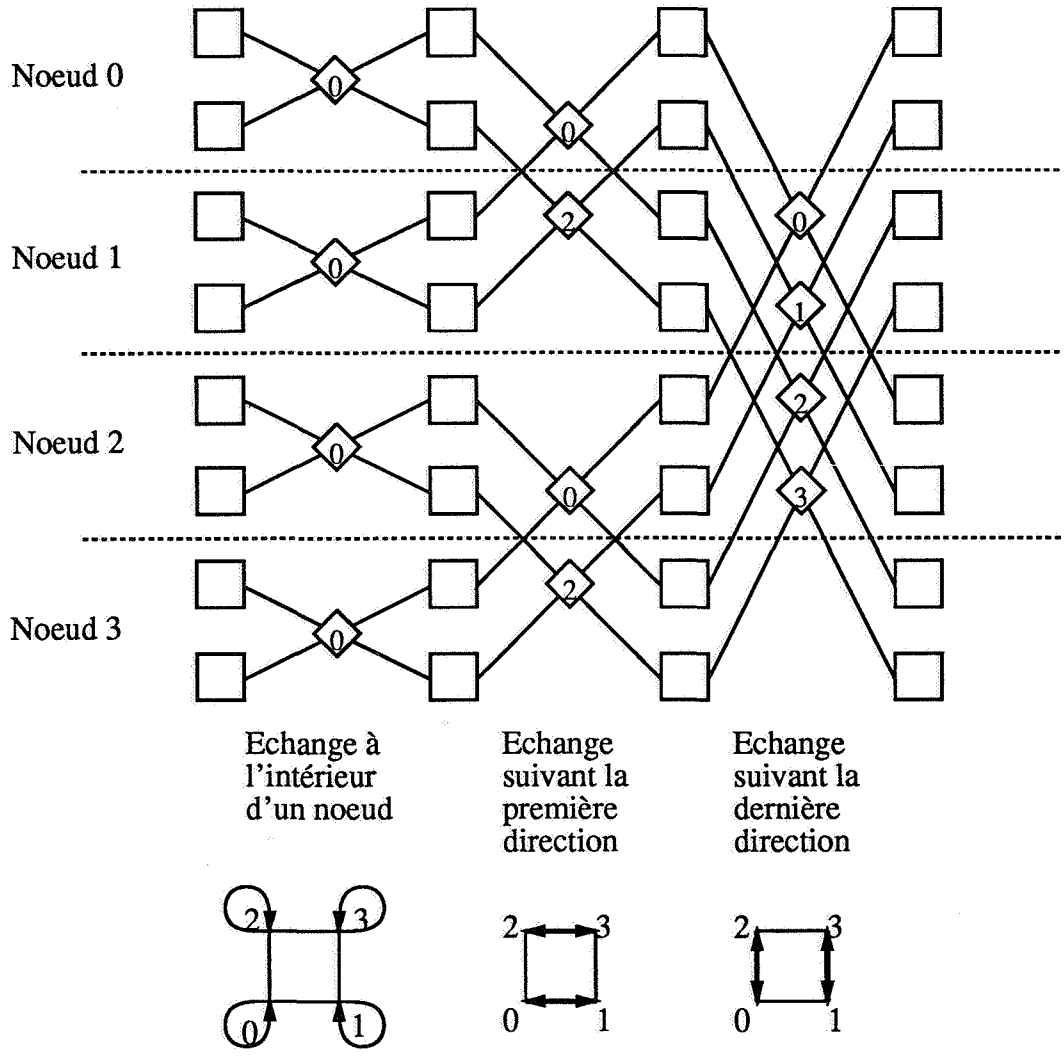


Figure II.7 : Application de l'algorithme de Cooley-Tukey sur un 2-cube pour une F.F.T. de longueur 8

l'algorithme de Gentleman-Sande. Pour celui de Cooley-Tukey, c'est le même type de schéma. Seul change l'ordre des directions suivant lequel on réalise les échanges. Pour Gentleman-Sande on agit dans l'ordre des bits de poids décroissant, alors que pour Cooley-tukey c'est le contraire.

Dans le cas d'une taille de transformée $n = 2^l$ supérieure au nombre $m = 2^d$ de noeuds du cube, on fragmente la liste de blocs en sous-listes de 2^{l-d} blocs

contigus, de manière que le nombre de sous-listes soit le nombre de noeuds du cube. Alors, pour Cooley-Tukey, on commence par réaliser les combinaisons à l'intérieur des noeuds (pour l-d passes) puis on réalise les phases qui combinent des blocs différents (d passes). A la figure II.7 on a un exemple avec une transformée de Fourier de longueur 8 appliquée à un 2-cube.

Pour l'algorithme de Gentleman-Sande on fait la même répartition des blocs. Cependant, on commence par les combinaisons entre blocs qui sont des noeuds différents puis on termine par celles entre blocs sur un même noeud. On en voit une illustration dans le cas d'une transformée de Fourier de longueur 8 sur un 2-cube à la figure II.8.

II.3 Résultats d'expérimentation numérique

Nous allons donner ici les résultats des implémentations sur l'iPSC-2 des algorithmes que nous avons présentés précédemment. Nous allons noter dans toute cette section m le nombre de points par bloc. Nous lions m au nombre de blocs *nblock* par $m = nblock/2$. Ceci signifie que lorsque l'on augmente le nombre de blocs, on augmente aussi le nombre de points par blocs. Cela permet un raffinement du maillage à la fois suivant les parallèles et suivant les méridiens. Le facteur 1/2 correspond au maillage d'une sphère avec le même angle élémentaire sur les méridiens et sur l'équateur. Ainsi, le nombre de degrés de liberté est $2m^2$. Dans le premier tableau nous donnons les temps de calcul en secondes pour

m	128	64	32	16
$5-cube$	43,02	4,72	0,56	0,14
$4-cube$	84,88	9,35	1,02	0,14
$3-cube$	167,25	18,22	1,96	0,25
$2-cube$	*	35,81	3,8	0,42

Tableau II.1 : Temps de calcul de la diagonalisation par blocs.

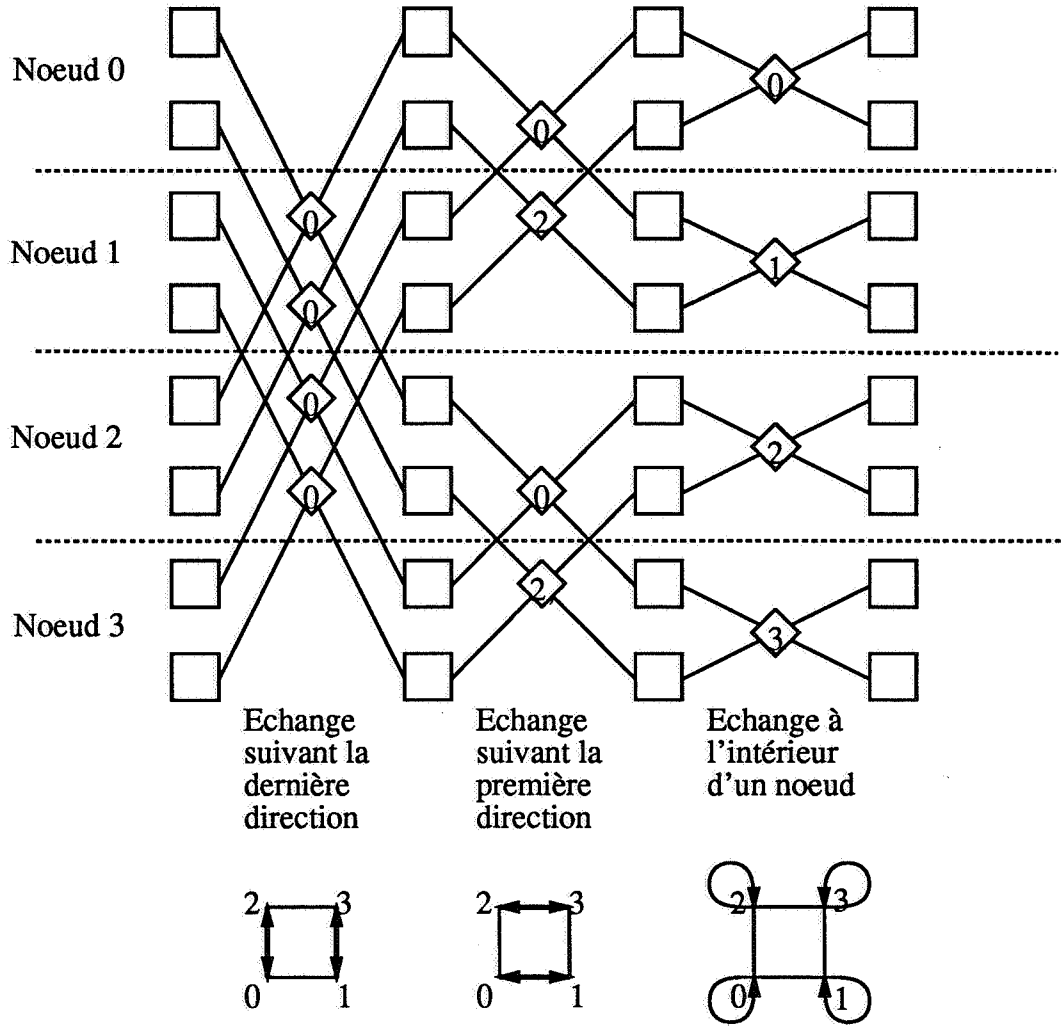


Figure II.8 : Application de l'algorithme de Gentleman-Sande sur un 2-cube pour une F.F.T. de longueur 8

l'algorithme de diagonalisation par blocs de la matrice (* signifie ici que la taille mémoire n'est pas suffisante pour passer ce cas). Nous montrons ensuite dans le tableau II.2 le temps pour le produit matrice-vecteur. Enfin, dans le tableau II.3 nous donnons le temps de factorisation de la matrice. Du fait du plus fort coût de calcul, nous ne présentons pas le cas du 2-cube. Nous allons maintenant nous intéresser d'une part à l'efficacité du parallélisme pour ces algorithmes et d'autre part, à une vérification numérique des calculs de complexité. Nous pouvons tout

m	128	64	32	16
$5-cube$	3,39	0,58	0,14	0,06
$4-cube$	6,59	1,05	0,23	0,08
$3-cube$	12,93	1,99	0,39	0,12
$2-cube$	*	3,8	0,66	0,16

Tableau II.2 : Temps de calcul du produit matrice-vecteur.

m	128	64	32	16
$5-cube$	140,21	8,66	0,56	0,04
$4-cube$	280,05	17,3	1,12	0,08
$3-cube$	561,01	34,6	2,25	0,15

Tableau II.3 : Temps de calcul de la factorisation par blocs.

d'abord remarquer que dans le cas de la factorisation les choses se passent assez bien. En effet, comme il n'y a pas de communication entre les noeuds, l'efficacité vaut 1. De même, on observe que le temps de calcul est en m^4. Pour les courbes d'efficacité que nous allons montrer, nous avons pris comme référence le 2-cube. Dans le cas $m = 128$, c'est le 3-cube. A la figure II.9 on a l'efficacité dans le cas du produit matrice-vecteur. Nous produisons à la figure II.10 l'efficacité de l'algorithme de diagonalisation par blocs. On peut voir sur ces deux figures, que d'une part l'efficacité diminue lorsque la dimension du cube croît, ce qui est normal, mais que dans les deux cas elle est convenable, dès que la taille du problème devient conséquente. Rappelons que pour $m = 64$, il y a 8192 degrés de liberté et que pour $m = 32$, il y en a encore 2048. Ceci est particulièrement vrai pour la diagonalisation par blocs, ce qui est intéressant puisque c'est l'algorithme le plus coûteux des deux.

Nous allons maintenant confronter nos calculs de complexité avec les résultats numériques. Sur la figure II.11 nous avons représenté le temps de calcul du produit matrice-vecteur en fonction de m^3. L'échelle est logarithmique. Le coût

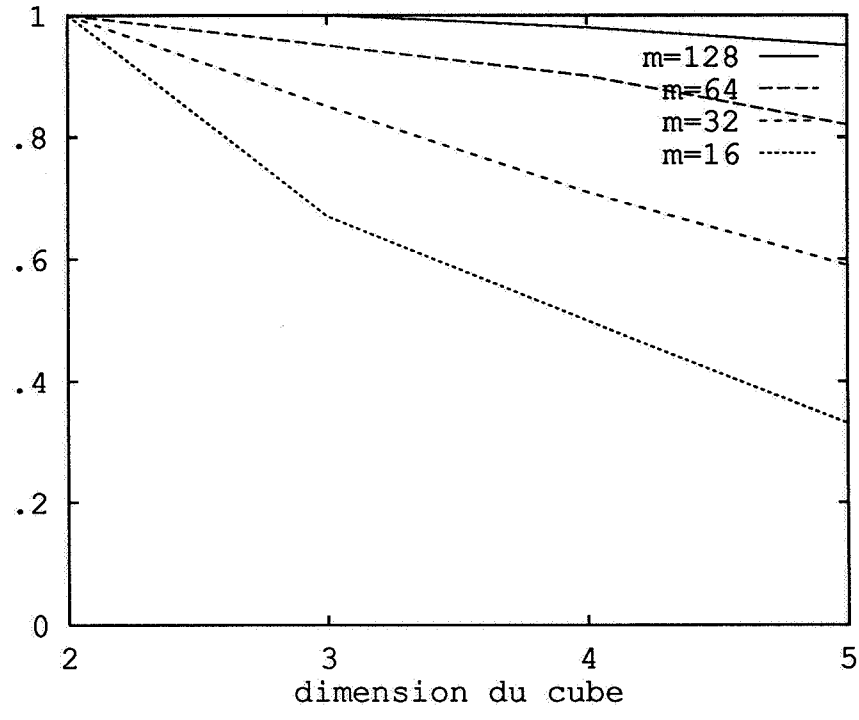


Figure II.9 : Efficacité du produit matrice-vecteur en fonction de la dimension du cube.

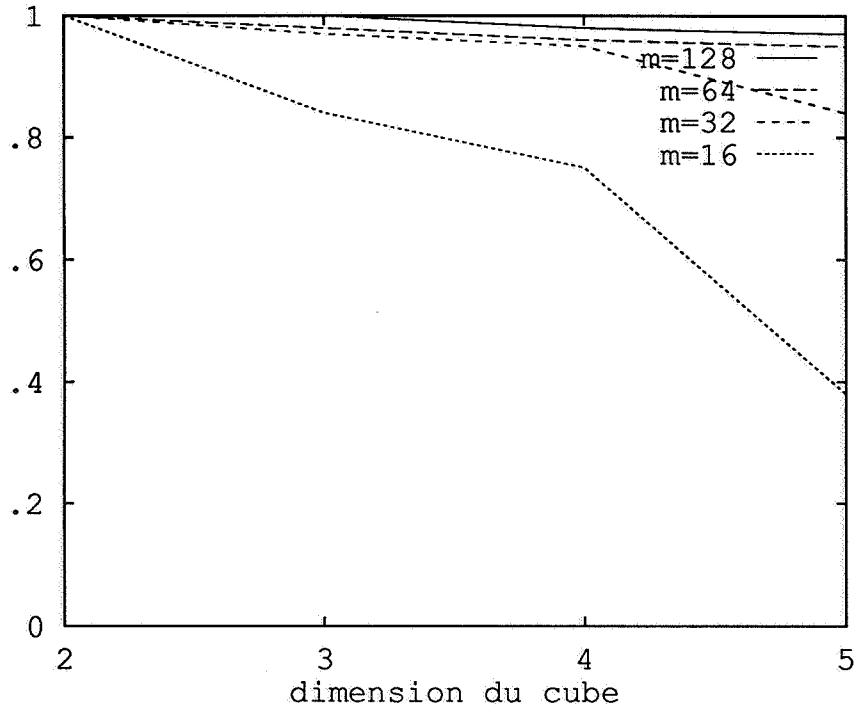


Figure II.10 : Efficacité de la diagonalisation par blocs en fonction de la dimension du cube.

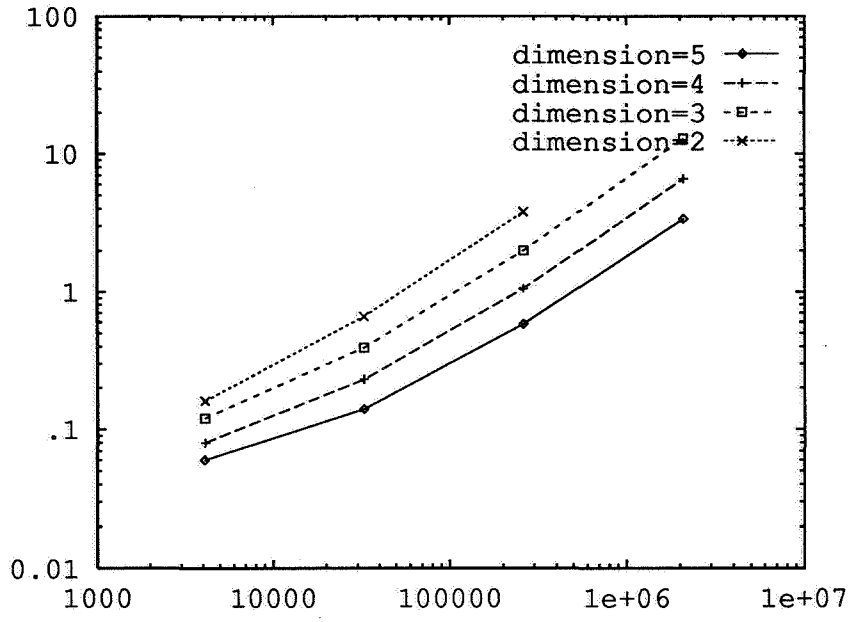


Figure II.11 : Complexité du produit matrice-vecteur.

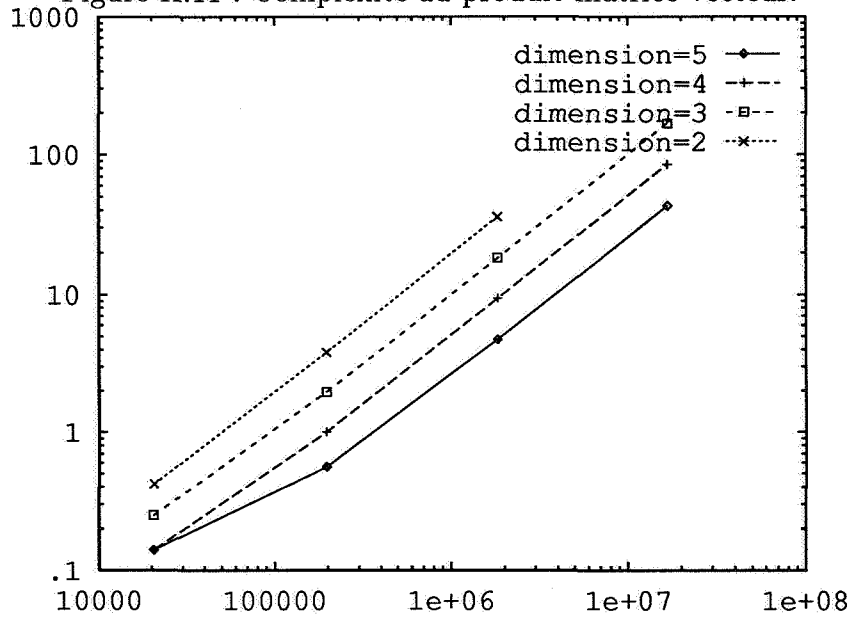


Figure II.12 : Complexité de la diagonalisation par blocs.

de calcul de la diagonalisation par blocs en fonction de $m^3 \log_2 m$ est présenté sur la figure II.12. Sur ces deux figures on se rend compte que les pentes des courbes sont à peu près égales à 1 dans leur partie terminale. Ceci montre que l'estimation que nous avions faite de la complexité de ces deux algorithmes se trouve confortée. Toutefois, dans le cas du produit matrice-vecteur, on peut noter que l'on est moins proche de la pente 1. Cela signifie que le coût des F.F.T. (en $m^2 \log_2 m$) n'est pas encore complètement négligeable vis à vis des produits par blocs. Ceci est surtout dû au poids des communications entre processeurs dans les transformées de Fourier.

Chapitre III

Amélioration du traitement des pôles

La méthode proposée précédemment comporte un inconvénient de taille pour pouvoir augmenter la fréquence des ondes incidentes dont on souhaite observer la diffraction par un objet donné. En effet, plus elle croît et plus on doit raffiner le maillage. Ceci a pour effet d'aplatir de plus en plus les triangles au voisinage des pôles, ce qui a deux conséquences néfastes. D'une part le conditionnement est de ce fait détérioré, d'autre part il en résulte une perte de précision. Nous allons donc développer une méthode qui transforme le maillage qui était utilisé dans la section précédente autour de l'objet diffractant, de manière à diminuer le nombre d'éléments au voisinage des deux pôles. Toute la difficulté consiste à faire ceci tout en conservant une certaine structure au maillage, de manière à ne pas perdre la rapidité obtenue plus haut. Nous allons commencer par exposer l'algorithme qui nous permet de constuire le maillage qui nous intéresse. Ensuite, nous montrerons comment nous faisons pour calculer la matrice d'équations intégrales et pour réaliser les produits matrice-vecteur ou les inversions par méthode directe. Enfin, nous donnerons quelques propriétés de la méthode.

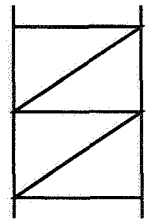
III.1 L'algorithme de maillage

On suppose que l'axe qui passe par les pôles est vertical pour la commodité de l'explication.

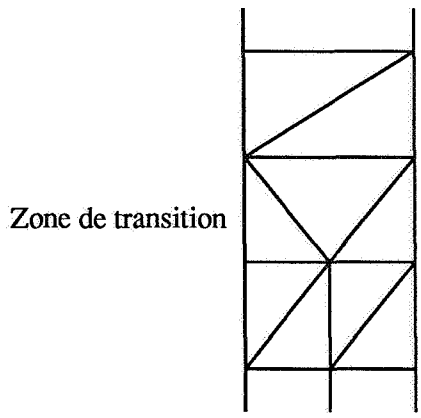
L'algorithme qui construit le maillage est le suivant.

- Mailler le voisinage des pôles en quatre triangles.

- Tant que la largeur d'une maille du maillage est inférieure à un critère donné (par exemple $h < \frac{\lambda}{5}$), mailler chaque bande verticale (i.e. chaque quartier) en triangles sans augmenter le nombre de mailles dans le sens des "parallèles".



- Si la largeur des triangles créés devient supérieure au critère de qualité retenu, on double le nombre de bandes en divisant chacune en deux.



Nous pouvons voir dans les figures III.1, III.2 et III.3 plusieurs exemples de tels maillages pour des sphères.

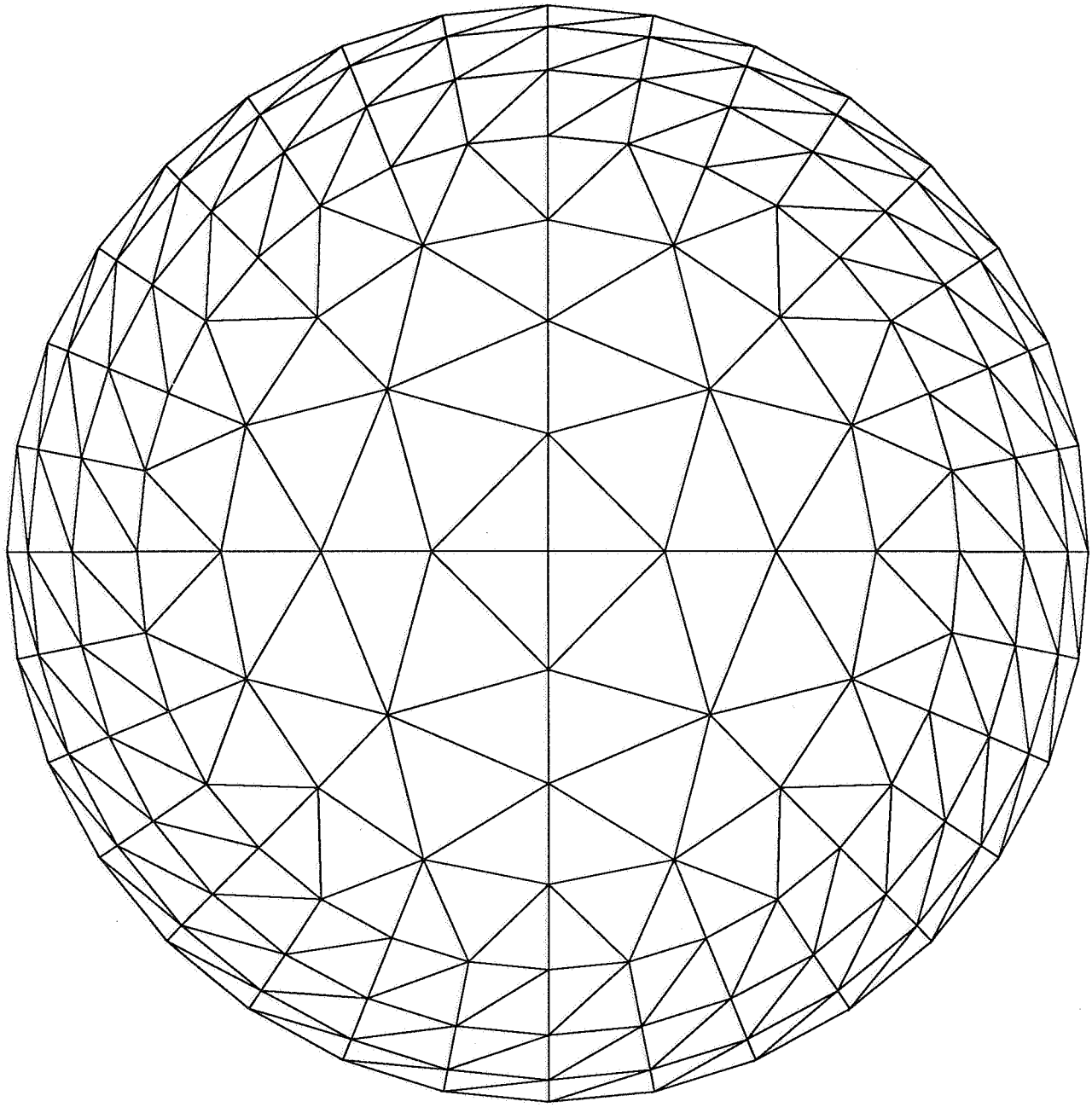


Figure III.1 : Vue du pôle d'un maillage de sphère avec 688 éléments

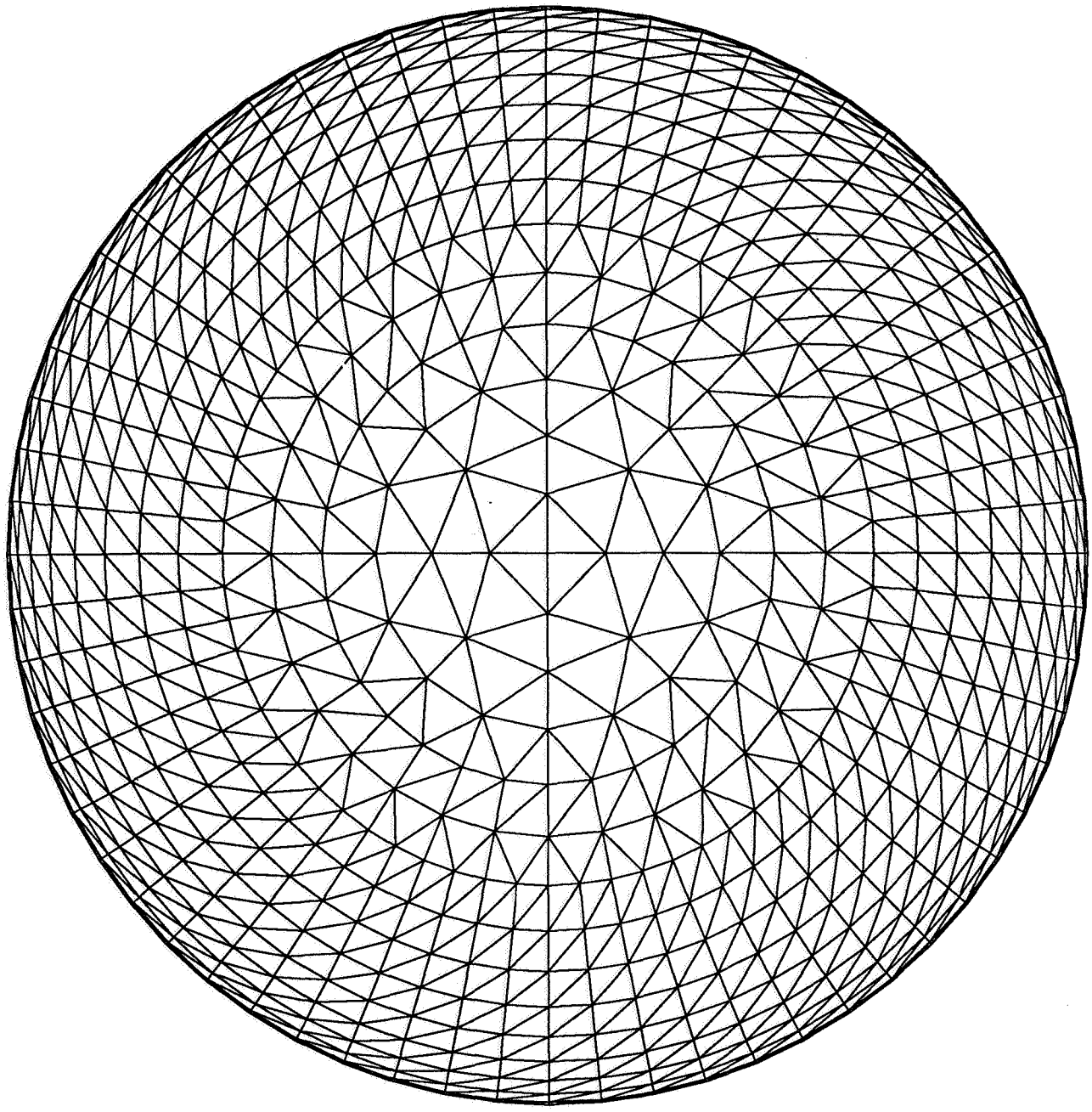


Figure III.2 : Vue du pôle d'un maillage de sphère avec 3056 éléments

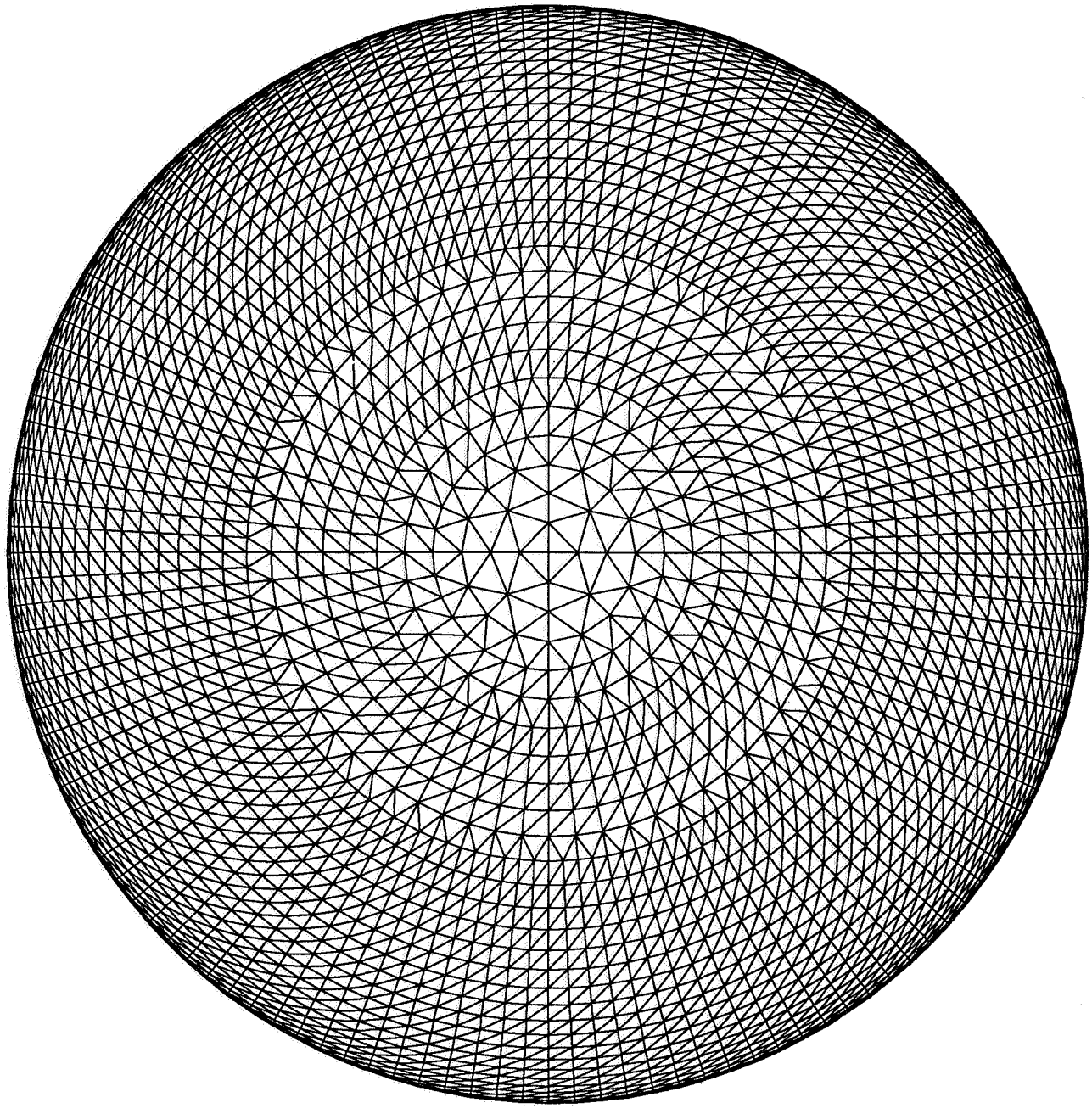


Figure III.3 : Vue du pôle d'un maillage de sphère avec 12656 éléments

Nous allons fixer ici quelques notations. Nous dirons qu'un point du maillage est de niveau i, si le support du degré de liberté qui lui est attaché se répète à l'identique 2^i fois sur le cercle parallèle qui lui est attaché. Nous notons N_i l'ensemble des points de niveau i. Par exemple N_0 est l'ensemble qui contient les deux pôles. Nous noterons N_{niv} le nombre de niveaux. Nous remarquons qu'il n'y a pas de niveau 1 dans un tel maillage puisque le premier découpage comporte quatre triangles.

Chaque niveau est décomposé en tranches verticales suivant les méridiens. Dans le niveau i, il y a 2^i tranches. Nous noterons $T_i^{j_t}$ la $j_t^{\text{ième}}$ tranche du niveau i. Tous les niveaux auront leurs tranches numérotées à partir du même méridien, et dans le même sens de rotation. Il faut souligner que les tranches des niveaux les plus proches de l'équateur (i.e. d'indice les plus élevés) ont beaucoup plus de degrés de liberté que celles qui sont proches des pôles.

Si u est un vecteur sur la base des degrés de liberté du maillage qui vient d'être décrit, nous noterons $u = \left[(u_i^{j_t})_{j_t=0,2^i-1)}\right]_{i=0,N-1}$ où $u_i^{j_t}$ est le vecteur, résultat de la projection de u sur $T_i^{j_t}$.

De même nous noterons $\mathcal{F}(u) = \left[(\hat{u}_i^{j_t})_{j_t=0,2^i-1)}\right]_{i=0,N-1}$ où $(\hat{u}_i^{j_t})_{j_t=0,2^i-1)}$ est la transformée de Fourier par bloc de longueur 2^i de $(u_i^{j_t})_{j_t=0,2^i-1)}$. Similairement $\bar{\mathcal{F}}(u)$ est obtenu avec les transformées de Fourier inverses. Nous précisons enfin ce que nous appelerons *bloc de degrés de liberté* dans la suite du texte. Un bloc de degrés de liberté est un ensemble comprenant tous les degrés de liberté d'une tranche d'un niveau. C'est aussi l'ensemble correspondant après une transformée de Fourier par bloc sur le niveau en question.

III.2 Les algorithmes de calcul

Nous allons présenter ici comment nous réalisons un produit matrice-vecteur de manière accélérée avec un maillage de la structure exposée plus haut. Nous verrons que nous avons encore une structure très creuse et donc un coût faible de calcul. Nous allons utiliser la transformée de Fourier par niveau que nous

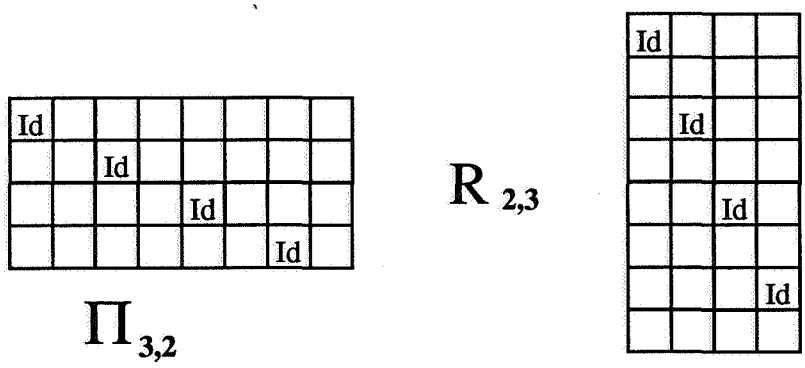


Figure III.4 :

avons appelée $\mathcal{F}(u)$ dans la section précédente. Si A est la matrice d'équations intégrales, alors on note $A_{i,j}$ la sous matrice d'interaction du niveau i avec le niveau j (i est l'indice de ligne). Nous allons regarder de plus près la structure de $A_{i,j}$. Introduisons auparavant deux familles d'opérateurs. On note, pour $i \leq j$ $\Pi_{i,j}$ l'opérateur de l'ensemble des $2^j - uplets$ de blocs de degrés de liberté dans l'ensemble des $2^i - uplets$ qui consiste à ne sélectionner qu'un bloc de degrés de liberté sur 2^{j-i} de la manière qui suit. On ne selectionne le $l^{ième}$ bloc que si l est un multiple de 2^{j-i}. Alors le bloc l est mis à l'emplacement k où k vérifie $2^{j-i}k = l$. Plus précisément,

$$[\Pi_{i,j}]_{k,l} = \delta_{k.2^{j-i},l},\ k = 0, 2^i - 1,\ l = 0, 2^j - 1.$$

De même, on note, pour $i \geq j$, $R_{i,j}$ l'opérateur du niveau j dans le niveau i qui est le transposé de $\Pi_{j,i}$. On a

$$[R_{i,j}]_{k,l} = \delta_{k,l.2^{i-j}},\ k = 0, 2^i - 1,\ l = 0, 2^j - 1.$$

On a noté $\delta_{a,b}$ le symbole de kroneker, par bloc. $\delta_{a,b} = Id$ si a=b et 0 sinon. On peut voir à la figure III.4 deux exemples de tels opérateurs. On introduit encore deux opérateurs. D'une part, pour $i \leq j$, $\tilde{\Pi}_{i,j}$ du niveau j dans le niveau i, avec $\left[\tilde{\Pi}_{i,j}\right]_{k,l} = Id$ si $k \equiv l[2^i]$ et 0 sinon; d'autre part, pour $i \geq j$, $\tilde{R}_{i,j}$ qui est le transposé de $\tilde{\Pi}_{i,j}$.

Proposition 12 *Explicitons la forme de la matrice d'interaction entre les niveaux i et j, $A_{i,j}$. Plusieurs cas sont à considérer.*

- *Si i=0, $A_{i,j}$ est de la forme $(a, ..., a)$ où il y a 2^j blocs a. Le bloc* a *est de dimension* $2\times$ *"taille d'une tranche du niveau j".*
- *Si j=0, $A_{i,j}$ est de la forme* $\begin{pmatrix} b \\ \text{“} \\ \text{“} \\ \text{“} \\ b \end{pmatrix}$ *où il y a 2^i blocs b. Le bloc* b *est de dimension "taille d'une tranche du niveau i"*$\times 2$.
- *Si $i \leq j$, $A_{i,j}$ est de la forme $\Pi_{i,j} circ(a_0, a_1, ..., a_{2^j-1})$ où les a_k sont des blocs. Les a_k sont de taille "taille d'une tranche du niveau i"$\times$ "taille d'une tranche du niveau j".*
- *Si $i \geq j$, $A_{i,j}$ est de la forme $circ(b_0, b_1, ..., b_{2^i-1})R_{i,j}$ où les b_k sont des blocs. Les b_k sont de taille "taille d'une tranche du niveau i"$\times$ "taille d'une tranche du niveau j".*

Il suffit, pour cela, de constater que les interactions entre deux points x et y ne dépendent que du vecteur x-y qui les sépare et sont invariantes par rotation autour de l'axe de révolution. Afin de rendre claire la manière dont agissent $\Pi_{i,j}$ et $R_{i,j}$, nous donnons à la figure III.5 un exemple pour $\Pi_{3,2}$ et $R_{2,3}$.

Nous allons maintenant montrer comment on peut accélérer le calcul de $A_{i,j}u_j$ (pour l'inversion directe, l'algorithme sera similaire). Nous allons d'abord montrer un lemme.

Lemme 21 *Si on note $\mathcal{F}_i$ la transformée de Fourier de longueur 2^i, alors, pour $i \leq j$, $\Pi_{i,j}\mathcal{F}_j = \mathcal{F}_i\tilde{\Pi}_{i,j}$. De même, pour $i \geq j$, $\mathcal{F}_i R_{i,j} = \tilde{R}_{i,j}\mathcal{F}_j$.*

Nous n'allons pas à proprement parler prouver ce lemme, ce qui serait assez technique. Nous allons plutôt donner une idée de la preuve, sachant que les

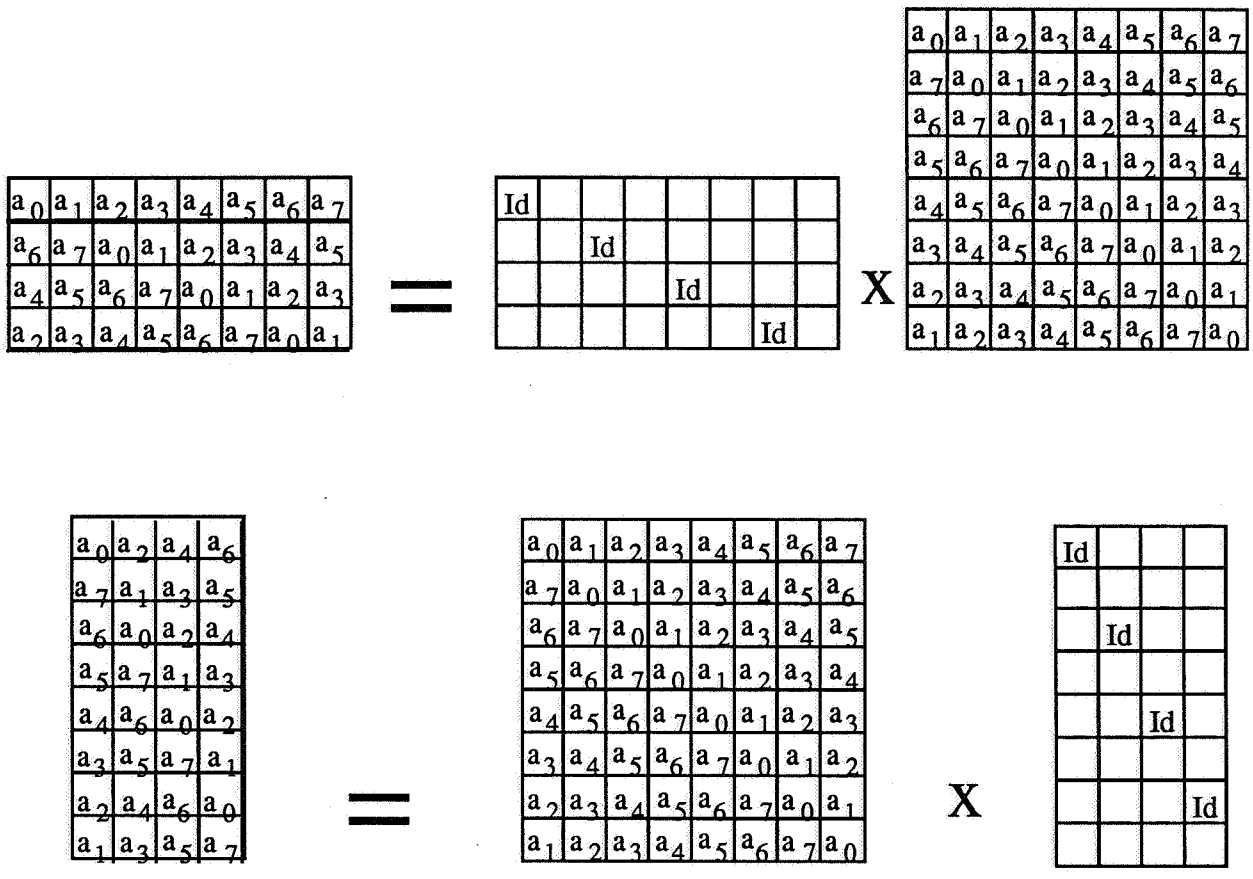


Figure III.5 : Action de $\Pi_{3,2}$ puis de $R_{2,3}$.

détails techniques pour la justifier se trouvent dans [Tem76]. Pour la première assertion du lemme nous allons utiliser l'algorithme de Cooley-Tukey. L'opérateur $\Pi_{i,j}$ ne retient en sortie que les lignes d'indice divisible par 2^{j-i}. Alors, en remontant l'algorithme on s'aperçoit que les i dernières passes de l'algorithme sont celles de Cooley-Tukey de rang i. En entrée de celui-ci on a le vecteur $\left(\sum_{k,\, k\equiv\alpha[2^i]} u_k\right)_{\alpha=0,2^i-1} = \tilde{\Pi}_{i,j}u$.

Pour la deuxième assertion, il faut au contraire regarder l'algorithme de Gentleman-Sande. En effet, le produit par $R_{i,j}$ consiste à mettre en entrée de cet algorithme des zéros partout sauf sur les indices multiples de 2^{i-j} où les composantes du vecteur sont mises à la place. Alors les j premières passes de Gentleman-Sande appliquées aux composantes du vecteur forment la partie combinaisons du Gentleman-Sande d'ordre j. Les dernières passes ne sont que des combinaisons avec des zéros. On a donc, en sortie, la transformée de Fourier de u répétée 2^{i-j}, c'est ce que réalise l'opérateur $\tilde{R}_{i,j}$.

Nous pouvons maintenant détailler le calcul de $A_{i,j}u_j$.

Proposition 13 *Notons $(\hat{a}_l)_{l=0,2^i-1}$ la transformée de Fourier de longueur 2^i de la suite $(a_l)_{l=0,2^i-1}$*

- *Si $i \leq j$, alors en écrivant $A_{i,j}$ sous la forme $\Pi_{i,j}circ(a_0, a_1, ..., a_{2^j-1})$, on a $A_{i,j}u_j = \mathcal{F}_i \left[\tilde{\Pi}_{i,j} diag(\hat{a}_l)_{l=0,2^j-1}\right] \bar{\mathcal{F}}_j u_j$.*
- *Si $i \geq j$, alors en écrivant $A_{i,j}$ sous la forme $circ(b_0, b_1, ..., b_{2^i-1})R_{i,j}$, on a $A_{i,j}u_j = \mathcal{F}_i \left[diag(\hat{b}_l)_{l=0,2^i-1}\tilde{R}_{i,j}\right] \bar{\mathcal{F}}_j u_j$.*

Preuve :

Montrons d'abord la première partie de la proposition. Nous voulons calculer $\Pi_{i,j}circ(a_0, ..., a_{2^j-1})u_j$.

Nous savons que $circ(a_0, ..., a_{2^j-1})u_j = \mathcal{F}_j diag(\hat{a}_l)_{l=0,2^j-1}\bar{\mathcal{F}}_j u_j$, alors en utilisant le lemme 21 on peut conclure. On fait de même pour la deuxième partie de la proposition. Nous avons de plus le résultat suivant.

Proposition 14 *Si* $i = 0$ *et* $A_{i,j} = \Pi_{i,j} circ(a_0, ..., a_{2^j-1})$, *alors* $\hat{a}_l = 0$ *pour* l *non nul. De même si* $j = 0$ *et* $A_{i,j} = circ(a_0, ..., a_{2^i-1}) R_{i,j}$, *alors* $\hat{a}_l = 0$ *pour* l *non nul.*

Preuve :
D'après la proposition 12, on sait que dans ces deux cas, tous les a_l sont égaux. Il en découle de manière immédiate que seul le premier terme de la transformée de Fourier est non nul.

On voit donc que l'on peut remplacer le calcul de Au par celui de $\mathcal{F}B\bar{\mathcal{F}}(u)$ où B est une matrice bloc dont le bloc $B_{i,j}$ est le bloc transformé du bloc $A_{i,j}$ de A à l'aide de la proposition 13. On peut voir sur la figure III.6 un exemple de blocs de B. A la figure III.7 c'est la matrice B toute entière que nous avons

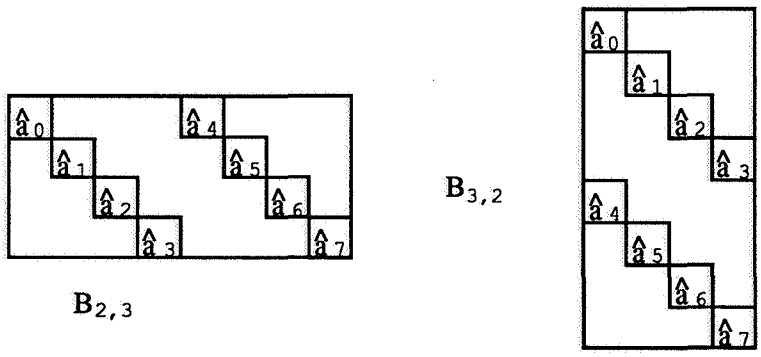


Figure III.6 : Blocs de la matrice B

représentée.

Nous allons maintenant faire une évaluation de la complexité de l'algorithme présenté ici. On peut constater dans un premier temps que la matrice B se présente de manière plus compliquée que dans le cas du maillage en quartiers d'oranges, puisque ici il y a des termes non diagonaux. En fait, nous allons voir que les ordres de complexité des algorithmes sont les mêmes que dans le cas précédent. Nous allons noter N_{niv} le nombre de niveaux. Dans notre cas n_{eq} sera le nombre de divisions qu'il y a sur l'équateur si bien que $n_{eq} = 2^{N_{niv}}$ et n_{me} sera le nombre de divisions d'un méridien. Nous appelerons encore l_i le nombre de

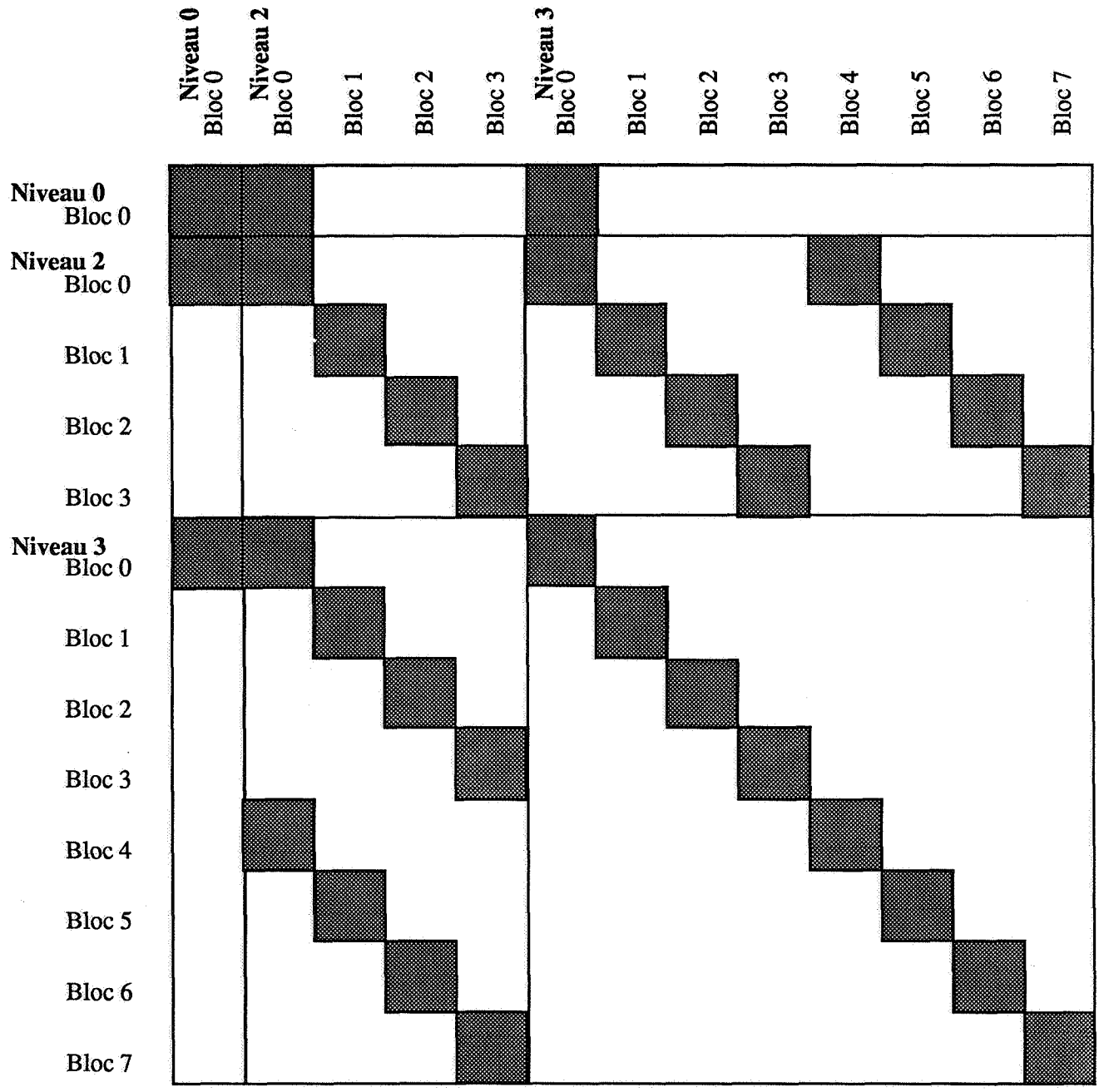


Figure III.7 : La matrice B pour 3 niveaux

degrés de liberté sur une tranche du niveau i. Ainsi $n_me = \sum_{i=0,N_{niv}} l_i$. Enfin, on notera N_p le nombre de degrés de liberté du maillage. On aura, pour une forme donnée, $n_{eq} \sim n_{me} \sim N_p^{1/2}$.

Proposition 15 *Avec ces hypothèses on a :*

- *le calcul de $\mathcal{F}u$ est en $\mathcal{O}(N_p \log N_p)$.*
- *le calcul de B est en $\mathcal{O}(N_p^{3/2} \log N_p)$.*
- *le calcul de Au est en $\mathcal{O}(N_p^{3/2})$.*

Preuve :

Pour calculer $\mathcal{F}u$ il faut réaliser une F.F.T. pour chaque niveau. Au niveau i la F.F.T. coûte $l_i i 2^i$. Nous majorons ce coût par $l_i N_{niv} 2^{N_{niv}}$. Alors le coût du calcul de $\mathcal{F}u$ est majoré par $N_{niv}2^{N_{niv}} \sum_{i=0,N_{niv}} l_i = N_{niv}2^{N_{niv}} n_{me}$, ce qui est équivalent à $N_p \log N_p$.

Pour le calcul de B, il s'agit, pour chaque couple (i,j), de réaliser une F.F.T. de longueur $\max(i,j)$ sur des blocs de taille $l_i l_j$. Pour un couple (i,j) on a donc un coût de calcul dominé par $l_i l_j N_{niv} 2^{N_{niv}}$. Alors le coût du calcul de B est en $N_{niv}2^{N_{niv}} \sum_{i,j} l_i l_j = N_{niv}2^{N_{niv}} \sum_i l_i . \sum_j l_j = N_{niv}2^{N_{niv}} n_{me}^2$, ce qui est équivalent à $N_p^{3/2} \log N_p$.

Pour le calcul de Au on suppose que B est déjà calculé. On doit donc calculer une fois $\mathcal{F}$ et une fois $\bar{\mathcal{F}}$, ce qui est en $\mathcal{O}(N_p \log N_p)$ puis réaliser pour chaque couple (i,j), le produit de $B_{i,j}$ par $\hat{u}_j$. Ce produit est en $l_i l_j 2^{\max(i,j)}$. Alors, le coût de Au est borné par $2\sum_{i,j} l_i l_j 2^j = 2\sum_i l_i \sum_j j 2^j = n_{me} N_p$, qui est équivalent à $N^{3/2}$. Ceci termine la preuve de la proposition.

Ainsi, malgré la complexité rajoutée pour éviter les problèmes de triangles dégénérés aux pôles, on obtient une méthode qui est d'un coût de calcul équivalent à celui obtenu pour un maillage en quartiers d'orange. Ceci est dû au fait que

la diminution du nombre de degrés de liberté du maillage compense l'apparition de blocs non diagonaux.

III.3 Renumérotation et optimisation du remplissage

Dans la section précédente, nous n'avons pas parlé d'algorithmes d'inversion par méthode directe. Ceci est dû au fait qu'en l'état la matrice B à un profil creux ainsi qu'on a pu le voir à la figure III.7. Ceci à pour conséquence qu'une factorisation LU remplirait alors une partie non négligeable de la matrice. Nous allons montrer ici que l'on peut renuméroter les blocs de degrés de liberté de $\mathcal{F}u$ pour obtenir un profil plein pour B. Nous donnerons ensuite une estimation du coût de calcul d'une telle inversion.

Nous allons commencer par étudier le graphe de connectivité de B afin de le présenter sous une forme simple. Nous avons déjà vu auparavant que le profil de B est symétrique.

Proposition 16 *Supposons que $i \leq j$. si $i > 0$, alors dans la sous-matrice $B_{i,j}$, un bloc est non nul si et seulement si il couple le degré de liberté par bloc numéro l du niveau j avec le degré de liberté par bloc numéro k du niveau i et $l \equiv k[2^i]$. Si $i = 0$, un bloc de $B_{0,j}$ est nul si et seulement si il couple le degré de liberté 0 du niveau 0 avec le degré de liberté 0 du niveau j.*

Preuve :
Pour la première partie de la proposition, il suffit de remarquer que la connectivité est donnée par les opérateurs $\tilde{\Pi}_{i,j}$ et $\tilde{R}_{i,j}$ introduits dans la section précédente. Pour la seconde, il faut de plus utiliser la proposition 14.

Nous allons maintenant introduire une représentation de ce graphe de connectivité à l'aide d'arbres. Considérons les arbres suivants, dont les noeuds sont blocs de degrés de liberté au sens précisé page 118.

Définition 2 *$\mathcal{A}$ est un B-arbre si ses noeuds sont des degrés de liberté par bloc de $\mathcal{F}u$ et ses arêtes représentent "la relation est le père de" donnée par : b est le père de a si et seulement si, en notant n_a et n_b les niveaux de a et b, $n_b \neq 0$, $n_a = n_b + 1$ et $a \equiv b[2^{n_b}]$.*

On peut montrer sur ces arbres plusieurs propriétés. On va uniquement s'intéresser aux arbres maximaux.

Proposition 17 *Il y a 4 B-arbres maximaux : $(\mathcal{A}_i)_{i=0,3}$, et ils forment une partition de l'ensemble des degrés de liberté moins le degré zéro du niveau 0.*
$\mathcal{A}_i$ est déterminé uniquement par sa racine qui est le degré i du niveau 2.
Ces arbres sont des arbres binaires.

Preuve :

Montrons d'abord que tout arbre maximal contient un et seul sommet de niveau 2. Il est facile de voir que deux sommets de même niveau ne peuvent être dans un même arbre que s'ils ont un antécédent commun. Or, les sommets de niveau 2 ne peuvent pas avoir de père. En effet il n'y a pas de niveau 1, et on a retranché le seul élément du niveau 0.
Supposons que l'arbre $\mathcal{A}$ est maximal, soit a sa racine et n_a 2 le niveau de celle-ci. Notons alors i l'unique entier dans $[0,3]$ qui vérifie $a \equiv i[4]$. Alors il existe une suite $(a_l)_{l=2,n_a}$, telle que a_l est de niveau l, a_2 est l'élément numéro i du niveau 2 et $a_{n_a} = a$. En effet, il suffit de choisir dans le niveau l le seul élément a_l vérifiant $a \equiv a_l[2^l]$. On a alors une contradiction. Ceci prouve d'abord que tout arbre maximal a une racine de niveau 2, puis qu'il y a 4 arbres maximaux déterminés chacun par leur racine.

Montrons ensuite qu'un élément a du niveau n_a est dans l'arbre $\mathcal{A}_i$ si et seulement si $a \equiv i[4]$. En effet, de même que précédemment, si $a \equiv i[4]$, on peut trouver une suite qui relie a à l'élément i du deuxième niveau. Alors a se trouve dans l'arbre maximal $\mathcal{A}_i$. Réciproquement, si a est dans $\mathcal{A}_i$, alors il existe une suite a_l qui relie a à la racine, mais la relation utilisée montre que,

comme $a_l \equiv a_{l+1}[2^l]$, en particulier $a_l \equiv a_{l+1}[4]$ et par transitivité, que $a \equiv i[4]$. Ceci prouve que les 4 arbres maximaux forment une partition de l'ensemble des degrés de liberté moins le degré 0 du niveau 0. Nous voyons à la figure III.8 un ces quatre arbres dans le cas de quatre niveaux.

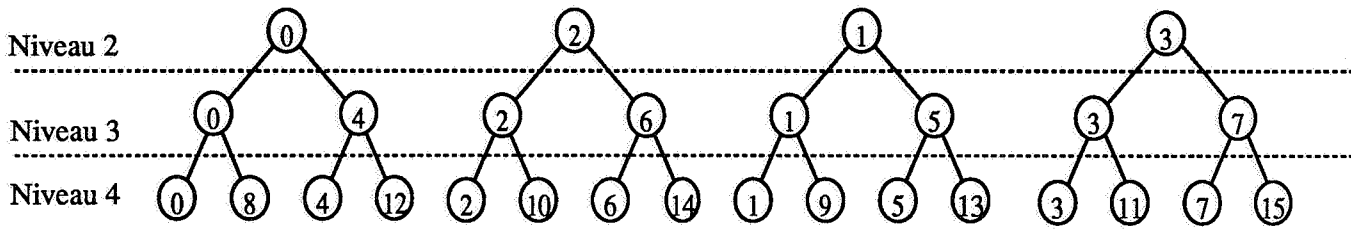


Figure III.8 : Les 4 B-arbres maximaux, pour 4 niveaux

Revenons au graphe de connectivité de la matrice B. Nous allons construire une nouvelle relation à partir des arbres $\mathcal{A}_i$.

Définition 3 *Nous notons $\mathcal{R}$ la relation entre les degrés de liberté par bloc de $\mathcal{F}u$ définie par :*

- *Si le niveau de a est non nul, alors $a\mathcal{R}b$ s'il existe une suite (a_l) telle que le premier élément en soit a, le dernier b, et que $\forall l$, a_l soit le père de a_{l+1}.*
- *Si le niveau de a est nul, alors $a\mathcal{R}b$ si b est l'élément 0 de son niveau.*
- *De plus $\forall a, a\mathcal{R}a$.*

Nous définissons ensuite la relation $\tilde{\mathcal{R}}$ comme étant la symétrisée de $\mathcal{R}$. Ainsi $a\tilde{\mathcal{R}}b$ si et seulement si $a\mathcal{R}b$ ou $b\mathcal{R}a$.

Nous allons alors relier cette dernière relation au graphe de connectivité de B.

Proposition 18 *Le graphe de connectivité de B est le graphe de $\tilde{\mathcal{R}}$.*

Preuve :

Soit deux éléments a et b, de niveaux n_a et n_b. Supposons que $n_a \geq n_b$. Alors, si

$n_b = 0$, $b\tilde{\mathcal{R}}a$ si et seulement si a est l'élément 0 du niveau n_a ce qui est équivalent à dire que a et b sont connectés par B, d'après la proposition 16.
De même, si $n_b > 0$, $b\tilde{\mathcal{R}}a$ si et seulement si il existe un chemin (a_l) qui va de a à b dans le graphe de $\mathcal{R}$. Ceci revient à dire que $a_l \equiv a_{l+1}[2^l]$, et donc que $a_l \equiv a_{l+1}[2^{n_b}]$, par transitivité, $a \equiv b[2^{n_b}]$. Ceci implique, d'après la proposition 16, que a et b sont connectés par B. Réciproquement, si $a \equiv b[2^{n_b}]$, on a vu qu'il existe un chemin qui relie a à b dans un des B-arbres maximaux. Ceci achève la preuve de la proposition.

On a donc que le graphe de connectivité de B a 4 composantes connexes, une par B-arbre maximal. Nous allons maintenant montrer comment on peut réordonner les degrés de liberté, pour obtenir un profil plein pour la matrice B. On va pour cela se servir des arbres que nous avons exhibés. Notons auparavant qu'il suffit de renuméroter les sommets par composante connexe. Les quatre composantes connexes sont les graphes obtenus par la relation $\tilde{\mathcal{R}}$ sur $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$ et $\mathcal{A}_0 \cup \omega$ où ω est l'élément 0 du niveau 0. Considérons l'arbre $\mathcal{A}$ formé des $\mathcal{A}_i$ et ω avec une arête de ω vers la racine de chacun des B-arbres maximaux. Nous représentons $\mathcal{A}$ en rangeant les éléments par niveau, et, à l'intérieur de chaque niveau nous les ordonnons à l'aide du B.R.O. (bit-reverse-ordering). Un exemple d'un tel arbre est donné à la figure III.9 pour 4 niveaux. On peut alors montrer

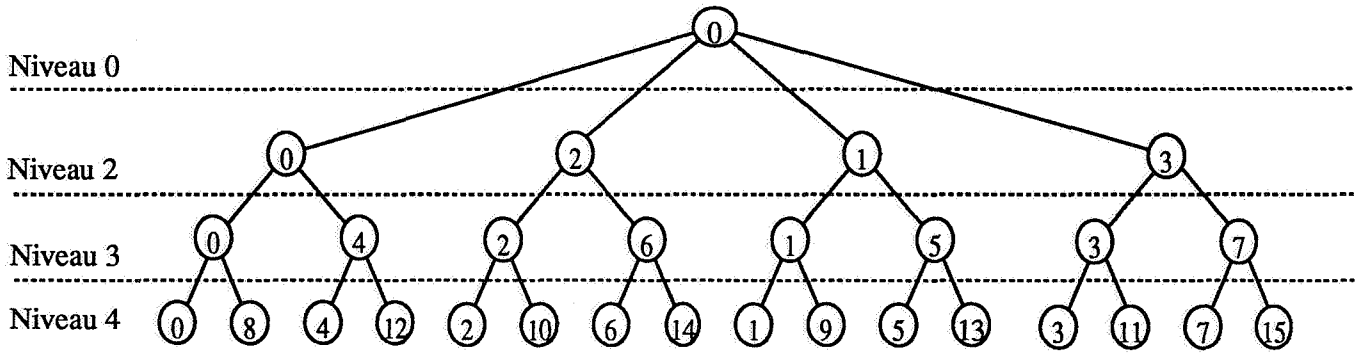


Figure III.9 : l'arbre $\mathcal{A}$ pour 4 niveaux

la proposition suivante.

Proposition 19 *Avec une telle représentation, si a est le $l^{ième}$ élément du niveau i ($i > 0$), alors il est le père des $(2l)^{ième}$ et $(2l+1)^{ième}$ éléments du niveau i+1.*

Preuve :
Notons n_a le niveau de a et $b_0 b_1 ... b_{2^{n_a}-1}$ l'écriture de a en base 2. Alors l s'écrit en base 2 $b_{2^{n_a}-1} ... b_1 b_0$. L'élément a est le père de $0 b_0 b_1 ... b_{2^{n_a}-1}$ et $1 b_0 b_1 ... b_{2^{n_a}-1} 1$ dans le niveau suivant qui sont alors les éléments de rang $b_{2^{n_a}-1} ... b_1 b_0 0$ et $b_{2^{n_a}-1} ... b_1 b_0 1$ soit $2l$ et $2l+1$ dans ce niveau. Nous pouvons maintenant donner l'algorithme que nous utilisons pour réordonner les éléments.

Algorithme 3 (RLRDF) *Pour réordonner les degrés de liberté par bloc de $\mathcal{F}u$ nous les ordonnons d'abord en parcourant $\mathcal{A}$ en profondeur d'abord et de la gauche vers la droite, puis nous renversons cet ordre. (RLRDF signifie Reverse Left to Right Deep First).*

Nous donnons maintenant la propriété essentielle de cet algorithme.

Proposition 20 *L'algorithme RLRDF réduit de façon optimale le profil de B (il n'y a plus de blocs nuls dans ce profil).*

Preuve :
L'élément ω est le dernier numéroté. Le parcours de gauche à droite et en profondeur d'abord assure que les éléments juste avant lui sont d'abord les éléments 0 rangés par ordre de niveaux décroissants. Ce sont tous ceux qui sont connectés à lui par B. Il en découle que le profil de la ligne de ω est plein.
Si l'élément b est différent de ω les éléments qui sont numérotés juste après lui par le parcours de gauche à droite et en profondeur sont d'abord ceux du sous-arbre dont il est la racine, et qui lui sont connectés, puis les autres. Comme dans cette numérotation, les antécédents de b sont numérotés avant lui, les seuls éléments connectés à b qui sont numérotés après lui sont ceux du sous-arbre précédemment cité. Il en résulte, qu'après renversement de la numérotation,

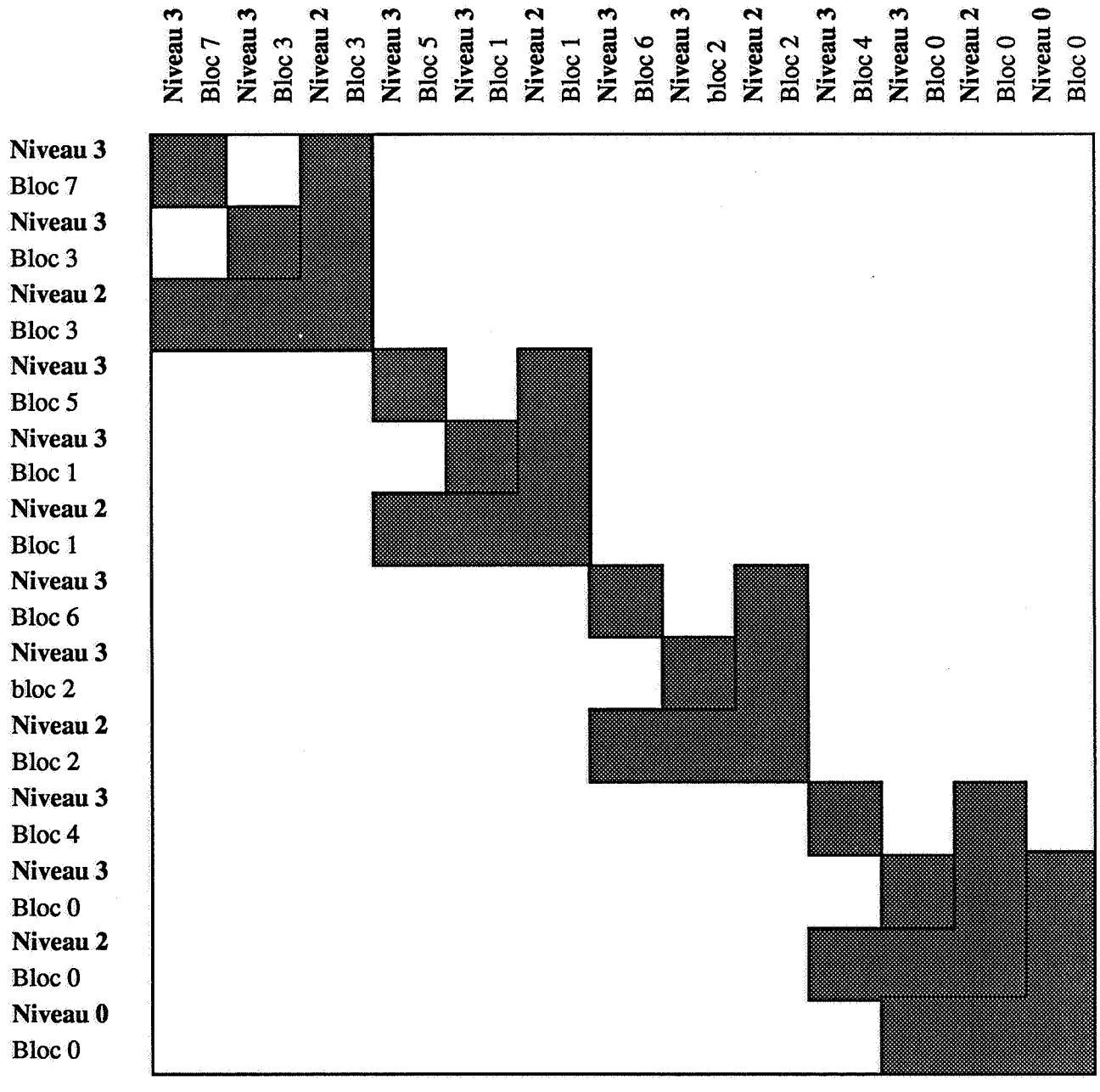


Figure III.10 : Profil de B pour 3 niveaux après RLRDF.

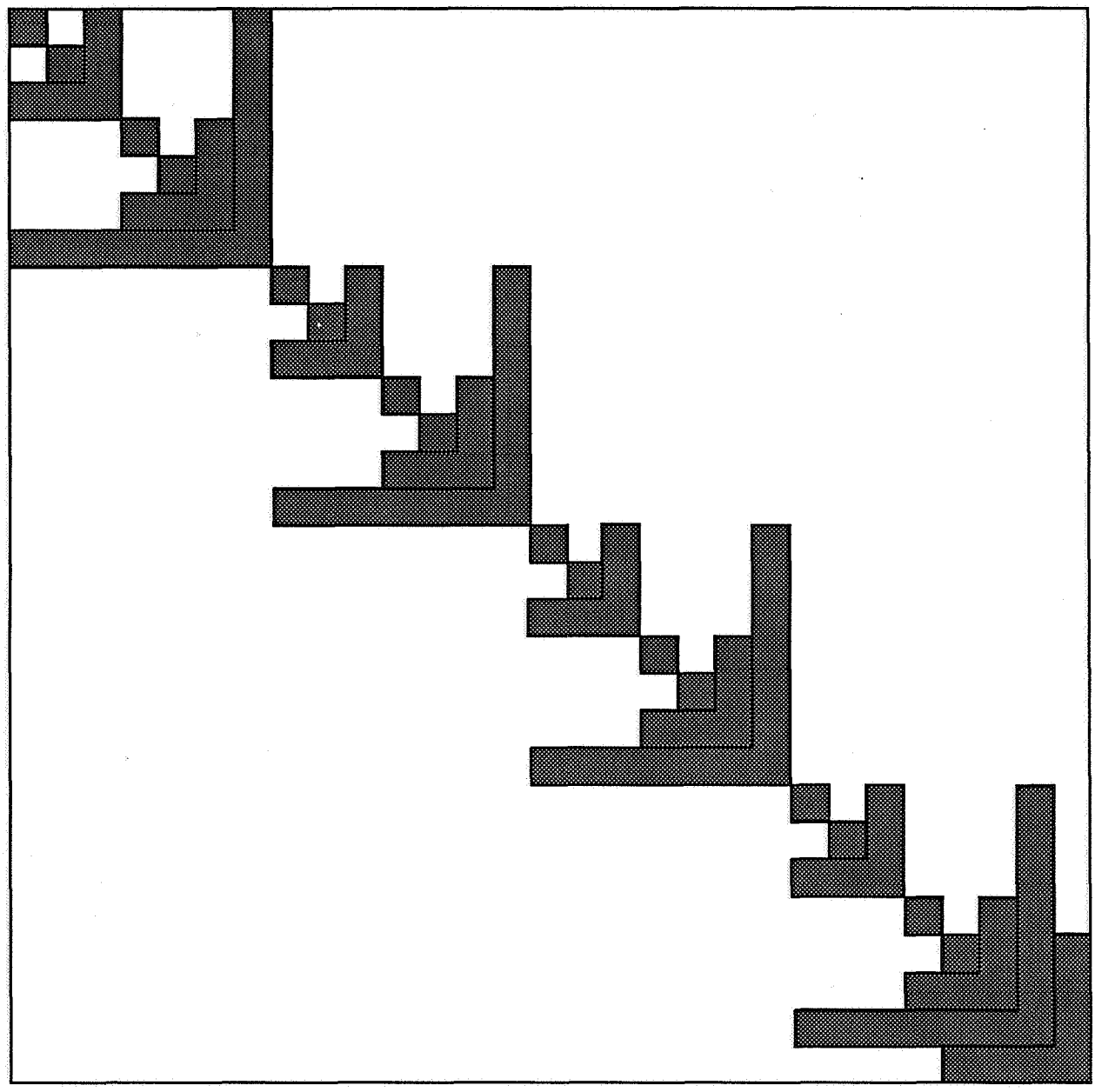


Figure III.11 : Profil de B pour 4 niveaux après RLRDF.

dans la partie de la ligne de b qui est avant b, on a d'abord des éléments qui ne sont pas connectés à b puis des éléments qui sont connectés à b. En conséquence, la ligne de b est à profil plein. Ceci achève la preuve de la proposition. On peut voir aux figures III.10 et III.11 des représentations du profil de B pour 3 et 4 niveaux.

Avec un tel remplissage on peut maintenant envisager l'inversion par factorisation LU en étant assuré de l'optimalité du point de vue du nombre d'opérations effectuées. Nous allons alors donner une évaluation de ce nombre d'opérations. Nous reprenons les mêmes notations que dans la section précédente, que nous rappelons.

- N_{niv} est le nombre de niveaux.
- n_{eq} est le nombre de divisions de l'équateur : $N_{niv} = 2^{n_{eq}}$.
- n_{me} est le nombre de divisions d'un méridien.
- l_i est le nombre de degrés de liberté sur une tranche de niveau i : $\sum_{i=0,N_{niv}} l_i = n_{me}$.
- N_p est le nombre de degrés de liberté du maillage.

Proposition 21 *le coût de la factorisation LU de B après renumérotation RL-RDF est en* $\mathcal{O}(N_p^2)$.

Preuve :

Nous allons commencer par négliger le coût de calcul dû à la partie de B concernant le niveau 0. En effet, ce niveau n'a que deux degrés de liberté, le pôle nord et le pôle sud, et les lignes sont de longueur m.

Il nous reste alors 4 morceaux identiques qui viennent des quatre B-arbres maximaux. Nous allons procéder par récurrence. Notons C_i le coût de l'inversion d'un bloc provenant d'un sous-arbre dont la racine est de niveau i. Le coût que

l'on cherche est $4C_2$. Un tel bloc a la structure suivante :

A_0	0	e
0	A_1	f
c	d	C

où A_0 et A_1 sont des blocs provenant de sous-arbres dont la racine est de niveau i+1. Alors, $C_i = 2C_{i+1} + 4l_iR_i + 2l_{i+1}l_i$, où R_i est le nombre de coefficients non nuls de A_0 ou A_1. En effet, il faut d'abord calculer les factorisées de A_0 et A_1, puis cA_0^{-1}, $A_0^{-1}e$, dA_1^{-1} et $A_1^{-1}f$, et enfin $C - cA_1^{-1}e - dA_1^{-1}f$. Nous allons majorer C_i. Nous allons utiliser le fait que $R_i \leq \frac{R}{2^i}$ où R est le nombre de coefficients de la matrice B toute entière. Alors $C_i \leq 2C_{i+1} + 4l_i\frac{R}{2^i} + 2l_{i+1}l_i$. Remarquons que $l_{i+1}l_i \leq R_i$. nous pouvons alors écrire que $C_i \leq 2C_{i+1} + 6l_i\frac{R}{2^i}$. En utilisant cette inégalité nous obtenons que $C_2 \leq 2^{N_{niv}-2}C_{N_{niv}} + CR\sum_{l=3,N_{niv}} l_i$ où C est une constante. Comme $C_{N_{niv}} \leq l_{N_{niv}}^3 \leq \frac{R}{2^{N_{niv}}}n_{me}$. Alors n_{me}. Or, on a vu que le remplissage de B est en $\mathcal{O}(N_p^{3/2})$. Comme $n_{me} \sim N_p^{1/2}$ on obtient finalement que le coût de factorisation LU de B est dominé par CN_p^2. Ceci achève la preuve de la proposition.

Le résultat principal que l'on est parvenu à obtenir dans ce chapitre est donc que l'on peut à la fois éviter le problème de dégénerescence des triangles au voisinage des pôles et garder un coût de calcul identique à celui obtenu dans le cas d'un maillage en quartiers d'orange.

Conclusion

Dans le travail que nous venons de présenter nous nous sommes attachés à réduire le nombre d'itérations de la méthode de résolution que nous avions choisie (GCRA) pour diminuer le temps de résolution et le stockage des directions de descente. Nous avons obtenu, en nombre d'itérations, une amélioration notable de la vitesse de convergence.

La deuxième partie a consisté à présenter une méthode de calcul qui diminue le stockage et le temps de calcul de la matrice. Le changement d'échelle obtenu dans la complexité des algorithmes permet d'espérer, dans un avenir raisonnablement proche, passer de cas de quelques dizaines de milliers de degrés de liberté pour les meilleurs codes actuels qui sont en $\mathcal{O}(k^4)$, à presque un million pour un code en $\mathcal{O}(k^3)$ comme celui dont nous avons décrit l'algorithme dans la dernière partie (on suppose que n est proportionnel à k^2). C'est donc un seuil quantitatif important que cette méthode permet de franchir.

Néanmoins de nombreux points restent à étudier. Pour le préconditionneur, il est nécessaire, si l'on veut atteindre une précision plus grande et une plus grande vitesse de calcul, d'accélerer son inversion sans pour autant détériorer la parallélisme de la méthode. Pour la réduction du volume mémoire, le problème du choix de l'algorithme de résolution de la méthode couplée est encore ouvert.

Bibliographie

[AS64] M. Abramowitz and I. A. Stegun, editors. *Handbook of mathematical functions*, National Bureau of Standards, U.S. Government, Washington D.C., 1964.

[Ben84] Abderrahmane Bendali. *Approximation par éléments finis de surface de problèmes de diffraction des ondes électromagnétiques.* Thèse d'état, Université Pierre et Marie Curie, 1984.

[BH87] A. Bamberger and T. Ha Duong. Diffraction d'une onde acoustique par une paroi absorbante: nouvelles équations intégrales. *Math. Meth. in the Appl. Sci.*, 9:431–454, 1987.

[Cia91] P.G. Ciarlet. Basic error estimates for elliptic problems. In P.G. Ciarlet and J.L. Lions, editors, *Handbook of numerical analysis*, page 135, North-Holland, 1991.

[CP81] J. Chazarain and A. Piriou. *Introduction à la théorie des équations aux dérivées partielles linéaires.* Gauthier-Villars, Paris, 1981.

[CT65] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comp.*, 19:297–301, 1965.

[DL85a] R. Dautray and J.L. Lions. *Analyse mathématique et calcul numérique pour les sciences et les techniques*, chapter XI, pages 553–702. Volume 2, Masson, 1985.

[DL85b] R. Dautray and J.L. Lions. *Analyse mathématique et calcul numérique pour les sciences et les techniques*, chapter XI, pages 27–29. Volume 1, Masson, 1985.

[Dui73] J. J. Duistermaat. *Fourier integral operators.* Courant Institute of Mathematical Sciences, New York, 1973.

[Foc46] V. Fock. The distribution of currents induced by a plane wave on the surface of a conductor. *Journal of Physics*, 10:130–136, 1946.

[GS66] W. M. Gentleman and G. Sande. Fast Fourier transforms for fun and profit. In *1966 Fall Joint Computer Conference, AFIPS Proc.*, pages 297–301, 1966.

[Ham81] M.A. Hamdi. Une formulation variationnelle par équations pour la résolution de l'équation de Helmholtz avec des conditions aux limites mixtes. *Note au CRAS*, Série II, t. 292:17–20, 1981.

[Jol88] Pascal Joly. Méthodes de gradient conjugué, cours de D.E.A. *Publications du laboratoire d'analyse numérique de l'université de Paris 6*, 1988.

[LM68] J. L. Lions and E. Magenes. *Problèmes aux limites non homogènes et applications.* Volume 1, Dunod, Paris, 1968.

[Ned78] J.C. Nedelec. *Notions sur les équations intégrales de la physique, théorie et approximation.* Technical Report, Ecole Polytechnique, 1978.

[Ned82] J.C. Nedelec. *Integral equations and operator theory*, chapter Integral equations with non integrable kernels. Volume 5, Birkhäuser Verlag, Bâle, 1982.

[NU88] Arnold F. Nikiforov and Vasilii B. Uvarov. *Special fonctions of mathematical physics.* Birkhäuser, Basel Boston, 1988.

[Rok83] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *Journal of Computational Physics*, 60:187–207, 1983.

[Rou91] F.X. Roux. Spectral analysis of the interface operators associated with the preconditioned saddle-point principle domain decomposition method. In *Domain Decomposition Conference (Norfolk)*, 1991.

[SS85] Y. Saad and M.H. Schultz. *Topological properties of hypercubes.* Technical Report YALEU/DCS/RR-389, Yale University, 1985.

[Tem76] C. Temperton. *Mixed-radix fast Fourier transforms without reordering.* Technical Report, European Centre For Medium Range Weather Forecasts, Bracknell, United Kingdom, 1976.

[VdSVdV86] A. Van der Sluis and H.A. Van der Vorst. The rate of convergence of conjugate gradients. *Numerische Matematik*, 48:543–560, 1986.

Table des Matières