

Rapport final Magistère de Mathématiques

BERA Antoine

2012 - 2016



Trois années de Magistère effectuées à l'Université Paris-Sud

Table des matières

Table des matières	1
1 Mes années de Magistère	2
1.1 La L3...	2
1.2 Le M1...	3
1.3 La prépa Agreg...	4
1.4 Le M2...	4
1.5 Le rôle du Magistère	5
2 Les problèmes d'invisibilité	6
2.1 Les mathématiques utilisées	6
2.2 L'obstacle pénétrable	7
2.3 Les autres type d'invisibilité dans le guide d'ondes.	9
2.4 Changement de point de vue	11
2.5 L'espace libre	11
Références	12
3 Annexes	13
3.1 Projet L3 - Caractérisation de la norme L^p du gradient	13
3.2 Projet M2 - Réduction de modèles	33
3.3 Projet M2 - Résolution des problèmes de diffraction par équations intégrales	41
3.4 Projet M2 - Création de la classe des matrices creuses	58
3.5 Stage M2 - Conception d'obstacles invisibles à fort contraste dans un guide d'ondes	70

1 Mes années de Magistère

Après avoir effectué 3 années de classes préparatoires au lycée Faidherbe à Lille, je me suis tourné vers le Magistère d'Orsay, avec pour unique objectif de passer l'agrégation. J'ai effectué 2 années de Licence 3 et Master 1 dans cette université puis une année de préparation à l'agrégation suivie d'un Master 2 à l'Université Paris-Saclay.

Durant ces deux merveilleuse années j'ai pu découvrir de très jolies Mathématiques, mais aussi d'excellents professeurs et mathématiciens.

Ma vie d'étudiant ne s'est bien entendu pas arrêtée aux Mathématiques puisque je me suis fais des amis et des rencontres toutes plus extraordinaires les unes que les autres. J'ai une pensée toute particulière pour Nathalie Carrière qui a été une secrétaire remarquable pendant mes 3 années de Magistère.

1.1 La L3...

En arrivant dans un endroit aussi loin de chez moi, aussi grand, je me suis senti perdu... Quoi de mieux pour s'intégrer qu'une journée d'intégration menée par le BDE!? Cette journée m'a permis de découvrir le campus, mes camarades, les "M1", mais aussi les rumeurs sur le prof de Lebesgue qui ne mettait jamais au dessus 10...

Les cours ont démarré et je me suis assez rapidement senti dans mon élément avec des cours essentiellement portés sur l'analyse, qui est mon domaine favori. Les cours de topologie, de Lebesgue, de Fourier, d'équa-diff, de fonctions holomorphes, etc. étaient pour moi le bonheur en comparaison aux cours d'anglais et d'algèbre.

En plus de tous ces cours fort sympathiques, de par mon choix de faire le Magistère, j'ai dû choisir 2 options (au lieu de une... c'est vache de nous imposer autant de souffrance!) qui ont été l'informatique et la théorie de graphes. Ces cours n'étaient pas nécessairement ma spécialité mais j'ai été fortement intéressé par la pédagogie des professeurs et la culture qu'ils pouvaient m'apporter.

Au second semestre, la pique fût plus importante : la topologie générale... Un cours assez formel, manquant d'exercices, mais fort intéressant. Je n'ai compris qu'en agrégation l'importance (autre que culturelle) de ce cours, qui m'était assez mystérieux.

Cette année était ponctuée, en binôme, d'un Travail Encadré de Recherche (TER) encadré par Radu Ignat. Il portait sur de l'intégration, et en particulier sur la caractérisation de la norme L^p du gradient (cf. Annexe 1). Je me suis senti à l'aise avec le sujet, même si j'ai dès lors ressenti que la recherche n'était pas trop faite pour moi.

Ce fût une année particulièrement riche mathématiquement parlant, puisque j'y ai découvert et/ou approfondi énormément de domaines mathématiques. J'ai eu la chance de recevoir une bourse FMJH pour mes années futures.

1.2 Le M1...

Le M1 a été une année durant laquelle il a fallu choisir une orientation. Sans trop d'hésitations je me suis tourné vers l'analyse avec en particulier vers les cours de Distributions et de problèmes d'évolution. Ces cours sont très utiles dans mon domaine de recherche actuel puisque je me suis tourné vers les EDP.

Le cours de Mathématiques assistées par Ordinateur (MAO) m'a porté vers ce domaine de recherche puisque j'y ai appris les bases de l'analyse numérique des EDP (pile poil mon domaine aujourd'hui..). Il m'a aussi énormément servi lors de la préparation à l'agrégation : la quasi-totalité du programme de mon option à l'agrégation avait été vue lors de ce cours.

La deuxième voie suivie durant cette année a été naturellement celle de l'agrégation avec les cours d'Analyse/Maths-Générales répartis sur deux semestres. Un cours m'a particulièrement marqué : celui de Laurent Moonens où l'on a découvert en détail la construction et l'utilisation de l'intégrale de Kurzweil-Henstock. Durant ce cours Laurent Moonens m'a demandé de relire un article, écrit avec un étudiant, portant sur ce sujet. Ca a été une expérience aussi intéressante que nouvelle.

Le parcours de Magistère nous demandait de suivre deux cours supplémentaires, qui ont traité de la théorie spectrale et de la géométrie différentielle. Le premier cours, présenté avec des exercices, était un changement radical avec la première année. Cette belle idée que sont les exercices n'a, malheureusement, pas été prolongée avec le cours de géométrie différentielle. L'ironie de ces cours est que ces thèmes très théoriques ne m'ont que très rarement (pour ne pas dire jamais) été utiles avant mon stage de M2 en mathématiques appliquées.

Le TER de M1 a été effectué avec Arnaud Personne, encadré par Laurent Moonens, et traitait de la mesure de Hausdorff, le Théorème de Besicovitch et de son application au Cantor quatre coins. Il a donné lieu à une présentation devant des étudiants de M2 AAG, lors d'un séminaire.

Durant l'été qui a suivi j'ai effectué mon apprentissage Hors-mur, au Bureau Veritas de Villeneuve-d'Asq. J'y ai appris les bases du métier de contrôleur de gestion et à utiliser Excel comme je ne l'imaginai pas auparavant.

1.3 La prépa Agreg...

Durant l'année précédente, j'ai présenté le concours de l'ENS Cachan, où j'ai été reçu. C'est donc à l'ENS Cachan que j'ai effectué ma préparation à l'agrégation.

Cette année a été riche en travail, sans grande découvertes mathématiques. Cependant elle m'a permis de poser les notions apprises jusqu'ici.

J'ai choisi de prendre l'option Calcul Scientifique reprenant, en grande partie le cours de MAO de M1. Ca n'a tout de même pas été aussi simple puisqu'il a fallu apprendre un nouveau langage de programmation : le Python. Un logiciel avec lequel j'ai eu beaucoup de mal à démarrer mais qui est devenu indispensable dans ma vie de jeune thésard.

J'ai obtenu l'agrégation et me suis tourné vers un M2 recherche.

C'est aussi durant cette année que j'ai trouvé un sujet de recherche portant sur l'analyse numérique de EDP : l'invisibilité !

1.4 Le M2...

Je pensais initialement me tourner vers le M2 EDP de l'université Paris-Sud, mais il avait malheureusement changé de nom du fait de la création de Paris-Saclay. C'est aujourd'hui le M2 AMS (Analyse Modélisation et Simulation) séparé en deux branches : AM (anciennement EDP) et MS. Fort de ma réussite en calcul scientifique, et ayant en vue ma thèse sur l'invisibilité, je me suis tourné vers le cursus MS.

J'y ai appris énormément de choses, notamment de C++ et la propagation d'ondes dans un guide d'ondes. Des projets étaient souvent au rendez-vous : on trouvera en Annexe quelques exemples.

Mais une année de M2 n'est rien sans un stage de 5 mois ! J'ai effectué mon stage avec mes futurs encadrants de thèse, Anne-Sophie Bonnet-Ben Dhia et Lucas Chesnel. Ce stage traitait de l'invisibilité dans un guide d'ondes.

Durant ce stage j'ai été réellement confronté à la recherche, mais également à des encadrants exigeants. Cette exigence m'a été bénéfique ! En effet, je n'ai aucune attirance pour la recherche, et mes qualités de rédaction de rapport sont inexistantes. Ils m'ont permis de progresser dans ce domaine, en rendant un rapport très appréciés du jury. Ils commencent également à me donner confiance en moi, qui me sentais incapable de faire de la recherche. Ces avancées personnelles me renforcent dans l'idée, étrange à mon goût, de me lancer à fond durant 3 ans pour réussir une thèse et ne pas décevoir leurs exigences.

C'est ainsi que je me lance dès à présent dans ce domaine très vaste qu'est l'invisibilité. On pourra lire, dans mon rapport, le cadre de mon domaine de recherche, avec notamment le cas du guide d'ondes et le lien avec la théorie spectrale.

1.5 Le rôle du Magistère

Ces cours supplémentaires étaient, pour mon plus grand plaisir, fort théoriques. J'avais cependant tendance à les oublier de par leur manque d'applications dans d'autres matières.

Mais ces cours ont été plus qu'utiles lors de ma préparation à l'agrégation, et le sont toujours aujourd'hui.. Certainement plus que d'autres cours du tronc commun. De plus, ils m'ont permis d'avoir la bourse FMJH, l'ENS et un bon rang à l'agrégation. Je suis convaincu que le supplément de cours du Magistère n'est pas anodin dans cette réussite.

Je recommande vivement cette voie à qui me pose la question, et suis heureux d'avoir eu la chance de pouvoir bénéficier de cette formation d'excellence.

2 Les problèmes d'invisibilité

On a tous déjà entendu parler d'invisibilité, ou même rêvé de devenir invisible... Mais que signifie réellement ce terme "invisible" ?

Pour fixer les idées, parlons d'abord de l'invisibilité lumineuse. La perception que j'ai d'un objet est traduite par les ondes lumineuses pénétrant dans mes yeux. On parle alors d'invisibilité dans la direction de mes yeux lorsque les ondes y pénétrant sont les mêmes avec ou sans objet. Mais les ondes lumineuses arrivant sur l'objet ne sont pas dirigées uniquement vers mes yeux. C'est pourquoi on parlera d'invisibilité lorsque, dans toutes les directions, les ondes se propagent de la même manière avec ou sans objet.

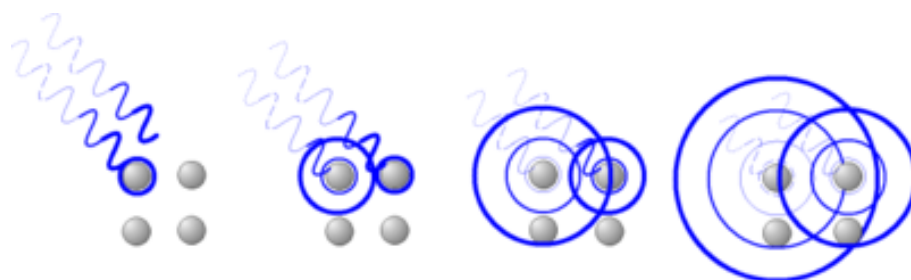


FIGURE 1 – Onde incidente et onde diffractée.

Dans notre étude, nous allons nous fixer un champ incident, et nous intéresser à l'invisibilité d'un objet dans un nombre fini de directions. Cette restriction est effectuée puisque dans la pratique, lorsque l'on observe quelque chose, on n'utilise qu'un nombre fini d'éléments de mesure.

2.1 Les mathématiques utilisées

Dans ce domaine des mathématiques, nous sommes amenés à étudier une équation aux dérivées partielles ; nous sommes donc naturellement dirigés vers l'utilisation de la théorie variationnelle, avec en particulier la formule de Green ou des théorèmes d'existence et d'unicité de solutions. Le cours [1] présente de manière précise cet aspect du problème.[2] et [3] présentent également l'invisibilité en utilisant ces outils.

On étudie principalement le problème de Helmholtz qui s'écrit assez simplement comme un problème aux valeurs propres. C'est pourquoi nous sommes amenés à utiliser la théorie spectrale. La partie "changement de point de vue", présentée plus tard, utilise particulièrement cet outil mathématique pour rechercher les fréquences d'invisibilité d'un obstacle en étudiant le spectre de l'opérateur associé.

Ensuite nous avons à utiliser des méthodes de scattering permettant d'analyser la diffraction

d'une onde sur par un obstacle. Ce n'est un outils mais bien une méthode, c'est-à-dire que l'on va utiliser des démarches classiques. [4] et [5]

Je serai peut-être amené à me pencher sur la géométrie différentielle, puisque l'on cherche à déterminer le lieu des zéros d'une fonction. Pour en comprendre la nature et l'utiliser correctement il peut être intéressant d'utiliser cette théorie.

Dans ces sujets de recherche, la mise en application par résultats numériques est primordiale (par exemple la méthode des éléments finis est importante au quotidien). C'est pourquoi j'ai utilisé et j'utiliserai toujours mon bagage en analyse numérique, en complétant selon les besoins.

Un dernier point, qui rejoint le précédent, est la réduction à un domaine borné chaque équation que l'on souhaite. Si l'on ne borne pas le domaine nous ne pourrons jamais résoudre informatiquement le problème. Pour cela nous utiliserons, dans le cas du guide d'onde la méthode des DtN (Dirichlet to Neumann) [1]. Nous utiliserons également la méthode de PML (Perfectly Matched Layers) présentée dans le cours [1] et dans l'article [6].

2.2 L'obstacle pénétrable

Ici, nous nous intéresserons plus précisément au cas d'un obstacle placé dans un guide d'ondes, ce qui restreint à 2 directions d'invisibilité. Nous enverrons une *onde incidente* provenant de la gauche, se dispersant sur l'obstacle en produisant une *onde réfléchie* (repartant vers la gauche) et une *onde transmise* (partant vers la droite).

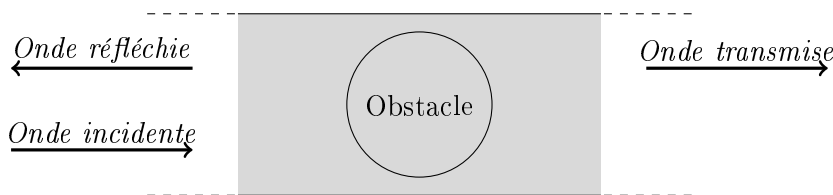


FIGURE 2 – Guide d'ondes.

L'objectif de cette étude est de parvenir à construire un obstacle pénétrable, invisible dans un guide d'onde. L'obstacle sera alors une inconnue dans toute notre démarche.

Fixons donc un champ incident, u_i , provenant de $-\infty$. Nous savons qu'à fréquence donnée, le champ diffracté se décompose en somme d'ondes propagatives (se propageant à l'infini) et d'ondes évanescentes (exponentiellement décroissantes en $\pm\infty$). On prendra ici le cas où la fréquence est suffisamment petite de telle sorte qu'il n'y ait qu'une seule onde propagative.

On va pouvoir ainsi définir les coefficients de réflexion et de transmission (R et T) représentant l'amplitude du champ total (champ incident + champ diffracté) dans les deux directions possibles du guide. L'objectif est d'obtenir $R = 0$ (non réflexion) et $T = 1$ (transmission parfaite). L'obstacle sera caractérisé par l'indice ρ du milieu en chaque point, qui est donc une inconnue du problème. En effet, ces deux coefficients R et T dépendent de l'indice dans tout le milieu, donc du défaut présent dans le guide d'ondes.

On remarque déjà que l'objectif de ces travaux est de trouver des zéros non triviaux d'une fonction donnée : par exemple, si on cherche à résoudre uniquement la non réflexion, il nous faut trouver ρ tel que $R(\rho) = 0$ (on peut également chercher à résoudre $R(\rho) = 0$ et $T(\rho) - 1 = 0$ simultanément afin d'avoir invisibilité totale). Intuitivement on imagine que pour annuler un nombre fini de valeurs avec un paramètre qui vit dans un espace de dimension infinie (une fonction), il y aura probablement de multiples façons d'en trouver. Nous ne devons pas seulement en trouver, mais plutôt construire des zéros on donnant une forme souhaitée.

Un principe important de la physique est la conservation de l'énergie. Ce principe affirme que la variation d'énergie d'un système est égale à la somme des énergies entrantes et sortantes du système. Dans notre cas, l'énergie entrant est transportée par le champ incident dans le guide d'ondes. L'énergie transportée par un champ u à travers une surface S est $Im \left(\int_S \bar{u} \frac{\partial u}{\partial \mathbf{n}} d\sigma \right)$. On peut alors calculer l'énergie entrant dans le guide d'onde (via le champ incident) ainsi que l'énergie sortant du guide d'onde (via le champ diffracté). Ce principe nous permet donc de donner une relation entre les coefficients R et T : $|R|^2 + |T|^2 = 1$. Nous détaillerons plus loin les calculs permettant d'obtenir ce résultat.

Si il n'y a pas d'obstacle, l'onde passe sans être diffractée, c'est une situation d'invisibilité connue, permettant d'annuler notre fonction : $\rho = 0$ est un point d'annulation de R et de $T - 1$. Nous allons alors utiliser une méthode de continuation, consistant à partir d'un point d'annulation connu (ici aucun obstacle, $\rho = 0$) afin de trouver un autre point d'annulation dans un voisinage du premier. Le voisinage dans lequel on a trouvé ce point d'annulation peut être petit, mais l'objectif est d'itérer le processus afin de s'éloigner le plus possible du point de départ. Il a déjà été prouvé que l'on peut construire des perturbations non triviales invisibles, et mon travail consiste en la compréhension et la réalisation informatique de cette première étape afin de pouvoir itérer la méthode ; c'est à dire partir de l'obstacle invisible trouvé pour en construire un nouveau etc. Cette démarche sera effectuée en gardant pour objectif de construire des obstacles invisibles pouvant éventuellement être réalisés physiquement.

Cette partie est développée en détail dans le rapport se trouvant en Annexe.

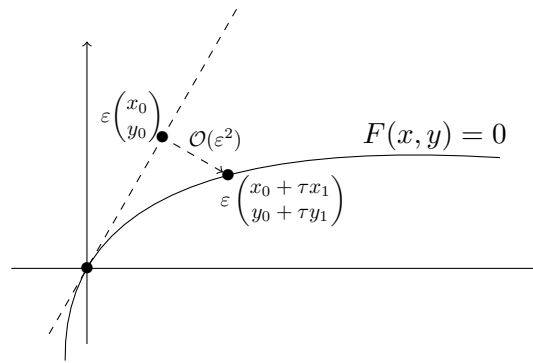
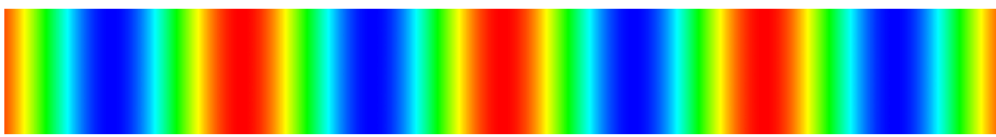
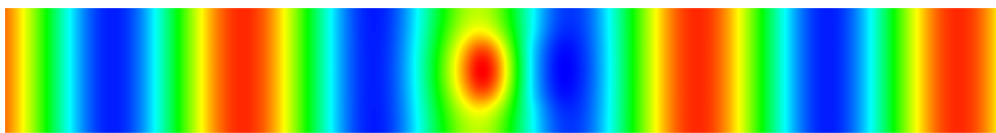


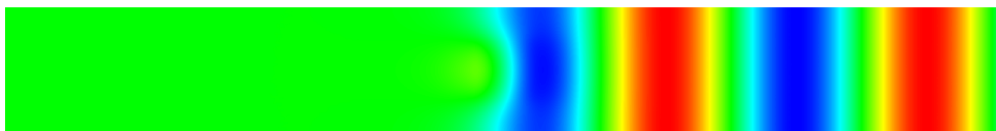
FIGURE 3 – Exemple de méthode de continuation



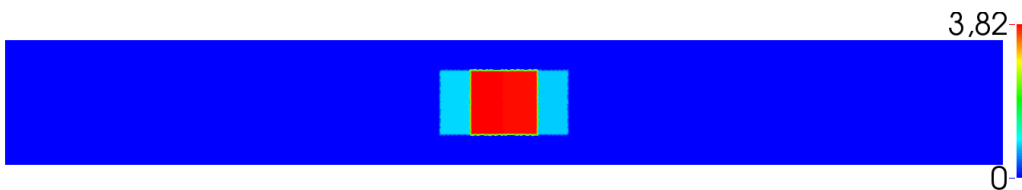
(a) Champ incident u_i



(b) Champ total u



(c) Champ diffracté u_s



(d) Perturbation ρ

FIGURE 4 – Diffraction d'une onde sur un défaut non réfléchissant - $L = 4$, $k = 3$.

2.3 Les autres type d'invisibilité dans le guide d'ondes.

Dans le domaine de l'invisibilité dans un guide d'onde, nous pouvons également nous intéresser à une perturbation du bord que l'on cherche à rendre invisible. La démarche est exactement la même, c'est-à-dire que nous utiliserons les coefficients de réflexion R et de transmission T .

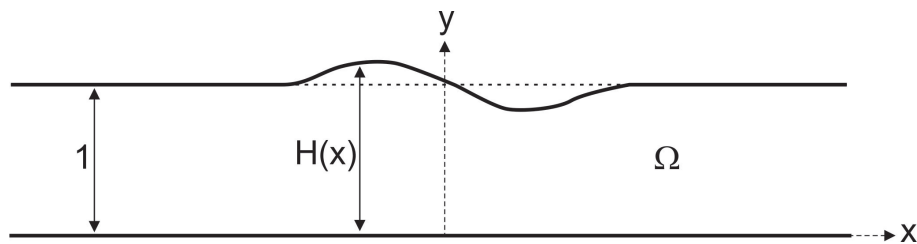


FIGURE 5 – Guide perturbé sur le bord.

Je me suis intéressé à ces question, par curiosité sur le sujet, de manière à comprendre les différents types d'invisibilité dans un guide d'ondes.

Il a été prouvé, de la même manière que précédemment, que l'on pouvait créer des "petites" perturbations du bord. Ces résultats sont en partie dûs à A-S Bonnet-BenDhia, L. Chesnel et S.A. Nazarov. Je leur cède d'ailleurs les droits d'images que j'utiliserai dans la suite.

On peut par exemple construire des petites cheminées, de largeur $\varepsilon > 0$, sur le bord d'un guide d'ondes qui seront invisibles à fréquence donnée initialement. On peut également imaginer d'autres géométries. Une idée peut donc être de réaliser une méthode de continuation... Si la recherche me mène dans cette direction j'essayerai sans doute d'adapter la méthode.

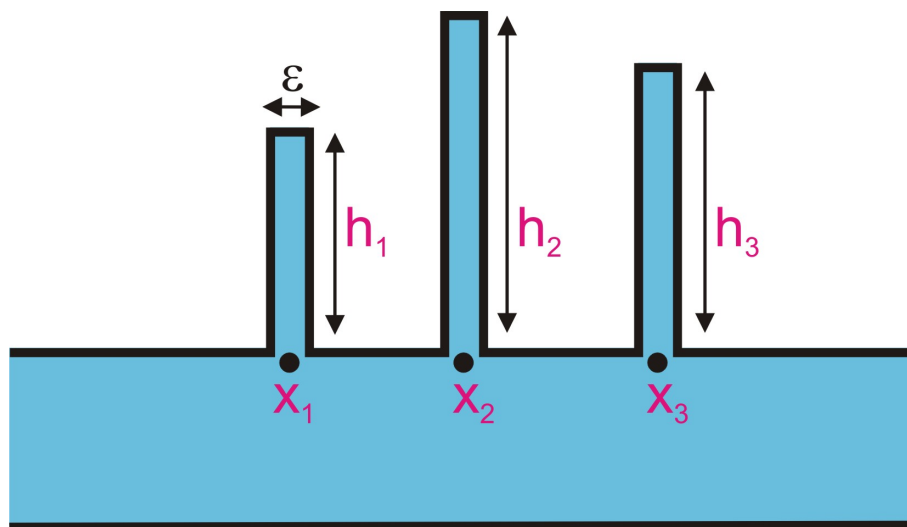


FIGURE 6 – Guide perturbé par des cheminées.

Dans un guide d'ondes, on peut aussi essayer de rendre invisible plusieurs objets et les disposant d'une certaine manière. Par exemple on peut essayer de rendre invisible des mouches dans un guide d'ondes. Il a été prouvé qu'en les disposant d'une bonne manière, elle peuvent être invisible. Comme le dit Lucas Chesnel "Flying as a team, fly can be invisible!"...

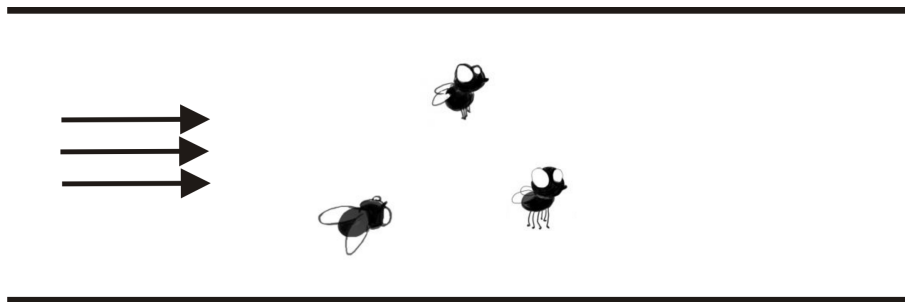


FIGURE 7 – "Flying as a team, fly can be invisible!".

Un dernier point de recherche, que je serai rapidement amené à étudier, est la montée en fréquence, ainsi que le cas multimodal. Ainsi on étudiera l'invisibilité dans des cas où plusieurs modes peuvent de propager, dans un guide dans lequel l'onde incidente se décompose selon plusieurs modes...

2.4 **Changement de point de vue**

Une autre idée peut être de changer totalement de point de vue : ici nous fixons une fréquence et nous cherchons un défaut invisible, mais on pourrait fixer une perturbation et chercher à savoir s'il y a oui ou non des fréquences pour lesquelles la perturbation est invisible, et si oui, comment les calculer.

Ce sujet est en cours d'étude depuis peu, et nous pensons étudier informatiquement un potentiel lien entre les fréquences d'invisibilité et une étude spectrale. Cette démarche devrait être faite rapidement.

2.5 **L'espace libre**

Le dernier cas d'invisibilité que je présenterai est le cas de l'espace libre. Nous avons vu que l'idée de nos études était de rendre invisible un objet dans un nombre fini de direction, sans nécessairement se placer dans le cas simple d'un guide d'ondes. A-S Bonnet-BenDhia, L. Chesnel et S.A. Nazarov ont travaillé sur ce sujet et ont montré l'existence de "petites" perturbations invisible en un nombre fini (données!) de direction.

Attention ! Il y a une direction pour laquelle il a été démontré qu'il est impossible d'obtenir invisibilité : la direction de propagation de l'onde incidente. Cela peut paraître contradictoire avec ce que l'on a affirmé pour le guide d'ondes, mais ca ne l'est pas puisque le guide d'ondes permet, grâce à ses bords, d'annuler cette direction si particulière.

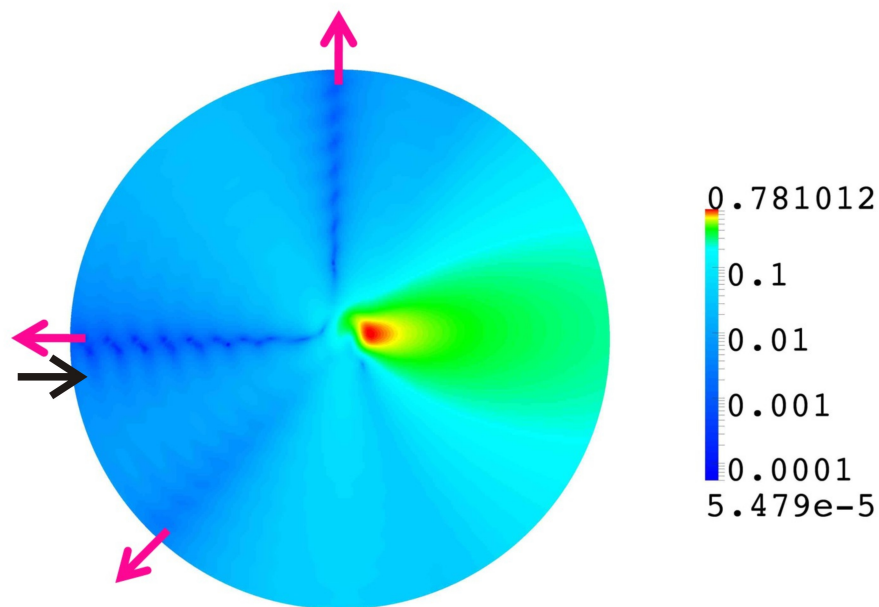


FIGURE 8 – Champ diffracté d'un obstacle invisible dans 3 directions.

Références

- [1] A.-S. BONNET-BEN DHIA ET E. LUNÉVILLE, 2015, *Propagation dans les guides d'ondes*.
- [2] A.-S. BONNET-BEN DHIA, L. CHESNEL ET S.A. NAZAROV, 2016, *Complete transmission invisibility for waveguides with sound hard walls*.
- [3] A.-S. BONNET-BEN DHIA ET S.A. NAZAROV, 2013, *Obstacles in acoustic waveguides becoming "invisible" at given frequencies*.
- [4] A.-S. BONNET-BEN DHIA, E. LUNÉVILLE, Y. MBEUTCHA ET S.A. NAZAROV, 2015, *A method to build non-scattering perturbations of two-dimensional acoustic waveguides*.
- [5] A.-S. BONNET-BEN DHIA, L. CHESNEL ET S.A. NAZAROV, 2015, *Non-scattering wave-numbers and far fields invisibility for a finite set of incident/scattering directions*.
- [6] E. BECACHE, A.-S. BONNET-BEN DHIA ET G. LEGENDRE, 2003, *Perfectly matched layers for the convected Helmholtz equation.*

3 Annexes

3.1 Projet L3 - Caractérisation de la norme L^p du gradient

Bera Antoine
Ndiaye Eugène

Mai 2013

Projet L3 MFA

Caractérisation de fonctions constantes et de la norme L^p du gradient

Encadré par Radu Ignat

Université Paris-Sud
Orsay

SOMMAIRE

- 1 Introduction
- 2 Norme L^p du gradient
- 3 Caractérisation des fonctions constantes
- 4 Généralisation de l'étude
- 5 Conclusion

Introduction :

Nous savons que l'on peut approcher la norme L^p d'une fonction $f \in C^2(\mathbb{R}^N, \mathbb{R})$ par $\|f * \rho_\epsilon\|_{L^p}$ lorsque $\epsilon \rightarrow 0$ où $(\rho_\epsilon)_{\epsilon \in \mathbb{R}^+}$ est une approximation de l'unité. Cette étude portera, de manière analogue, sur l'approximation de $\|\nabla f\|_{L^p}$ par

$$\iint_{\mathbb{R}^N \times \mathbb{R}^N} \frac{|f(x) - f(y)|^p}{|x - y|^p} \rho_\epsilon(|x - y|) dx dy$$

Mais peut-on passer à la limite et si elle existe, en quoi caractérise-t-elle la norme L^p du gradient? Enfin, nous généraliserons ces théorèmes en changeant $\frac{|f(x) - f(y)|^p}{|x - y|^p}$ par $\omega\left(\frac{|f(x) - f(y)|}{|x - y|}\right)$ pour certaines fonctions $\omega : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ dans \mathcal{W} , où $\mathcal{W} = \left\{ \omega \in C^0(\mathbb{R}^+, \mathbb{R}^+) \mid \omega(0) = 0 \text{ et } \omega(t) > 0 \ \forall t > 0 \right\}$. Pour tout $\omega \in \mathcal{W}$ nous obtenons une formule intégrale plus générale que la norme L^p du gradient de f . Cela en remarquant que l'on peut prendre $\omega(t) = t^p$ pour retrouver la formule de la norme L^p du gradient.

Cette étude utilisera des approximations de l'unité dont nous rappelons la définition : Soit une suite de fonctions $(\rho_{\epsilon_j})_{j \in \mathbb{N}}$ de $L^1(\mathbb{R}^N)$, on l'appelle approximation de l'unité lorsqu'elle vérifie les 4 propriétés suivantes :

$$\rho_{\epsilon_j} \geq 0 \quad , \quad \lim_{j \rightarrow +\infty} \rho_{\epsilon_j} = 0 \quad , \quad \int_{\mathbb{R}^N} \rho_{\epsilon_j} = 1 \quad \text{et} \quad \forall \delta > 0 \quad \lim_{j \rightarrow +\infty} \int_{|h| > \delta} \rho_{\epsilon_j}(h) dh = 0$$

Ces approximations de l'unité sont intéressantes puisque lorsque j est assez grand, elles approximent une masse de Dirac en 0. Elles permettent donc par convolution de ne garder que la fonction au voisinage d'un certain point. c'est ainsi que l'intégrale $\int_{\mathbb{R}^N} \frac{|f(x) - f(y)|^p}{|x - y|^p} \rho_\epsilon(|x - y|) dy$ sera proche d'une constante fois $\frac{|f(x) - f(y)|^p}{|x - y|^p}$ pour y proche de x qui est une approximation de $|\nabla f(x)|^p$. Ce qui nous a permis d'avoir cette idée de limite comme approche du gradient.

Il restera alors à déterminer cette constante, ce qui se fera grâce à une formule de changement de variable :

$$\int_{B(x, r_0)} f(y) dy = \int_0^{r_0} \int_{\mathbb{S}^{N-1}(x, r)} f(r, \gamma) d\sigma(\gamma) dr$$

Proposition 1

Soit (ρ_{ϵ_j}) une approximation de l'unité, pour tout $a > 0$,

$$\lim_{j \rightarrow +\infty} \int_{|h| \leq a} |h| \rho_{\epsilon_j}(h) dh = 0$$

Preuve :

Soit $0 < \epsilon < a$, on a :

$$\begin{aligned} \int_{|h| \leq a} |h| \rho_{\epsilon_j}(h) dh &= \int_{|h| \leq \epsilon} |h| \rho_{\epsilon_j}(h) dh + \int_{\epsilon < |h| \leq a} |h| \rho_{\epsilon_j}(h) dh \\ &\leq \epsilon \cdot \int_{|h| \leq \epsilon} \rho_{\epsilon_j}(h) dh + a \cdot \int_{|h| > \epsilon} \rho_{\epsilon_j}(h) dh \end{aligned}$$

Or

$$\int_{|h| \leq \epsilon} \rho_{\epsilon_j}(h) dh \leq \int_{\mathbb{R}^N} \rho_{\epsilon_j}(h) dh = 1$$

$$\lim_{j \rightarrow +\infty} \int_{|h| > \epsilon} \rho_{\epsilon_j}(h) dh = 0$$

Donc quelque soit ϵ assez petit,

$$0 \leq \overline{\lim}_{j \rightarrow +\infty} \int_{|h| \leq a} |h| \rho_{\epsilon_j}(h) dh \leq \epsilon.$$

D'où le résultat de la Proposition.

Théorème 1 (Norme L^p du gradient)

Soit $g \in C^2(\mathbb{R}^N)$ et (ρ_ϵ) une suite régularisante tel que :

$$(1) \iint_{\mathbb{R}^N \times \mathbb{R}^N} \frac{|g(x) - g(y)|^p}{|x - y|^p} \rho_\epsilon(|x - y|) dx dy \leq C \quad \text{quand } \epsilon \rightarrow 0$$

Alors :

$$\int_{\mathbb{R}^N} |\nabla g|^p dx \leq \frac{C}{K_{p,N}} \quad \text{avec } K_{p,N} = \int_{\mathbb{S}^{N-1}} |\sigma \cdot e|^p d\sigma$$

Et :

$$\lim_{\epsilon \rightarrow 0} \int_{\mathbb{R}^N} \int_{\mathbb{R}^N} \frac{|g(x) - g(y)|^p}{|x - y|^p} \rho_\epsilon(|x - y|) dx dy = K_{p,N} \int_{\mathbb{R}^N} |\nabla g|^p dx$$

Preuve :

Soit K un compact convexe de \mathbb{R}^N , $x \in K$ et $|h| \leq 1$.

Un développement de Taylor avec reste intégral donne :

$$g(x+h) - g(x) - \nabla g(x).h = \int_0^1 \frac{1-t}{1!} \nabla^{(2)} g(x+th) h^2 dt$$

$$|g(x+h) - g(x) - \nabla g(x).h| \leq |h|^2 \int_0^1 |(1-t) \nabla^{(2)} g(x+th)| dt$$

Comme $g \in C^2$, $\nabla^{(2)} g$ est continue sur le compact K donc il y atteint ses bornes. D'où il existe une constante C_K tel que :

$$|g(x+h) - g(x) - \nabla g(x).h| \leq C_K |h|^2$$

$$||\nabla g(x).h| - |g(x+h) - g(x)|| \leq |\nabla g(x).h - g(x+h) + g(x)| \leq C_K |h|^2$$

$$\Rightarrow |\nabla g(x).h| \leq |g(x+h) - g(x)| + C_K |h|^2$$

Comme $p \geq 1$, on a $\frac{1}{p} \leq 1 < (1+\theta)$ avec $\theta > 0$

$$\Rightarrow |\nabla g(x).h|^p \leq (1+\theta) |g(x+h) - g(x)|^p + C_{\theta,K} |h|^{2p}$$

$$(2) \quad |\nabla g(x).h|^p \leq (1+\theta) |g(x+h) - g(x)|^p + C_{\theta,K} |h|^{2p}$$

De plus, avec le changement de variable $\begin{cases} h = y - x \\ x \in K \text{ et } |h| \leq 1 \end{cases}$

dans

$$(1) \iint_{\mathbb{R}^N \times \mathbb{R}^N} \frac{|g(x) - g(y)|^p}{|x - y|^p} \rho_\epsilon(|x - y|) dx dy \leq C$$

,
on obtient :

$$(1) \Rightarrow \int_K dx \int_{|h| \leq 1} \frac{|g(x+h) - g(x)|^p}{|h|^p} \rho_\epsilon(|h|) dh \leq C$$

En appliquant à (2) cette dernière inégalité, on obtient :

$$\begin{aligned} \int_K dx \int_{|h| \leq 1} \frac{|\nabla g(x).h|^p}{|h|^p} \rho_\epsilon(|h|) dh &\leq (1+\theta) \underbrace{\int_K dx \int_{|h| \leq 1} \frac{|g(x+h) - g(x)|^p}{|h|^p} \rho_\epsilon(|h|) dh}_{\leq C} \\ &+ C_{\theta,K} \underbrace{\int_K dx \int_{|h| \leq 1} |h|^p \rho_\epsilon(|h|) dh}_{\substack{=|K| \\ \xrightarrow{\epsilon \rightarrow 0} 0}} \end{aligned}$$

Or un changement de variable radiale nous donne :

$$\int_{|h|\leq 1} \frac{|\nabla g(x).h|^p}{|h|^p} \rho_\epsilon(|h|) dh = K_{p,N} |\nabla g(x)|^p \underbrace{\int_0^1 \rho_\epsilon(r) r^{N-1} dr}_{=1}$$

$$\Rightarrow (3) \int_{|h|\leq 1} \frac{|\nabla g(x).h|^p}{|h|^p} \rho_\epsilon(|h|) dh = K_{p,N} |\nabla g(x)|^p$$

D'où

$$\boxed{K_{p,N} \int_K |\nabla g(x)|^p dx \leq (1 + \theta) C}$$

En reprenant (2) $|\nabla g(x).h|^p \leq (1 + \theta) |g(x+h) - g(x)|^p + C_{\theta,K} |h|^{2p}$ et intégrant on obtient :

$$\begin{aligned} \int_K dx \int_{|h|\leq 1} \frac{|\nabla g(x).h|^p}{|h|^p} \rho_\epsilon(|h|) dh &\leq \\ &\underbrace{= (3) K_{p,N} |\nabla g(x)|^p}_{\leq (1 + \theta) \int_K dx \int_{|h|\leq 1} \frac{|g(x+h) - g(x)|^p}{|h|^p} \rho_\epsilon(|h|) dh} \\ &+ C_{\theta,K} \int_K dx \int_{|h|\leq 1} \frac{|h|^{2p}}{|h|} \rho_\epsilon(|h|) dh \end{aligned}$$

D'après la proposition 1,

$$\lim_{\epsilon \rightarrow 0} \int_{|h|\leq 1} |h| \rho_\epsilon(|h|) dh = 0$$

Donc

$$\Rightarrow (4) \quad K_{p,N} \int_{\mathbb{R}^N} |\nabla g(x)|^p dx \leq \liminf_{\epsilon \rightarrow 0} \iint_{\mathbb{R}^N} \frac{|g(x) - g(y)|^p}{|x - y|^p} \rho_\epsilon(|x - y|) dx dy$$

En procédant de la même manière et, en échangeant les rôles de $|g(x+h) - g(x)|$ et $|\nabla g(x).h|$ dans (2), on obtient :

$$|g(x+h) - g(x)|^p \leq (1+\theta)|\nabla g(x).h|^p + C'_\theta|h|^{2p}$$

pour x et $x+h \in \text{supp}(g)$:

$$\begin{aligned} \limsup_{\epsilon \rightarrow 0} \int_{\mathbb{R}^N} dx \int_{\mathbb{R}^N} \frac{|g(x+h) - g(x)|^p}{|h|^p} \rho_\epsilon(|h|) dh \leq \\ (1+\theta) \int_K K_{p,N} |\nabla g(x)|^p dx \\ + C'_\theta |\text{supp}(g)| \underbrace{\int_{|h| \leq 1} |h|^p \rho_\epsilon(|h|) dh}_{\xrightarrow[\epsilon \rightarrow 0]{0}} \end{aligned}$$

Enfin lorsque $\theta \rightarrow 0$,

$$(5) \quad \limsup_{\epsilon \rightarrow 0} \int_{\mathbb{R}^N} dx \int_{\mathbb{R}^N} \frac{|g(x+h) - g(x)|^p}{|h|^p} \rho_\epsilon(|h|) dh \leq K_{p,N} \int_K |\nabla g(x)|^p dx$$

En combinant les inégalités (4) et (5) on obtient alors le théorème 1.

$$\lim_{\epsilon \rightarrow 0} \int_{\mathbb{R}^N} \int_{\mathbb{R}^N} \frac{|g(x) - g(y)|^p}{|x - y|^p} \rho_\epsilon(|x - y|) dx dy = K_{p,N} \int_{\mathbb{R}^N} |\nabla g|^p dx$$

Théorème 2 (Caractérisation des fonctions constantes)

Soit $\Omega \subset \mathbb{R}^N$ un ouvert connexe et $f : \Omega \rightarrow \mathbb{R}$ une fonction mesurable telle que :

$$\iint_{\Omega \times \Omega} \frac{|f(x) - f(y)|^p}{|x - y|^p} \frac{dxdy}{|x - y|^N} < +\infty$$

alors f est constante presque-partout dans Ω .

Preuve :

Soit $x_0 \in \Omega$ et $r > 0$ tel que $B(x_0, 3r) \subset \Omega$. On note $B_1 = B(x_0, r)$ et $B_2 = B(x_0, 2r)$.

On définit $f_n := \rho_n * f$ où ρ_n est une suite régularisante telle que $\text{supp} \rho_n \subset B(0, \frac{1}{n})$.

On a :

$$|f_n(x) - f_n(y)| = \left| \int_{B(0, \frac{1}{n})} f(x-z)\rho_n(z)dz - \int_{B(0, \frac{1}{n})} f(y-z)\rho_n(z)dz \right| \leq \int_{B_2} |f(x-z) - f(y-z)|\rho_n(z)dz$$

Soit la fonction convexe $w : t \rightarrow t^p$, $p \geq 1$. Par l'inégalité de Jensen, on obtient :

$$w\left(\frac{|f_n(x) - f_n(y)|}{|x - y|}\right) \leq \int_{B_2} \rho_n(z)w\left(\frac{|f(x-z) - f(y-z)|}{|x - y|}\right) dz$$

Puis, en intégrant :

$$\int_{B_2} \int_{B_2} \frac{|f_n(x) - f_n(y)|^p}{|x - y|^p} \frac{dxdy}{|x - y|^N} \leq \int_{B_2} \int_{B_2} \left(\int_{B(0, \frac{1}{n})} \rho_n(z) \frac{|f(x-z) - f(y-z)|^p}{|x - y|^p} \right) \frac{dxdy}{|x - y|^N}$$

Par changement de variable de translation de z , pour x et y on trouve :

$$\leq \int_{B(0, \frac{1}{n})} \rho_n(z) \left(\int_{B_2-z} \int_{B_2-z} \frac{|f(x) - f(y)|^p}{|x - y|^p} \frac{dxdy}{|x - y|^N} \right) dz$$

En majorant la double intégrale interne par $\int_{\Omega} \int_{\Omega} \frac{|f_n(x) - f_n(y)|^p}{|x - y|^p} \frac{dxdy}{|x - y|^N}$ ($B_2 - z \subset \Omega$ par construction de B_2 et n assez grand) et en remarquant que $\int_{B(0, \frac{1}{n})} \rho_n(z) dz = 1$, on obtient :

$$\int_{B_2} \int_{B_2} \frac{|f_n(x) - f_n(y)|^p}{|x - y|^p} \frac{dxdy}{|x - y|^N} \leq \int_{\Omega} \int_{\Omega} \frac{|f(x) - f(y)|^p}{|x - y|^p} \frac{dxdy}{|x - y|^N} < +\infty$$

Or d'après le théorème précédent, on a :

$$\int_{\mathbb{R}^N} \int_{\mathbb{R}^N} \frac{|f_n(x) - f_n(y)|^p}{|x - y|^p} \rho_\epsilon(|x - y|) dxdy \xrightarrow{\epsilon \rightarrow 0} K_{p,N} \int_{\mathbb{R}^N} |\nabla f_n|^p dx$$

On va donc considérer la fonction

$$\rho_\epsilon(r) = \begin{cases} \frac{\epsilon}{r^{N-\epsilon}} & \text{si } r < 1 \\ 0 & \text{sinon} \end{cases}$$

avec $\epsilon > 0$ de sorte qu'elle soit une approximation de l'unité.

En effet, on a :

$$\begin{aligned} \rho_\epsilon &\geq 0 \quad \text{et} \quad \forall \quad 0 < \delta < 1, \int_\delta^1 \frac{\epsilon}{r^{N-\epsilon}} < +\infty \\ \int_0^{+\infty} \rho_\epsilon(r) r^{N-1} dr &= \int_0^1 \frac{\epsilon}{r^{N-\epsilon}} r^{N-1} dr = \frac{\epsilon}{\epsilon} = 1 \\ \forall \delta > 0, \int_\delta^{+\infty} \rho_\epsilon(r) r^{N-1} dr &= \epsilon \int_\delta^1 \frac{r^{N-1}}{r^{N-\epsilon}} dr = (1 - r^\epsilon) \rightarrow_{\epsilon \rightarrow 0} 0 \end{aligned}$$

Maintenant écrivons

$$\int_{B_1} \int_{B_1} \frac{|f_n(x) - f_n(y)|^p}{|x - y|^p} \rho_\epsilon(|x - y|) dx dy \leq \epsilon \int_{B_2} \int_{B_2} \frac{|f_n(x) - f_n(y)|^p}{|x - y|^p} \frac{dx dy}{|x - y|^N}$$

Comme cette dernière intégrale est finie, il existe une constante C telle que :

$$K_{p,N} \int_{\mathbb{R}^N} |\nabla f_n|^p dx = \int_{B_1} \int_{B_1} \frac{|f_n(x) - f_n(y)|^p}{|x - y|^p} \rho_\epsilon(|x - y|) dx dy \leq C\epsilon \rightarrow_{\epsilon \rightarrow 0} 0$$

Donc,

$$\int_{\mathbb{R}^N} |\nabla f_n|^p dx = 0 \Rightarrow |\nabla f_n|^p = 0 \quad p.p \quad x \in \Omega$$

On peut alors conclure que f_n est constante $p.p \quad x \in \Omega$.

Et enfin

$$\|f - f_n\|_{L^p} \xrightarrow{n \rightarrow +\infty} 0$$

D'où, f est aussi constante $p.p \quad x \in \Omega$.

Théorème 3

Soit $\Omega \subset \mathbb{R}^N$ une partie bornée et $w : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ une fonction continue. Si $f \in C^2(\overline{\Omega})$, alors :

$$\lim_{j \rightarrow +\infty} \int_{\Omega} \int_{\Omega} w \left(\frac{|f(x) - f(y)|}{|x - y|} \right) \rho_{\epsilon_j}(|x - y|) dx dy = \int_{\Omega} \left(\int_{\mathbb{S}^{N-1}(0,1)} w(|\nabla f(x) \cdot \sigma|) d\mu(\sigma) \right) dx$$

Preuve :

$\forall x \in \Omega, \nabla f(x) \in \mathcal{L}(\mathbb{R}^N, \mathbb{R}^N)$. Comme $f \in C^2$, alors $\nabla f \in C^1(\Omega, \mathcal{L}(\mathbb{R}^N, \mathbb{R}^N))$.

De plus, $x \rightarrow \|\nabla f(x)\|$ atteint son maximum sur $\overline{\Omega}$ noté M_0 .

On a :

$$\begin{aligned} f(x) &= f(y) + \nabla f(x) \cdot (x - y) + (x - y) \cdot \xi(|x - y|) \forall x, y \in \Omega \\ &\Rightarrow |f(x) - f(y)| \leq M + \xi(|x - y|) \end{aligned}$$

avec $\xi(|x - y|) = \frac{|f(x) - f(y) - \nabla f(x) \cdot (x - y)|}{|x - y|}$ donc $(x, y) \rightarrow \xi(|x - y|)$ est continue sur $\overline{\Omega}^2$.

Il existe donc M_f tel que $\forall x, y \in \overline{\Omega}, \frac{|f(x) - f(y)|}{|x - y|} \leq M_f$ ($M \leq M_f$).

Proposition 2

Soit $w : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ uniformément continue sur $[0, M_f]$, alors :

$$\forall \delta > 0, \exists C_\delta > 0 : \forall x, y \in [0, M_f], |w(x) - w(y)| \leq \delta + C_\delta |x - y|$$

Preuve :

Renotons la définition de l'uniforme continuité :

$$\forall \delta > 0, \exists \eta > 0, \quad x, y \in [0, M_f], \quad |x - y| \leq \eta \Rightarrow |w(x) - w(y)| \leq \delta$$

Si $x, y \in [0, M_f]$ et $|x - y| > \eta$ en utilisant la continuité de w , il existe $A_{\delta, \eta} > 0$ tel que $|w(x) - w(y)| \leq A_{\delta, \eta} \leq \frac{A_{\delta, \eta}}{\eta} \cdot \eta \leq \frac{A_{\delta, \eta}}{\eta} \cdot |x - y|$.

En posant $C_\delta := A_{\delta, \eta}$ (car η ne dépend que de δ) on obtient :

$$|w(x) - w(y)| \leq \delta \cdot \chi_{|x - y| \leq \eta} + C_\delta \cdot |x - y| \chi_{|x - y| > \eta} \leq \delta + C_\delta \cdot |x - y|$$

Utilisons maintenant cette propriété avec $0 \leq \frac{|f(x) - f(y)|}{|x - y|} \leq M_f$ et $0 \leq \left| \nabla f(x) \cdot \frac{x - y}{|x - y|} \right| \leq M_0 \cdot \frac{|x - y|}{|x - y|} \leq M_f$

$\forall x, y \in \Omega,$

$$\begin{aligned} \left| w \left(\frac{|f(x) - f(y)|}{|x - y|} \right) - w \left(\left| \nabla f(x) \cdot \frac{x - y}{|x - y|} \right| \right) \right| &\leq \\ &\leq \delta + C_\delta \left| \frac{f(x) - f(y) - \nabla f(x) \cdot (x - y)}{x - y} \right| \leq \delta + C_\delta \cdot |x - y| \end{aligned}$$

Nous disposons donc d'une inégalité intégrable qui va permettre de donner cette limite.

Posons

$$\Delta_{\epsilon_j} := \left| \int_{\Omega} \int_{\Omega} w \left(\frac{|f(x) - f(y)|}{|x - y|} \right) \rho_{\epsilon_j}(|x - y|) dx dy - \int_{\Omega} \int_{\Omega} w \left(\left| \nabla f(x) \cdot \frac{x - y}{|x - y|} \right| \right) \rho_{\epsilon_j}(|x - y|) dx dy \right|$$

Par l'inégalité triangulaire on a :

$$\Delta_{\epsilon_j} \leq \int_{\Omega} \int_{\Omega} \left| w \left(\frac{|f(x) - f(y)|}{|x - y|} \right) - w \left(\left| \nabla f(x) \cdot \frac{x - y}{|x - y|} \right| \right) \right| \rho_{\epsilon_j}(|x - y|) dx dy$$

Nous pouvons maintenant séparer en 2 notre espace, pour utiliser 2 propriétés différentes.

$$\begin{aligned} \Delta_{\epsilon_j} &\leq \int_{\Omega} \int_{\Omega} \chi_{|x-y| \leq 1} (\delta + C_\delta \cdot |x - y|) \rho_{\epsilon_j}(|x - y|) dx dy \\ &\quad + \int_{\Omega} \int_{\Omega} \chi_{|x-y| > 1} \left| w \left(\frac{|f(x) - f(y)|}{|x - y|} \right) - w \left(\left| \nabla f(x) \cdot \frac{x - y}{|x - y|} \right| \right) \right| \rho_{\epsilon_j}(|x - y|) dx dy \\ &\leq C_\delta \int_{\Omega} \int_{\Omega} \chi_{|x-y| \leq 1} |x - y| \rho_{\epsilon_j}(|x - y|) dx dy \quad + \quad \delta \int_{\Omega} \int_{\Omega} \chi_{|x-y| \leq 1} \rho_{\epsilon_j}(|x - y|) dx dy \\ &\quad + \quad 2 \max_{[0, M_f]} |w| \int_{\Omega} \int_{\Omega} \chi_{|x-y| > 1} \rho_{\epsilon_j}(|x - y|) dx dy \end{aligned}$$

De plus, par le changement de variable $\begin{cases} u = x - y \\ v = x + y \end{cases}$ On obtient les 3 inégalités suivantes :

$$(1) \int_{\Omega} \int_{\Omega} \chi_{|x-y| \leq 1} |x - y| \rho_{\epsilon_j}(|x - y|) dx dy = \int_{\Omega} \int_{\Omega} \chi_{|u| \leq 1} |u| \rho_{\epsilon_j}(|u|) \frac{1}{2} du dv$$

$$= |\Omega| \frac{1}{2} \int_{\Omega} \chi_{|u| \leq 1} |u| \rho_{\epsilon_j}(|u|) du$$

$$(2) \int_{\Omega} \int_{\Omega} \chi_{|x-y| \leq 1} \rho_{\epsilon_j}(|x - y|) dx dy = \frac{1}{2} |\Omega| \int_{\Omega} \chi_{|u| \leq 1} \rho_{\epsilon_j}(|u|) du dv \leq \frac{|\Omega|}{2}$$

$$(3) \int_{\Omega} \int_{\Omega} \chi_{|x-y| > 1} \rho_{\epsilon_j}(|x - y|) dx dy = \frac{1}{2} |\Omega| \int_{\Omega} \chi_{|u| > 1} \rho_{\epsilon_j}(|u|) du$$

Donc

$$\Delta_{\epsilon_j} \leq |\Omega| \left[\frac{C_\delta}{2} \int_{\Omega} \chi_{|u| \leq 1} |u| \rho_{\epsilon_j}(|u|) du + \frac{\delta}{2} + \max_{[0, M_f]} |w| \cdot \int_{\Omega} \chi_{|u| > 1} \rho_{\epsilon_j}(|u|) du \right]$$

Parmi les trois termes de l'inégalité, le dernier tend vers 0 quand j tend vers $+\infty$ par définition de $(\rho_{\epsilon_j})_{j \in \mathbb{N}}$, ainsi que le premier terme d'après la Proposition 1.

Donc $\forall \delta > 0$,

$$0 \leq \overline{\lim}_{j \rightarrow +\infty} \Delta_{\epsilon_j} \leq \frac{|\Omega|}{2} \cdot \delta \quad \rightarrow_{\delta \rightarrow 0} 0$$

Finalement,

$$\lim_{j \rightarrow +\infty} \Delta_{\epsilon_j} = 0$$

$$\Rightarrow \boxed{\lim_{j \rightarrow +\infty} \int_{\Omega} \int_{\Omega} \left| w \left(\frac{|f(x) - f(y)|}{|x - y|} \right) - w \left(\left| \nabla f(x) \cdot \frac{x - y}{|x - y|} \right| \right) \right| \rho_{\epsilon_j}(|x - y|) dx dy = 0}$$

Nous savons donc que $\Delta_{\epsilon_j} \rightarrow_{j \rightarrow +\infty} 0$, et nous cherchons à étudier la limite du terme de gauche de Δ_{ϵ_j} . Intéressons nous donc au terme de droite, qui aura (si elle existe) la même limite que celui de gauche.

Écrivons

$$\int_{\Omega} \int_{\mathbb{R}^N} w \left(\left| \nabla f(x) \cdot \frac{h}{|h|} \right| \right) \rho_{\epsilon_j}(h) dx dh = \int_{\Omega} \int_{\mathbb{R}^N} w \left(\left| \nabla f(x) \cdot \frac{x - y}{|x - y|} \right| \right) \rho_{\epsilon_j}(x - y) dx dy$$

(par le changement de variable $y = x - h$)

$$= \int_{\Omega} \int_{\Omega} w \left(\left| \nabla f(x) \cdot \frac{x - y}{|x - y|} \right| \right) \rho_{\epsilon_j}(x - y) dx dy + \underbrace{\int_{\Omega} \int_{\mathbb{R}^N \setminus \Omega} w \left(\left| \nabla f(x) \cdot \frac{x - y}{|x - y|} \right| \right) \rho_{\epsilon_j}(x - y) dx dy}_{= \alpha_j}$$

$$|\alpha_j| \leq \max_{[0, M_f]} |w| \cdot \int_{\Omega} \int_{\mathbb{R}^N \setminus \Omega} \rho_{\epsilon_j}(x - y) dx dy$$

Soit $\beta > 0$, posons $\Omega_\beta = \{x \in \Omega \mid \text{dist}(x, \partial\Omega)\}$

En effectuant un changement de variable, après une séparation des variables de cette intégrale on obtient :

$$|\alpha_j| \leq \max_{[0, M_f]} |\omega| \cdot \left[|\Omega \setminus \Omega_\beta| \underbrace{\int_{|u| \leq \beta} \rho_{\epsilon_j}(u) du}_{\leq 1} + |\Omega| \int_{|u| > \beta} \rho_{\epsilon_j}(u) du \right]$$

et par passage à la limite en utilisant une propriété des approximatins de l'unité,

$$0 \leq \overline{\lim}_{j \rightarrow +\infty} |\alpha_j| \leq \max_{[0, M_f]} |\omega| \cdot |\Omega \setminus \Omega_\beta|$$

mais $\lim_{\beta \rightarrow 0} |\Omega \setminus \Omega_\beta| = 0$ car Ω est borné. Donc $\lim_{j \rightarrow +\infty} |\alpha_j| = 0$

Utilisons maintenant le changement de variable proposé en introduction à $f \in \mathcal{C}^0(\mathbb{S}^{N-1})$

$$\int_{\mathbb{R}^N} f\left(\frac{h}{|h|}\right) \rho_{\epsilon_j}(h) dh = \int_0^{+\infty} \int_{\mathbb{S}^{N-1}(0,r)} f\left(\frac{\gamma}{|\gamma|}\right) \rho_{\epsilon_j}(\gamma) d\sigma(\gamma) dr$$

On remarque que $|\gamma| = r$ donc en posant $\theta = \frac{\gamma}{r} \in \mathbb{S}^{N-1}(0,1)$ et en utilisant Fubini, on obtient :

$$\int_{\mathbb{R}^N} f\left(\frac{h}{|h|}\right) \rho_{\epsilon_j}(h) dh = \int_{\mathbb{S}^{N-1}(0,1)} f(\theta) \left(\int_0^{+\infty} \rho_{\epsilon_j}(r \cdot \theta) dr \right) d\theta$$

Cela nous fournit donc :

$$\begin{aligned} \int_{\Omega} \int_{\mathbb{R}^N} w\left(\left|\nabla f(x) \cdot \frac{h}{|h|}\right|\right) \rho_{\epsilon_j}(h) dx dh &= \int_{\Omega} \left(\int_{\mathbb{S}^{N-1}(0,1)} w(|\nabla f(x) \cdot \theta|) \left(\int_0^{+\infty} \rho_{\epsilon_j}(r \cdot \theta) dr \right) d\theta \right) dx \\ &= \int_{\Omega} \left(\int_{\mathbb{S}^{N-1}(0,1)} w(|\nabla f(x) \cdot \theta|) d\mu_{\epsilon_j}(\theta) \right) dx \\ &\xrightarrow{j \rightarrow +\infty} \int_{\Omega} \left(\int_{\mathbb{S}^{N-1}(0,1)} w(|\nabla f(x) \cdot \theta|) d\mu(\theta) \right) dx \end{aligned}$$

Ainsi le terme de droite dans Δ_{ϵ_j} admet une limite avec j , qui est l'expression précédente. Cela conclut donc la preuve du théorème 3.

Revenons maintenant à l'ensemble $\mathcal{W} = \{\omega \in C^0(\mathbb{R}^+, \mathbb{R}^+) \mid \omega(0) = 0 \text{ et } \omega(t) > 0 \ \forall t > 0\}$ et à une généralisation du théorème 1.

Théorème 4 (Généralisation du Théorème 1)

Soit $\Omega \subset \mathbb{R}^N$ un ouvert connexe. Soit $w \in \mathcal{W}$ telle que $\liminf_{t \rightarrow +\infty} \frac{w(t)}{t} > 0$

Soit $f : \Omega \rightarrow \mathbb{R}$ une fonction mesurable telle que :

$$\iint_{\Omega \times \Omega} w \left(\frac{|f(x) - f(y)|}{|x - y|} \right) \frac{dx dy}{|x - y|^N} < +\infty$$

alors f est constante presque-partout dans Ω .

Preuve :

Nous reprenons ici la démonstration du Théorème 1, et on se rappelle qu'une hypothèse important de la démonstration était la convexité de $t \rightarrow t^p$. Supposons tout d'abord que w est convexe, argument que nous justifierons à la fin de la démonstration.

Soit $x_0 \in \Omega$ et $r > 0$ tel que $B(x_0, 3r) \subset \Omega$. On note $B_1 = B(x_0, r)$ et $B_2 = B(x_0, 2r)$ et on pose $f_n := \rho_n * f$.

On a, par positivité de ρ_n :

$$\begin{aligned} |f_n(x) - f_n(y)| &= \left| \int_{B(0, \frac{1}{n})} f(x-z)\rho_n(z)dz - \int_{B(0, \frac{1}{n})} f(y-z)\rho_n(z)dz \right| \\ &\leq \int_{B(0, \frac{1}{n})} |f(x-z) - f(y-z)|\rho_n(z)dz \end{aligned}$$

La fonction w étant convexe, on applique l'inégalité de Jensen :

$$w \left(\frac{|f_n(x) - f_n(y)|}{|x - y|} \right) \leq \int_{B(0, \frac{1}{n})} \rho_n(z) w \left(\frac{|f(x) - f(y)|}{|x - y|} \right) dz$$

Puis, en intégrant sur $B_2 \times B_2$:

$$\int_{B_2} \int_{B_2} w \left(\frac{|f_n(x) - f_n(y)|}{|x - y|} \right) \frac{dx dy}{|x - y|^N} \leq \int_{B_2} \int_{B_2} \left(\int_{B(0, \frac{1}{n})} \rho_n(z) \cdot w \left(\frac{|f(x-z) - f(y-z)|}{|x - y|} \right) \right) \frac{dx dy}{|x - y|^N}$$

Par changement de variable de translation de z , pour x et y on trouve :

$$\leq \int_{B(0, \frac{1}{n})} \rho_n(z) \left(\int_{B_2-z} \int_{B_2-z} w \left(\frac{|f(x) - f(y)|}{|x - y|} \right) \frac{dx dy}{|x - y|^N} \right) dz$$

En majorant la double intégrale interne par $\int_{\Omega} \int_{\Omega} w \left(\frac{|f(x) - f(y)|}{|x - y|} \right) \frac{dx dy}{|x - y|^N}$ ($B_2 - z \subset \Omega$ par construction de B_2 et n assez grand) et en remarquant que $\int_{B(0, \frac{1}{n})} \rho_n(z) dz = 1$, on obtient :

$$\int_{B_2} \int_{B_2} w \left(\frac{|f_n(x) - f_n(y)|}{|x - y|} \right) \frac{dx dy}{|x - y|^N} \leq \int_{\Omega} \int_{\Omega} w \left(\frac{|f(x) - f(y)|}{|x - y|} \right) \frac{dx dy}{|x - y|^N}$$

Maintenant écrivons

$$\begin{aligned} \int_{B_2} \int_{B_2} w \left(\frac{|f_n(x) - f_n(y)|}{|x - y|} \right) \frac{dxdy}{|x - y|^N} &\geq \int_{B_1} \int_{B_1} w \left(\frac{|f_n(x) - f_n(y)|}{|x - y|} \right) \frac{dxdy}{|x - y|^N} \\ &\geq \int_{B_1} \left(\int_{B_1} w \left(\frac{|f_n(x) - f_n(y)|}{|x - y|} \right) \frac{dx}{|x - y|^N} \right) dy \\ &= \int_{B_1} \left(\int_{\mathbb{S}^{N-1}} \int_0^r w \left(\frac{|f_n(y + t.\sigma) - f_n(y)|}{|t.\sigma|} \right) \frac{t^{N-1} \cdot |\sigma| dt d\sigma}{|t.\sigma|^N} \right) dy \end{aligned}$$

Mais dans cette expression $|\sigma| = 1$ donc on obtient, par regroupement de toutes les inégalités, la minoration :

$$\int_{B_1} \left(\int_{\mathbb{S}^{N-1}} \left(\int_0^r w \left(\frac{|f_n(y + t.\sigma) - f_n(y)|}{t} \right) \frac{dt}{t} \right) d\sigma \right) dy < +\infty$$

On peut conclure que,

$$p.p. \quad y \in B_1 \quad \text{et} \quad \sigma \in \mathbb{S}^{N-1}, \quad \int_0^r w \left(\frac{|f_n(y + t.\sigma) - f_n(y)|}{t} \right) \frac{dt}{t} < +\infty$$

Comme, de plus $\int_0^r \frac{dt}{t} = +\infty$, on peut affirmer que

$$\lim inf_{t \rightarrow 0} w \left(\frac{|f_n(y + t.\sigma) - f_n(y)|}{t} \right) = 0 \quad p.p. \quad y \in B_1 \quad \text{et} \quad \sigma \in \mathbb{S}^{N-1}$$

En effet, supposons que $\lim inf_{t \rightarrow 0} w \left(\frac{|f_n(y + t.\sigma) - f_n(y)|}{t} \right) > 0$ (par positivité de w cette limite est positive ou nulle), alors il existerait une constante $c > 0$ telle que sur un voisinage $([0, a[$ avec $a > 0$) de 0, $w \left(\frac{|f_n(y + t.\sigma) - f_n(y)|}{t} \right) \geq c$. Alors $\int_0^r w \left(\frac{|f_n(y + t.\sigma) - f_n(y)|}{t} \right) \frac{dt}{t} = +\infty$, ce qui est contradictoire.

Maintenant, par continuité de w en 0 cela signifie que

$$w(|\nabla f_n(x) \cdot \sigma|) = 0 \quad p.p. \quad y \in B_1 \quad \text{et} \quad \sigma \in \mathbb{S}^{N-1}$$

On en déduit donc que f_n est constante sur B_1 . On peut donc conclure la démonstration du théorème lorsque w est supposé convexe.

Pour pouvoir achever la démonstration, il faut montrer la Proposition 3 qui permettra de procéder au même raisonnement mais avec \tilde{w} puisque

$$\iint_{\Omega \times \Omega} \tilde{w} \left(\frac{|f(x) - f(y)|}{|x - y|} \right) \frac{dxdy}{|x - y|^N} < \iint_{\Omega \times \Omega} w \left(\frac{|f(x) - f(y)|}{|x - y|} \right) \frac{dxdy}{|x - y|^N} < +\infty$$

On peut ainsi aboutir à la même conclusion mais avec \tilde{w} qui a les mêmes caractéristiques que w .

Proposition 3

Soit $w \in \mathcal{W} = \left\{ \omega \in C^0(\mathbb{R}^+, \mathbb{R}^+) \mid \omega(0) = 0 \text{ et } \omega(t) > 0 \quad \forall t > 0 \right\}$ telle que $\liminf_{t \rightarrow +\infty} \frac{w(t)}{t} > 0$

Alors il existe $\tilde{w} \in \mathcal{W}$ CONVEXE vérifiant $\forall t \geq 0, \quad \tilde{w}(t) \leq w(t)$

Preuve :

La fonction w considérée peut être très compliquée, de par sa régularité ou de sa façon de tendre vers 0. On considère ainsi la fonction de la figure 1 qui n'est pas dérivable en 0 et dont les minimums locaux convergent quadratiquement vers 0.

Essayons tout d'abord de construire \tilde{w} sur le segment $[0, 1]$. Posons $w_1 : t \mapsto \inf_{x \in [t, 1]} w(x)$ (bien définie et strictement positive car w est continue sur le compact $[t, 1]$, donc y atteint ses bornes.). La figure 2 montre la construction de w_1 pour notre exemple. w_1 est croissante par construction, est dans \mathcal{W} par la remarque précédente et minore w sur $[0, 1]$.

Nous pouvons donc restreindre l'étude à une fonction croissante de \mathcal{W} . Soit $a_0 \in [0, 1[$, et choisissons $a_1 \in [0, a_0[$. Définissons une fonction f_1 continue, convexe, nulle sur $[0, a_1]$, strictement positive sur $]a_1, 1]$ et étant un minorant de w sur $[0, 1]$. Cette fonction est bien définie sur $[a_1, 1]$ puisque w est strictement positive sur ce compact. De même définissons une fonction f_0 continue, convexe, nulle sur $[0, a_0]$ et strictement positive sur $]a_0, 1]$ et étant un minorant de f_1 sur $[0, 1]$. Cette fonction est bien définie sur $[a_0, 1]$ puisque f_1 est strictement positive sur ce compact. (f_1 et f_0 peuvent être construites simplement avec des fonctions affines)

Construisons par récurrence deux suites $(f_n)_{n \in \mathbb{N}}$ et $(a_n)_{n \in \mathbb{N}}$:

$a_{n+2} = \inf \{ t \in [0, a_n[\text{ tel que le segment } [(a_{n+2}, 0), (a_n, f_{n+1}(a_n))] \text{ soit en dessous de } w_1 \}$. Concrètement on part du point $(a_n, f_{n+1}(a_n))$ et on cherche quelle droite provenant de l'axe des abscisses va venir être tangente à w_1 , le point correspondant est a_{n+2} . Nous pouvons alors construire f_{n+2} :

$$f_{n+2} = 0 \quad \text{sur} \quad [0, a_{n+2}] \quad , \quad f_{n+2} = f_{n+1} \quad \text{sur} \quad [a_n, 1]$$

$$\text{et} \quad f_{n+2}(t) = -\frac{a_{n+2} \cdot f(a_n)}{a_n - a_{n+2}} + \frac{f(a_n)}{a_n - a_{n+2}} \cdot t \quad \text{sur} \quad [a_{n+2}, a_n]$$

f_{n+2} est une régression linéaire sur $[a_{n+2}, a_n]$ qui fera bien de f_{n+2} une fonction convexe puisque f_{n+1} est sous sa corde d'abscisse $[a_{n+2}, a_n]$. Cela se comprend bien sur la Figure 3. La suite (a_n) est strictement décroissante et minorée par 0 donc est convergente, cette limite ℓ vérifie $w(\ell) = 0$ (sinon $w(\ell) > 0$ et pour a_n et a_{n+2} assez proche de ℓ , a_{n+2} sera strictement inférieur à ℓ . Un dessin permet de s'en convaincre facilement) donc $\ell = 0$.

C'est ainsi que l'on définit une suite de fonctions, telles que les f_k seront égales sur $[a_n, 1]$ pour tout $k \geq n$. Définissons donc w ainsi :

soit $t \in]0, 1[$, $\exists n \geq 1$ tel que $a_n < t$

$$\tilde{w}(t) = f_n(t)$$

\tilde{w}_1 ainsi définie est dans \mathcal{W} , continue, convexe et minore de w sur $[0, 1]$. Prolongeons \tilde{w}_1 en prolongeant la droite définissant f_1 sur $[a_1, +\infty[$. L'hypothèse $\liminf_{t \rightarrow +\infty} \frac{w(t)}{t} > 0$, donne qu'il existe $A > 1$ et $b > 0$ tels que $w(t) > b.t$, $\forall t > A$.

Remarquons également que w est continue sur $[1, A]$ donc qu'elle est minorée et atteint sa borne inférieure, notée m . Et notons que si l'on multiplie \tilde{w}_1 par une constante $c \in]0, 1[$, la fonction est toujours dans \mathcal{W} et minorante de w sur $[0, 1]$. En choisissant bien cette constante pour que $\tilde{w} = c.\tilde{w}_1$ soit inférieure à m sur $[1, A]$, que $\tilde{w}(A) < b.A$ et enfin que la pente la droite définissant \tilde{w} sur $[a_1, +\infty[$ soit inférieure à b , alors \tilde{w} vérifie la conclusion de la Propriété 3.

Ce qui conclue la preuve de la proposition.

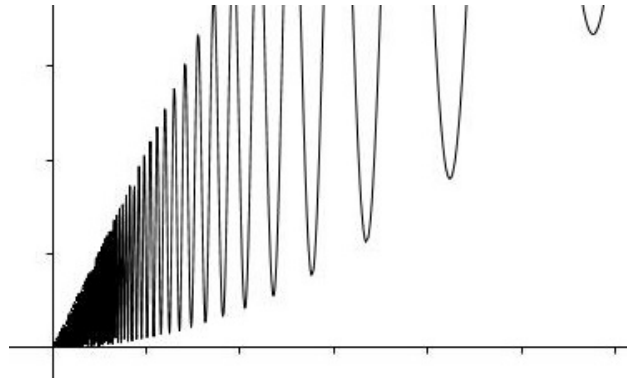


FIGURE 1 – Exemple de fonction w

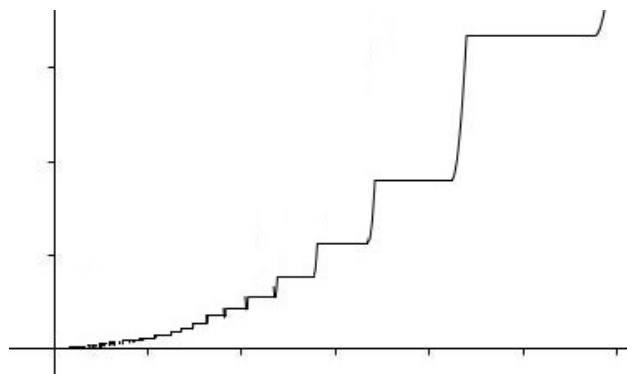


FIGURE 2 – Première modification de w pour la supposer croissante

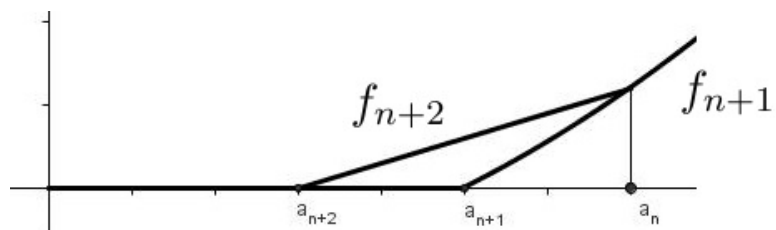


FIGURE 3 – Construction de f_n

Conclusion :

Quelles conditions sur w doivent être satisfaites pour qu'une fonction $f : \Omega \rightarrow \mathbb{R}$ mesurable vérifiant :

$$(*) \quad \iint_{\Omega \times \Omega} w \left(\frac{|f(x) - f(y)|}{|x - y|} \right) \frac{dx dy}{|x - y|^N} < +\infty$$

soit constante presque partout dans Ω ?

Nous avons commencé par le cas particulier $w : t \rightarrow t^p$ avec $p \geq 1$ qui est une fonction *convexe* et avons montré grâce à une approximation de la différentielle de f , que $\nabla f(x) = 0$. Ce qui caractérise la constance de la fonction pour presque tout x dans Ω .

Pour généraliser ce résultat, nous nous sommes restreint à l'espace

$$\mathcal{W} = \left\{ \omega \in C^0(\mathbb{R}^+, \mathbb{R}^+) \mid \omega(0) = 0 \text{ et } \omega(t) > 0 \quad \forall t > 0 \right\}$$

En supposant, en plus, que $\liminf_{t \rightarrow +\infty} \frac{w(t)}{t} > 0$, nous montrons que l'on peut se ramener à une fonction w convexe et nous concluons de la même manière que dans le cas particulier. Cette dernière hypothèse est essentielle car la fonction peut, arbitrairement et affreusement, osciller et on perd la stabilité par convolution de la condition (*). Ce qui rend difficile l'approximation.

Enfin, nous obtenons aussi une élégante formule exprimant, dans le cas particulier $w(t) = t^p$, la norme L^p du gradient :

textbf

$$\begin{aligned} \int_{\Omega} \left(\int_{\mathbb{S}^{N-1}(0,1)} w(|\nabla f(x) \cdot \sigma|) d\mu(\sigma) \right) dx \\ = \lim_{j \rightarrow +\infty} \int_{\Omega} \int_{\Omega} w \left(\frac{|f(x) - f(y)|}{|x - y|} \right) \rho_{\epsilon_j}(|x - y|) dx dy \end{aligned}$$

Nous pouvons, pour une étude ultérieure, imaginer un théorème analogue à la caractérisation des fonctions constantes, mais uniquement dans le cas f continue.

3.2 Projet M2 - Réduction de modèles

C02 - Réduction de modèles : TP

Antoine Bera

14 février 2016

1 Introduction

Dans ce document on va tenter de résoudre numériquement le problème elliptique paramétré suivant :

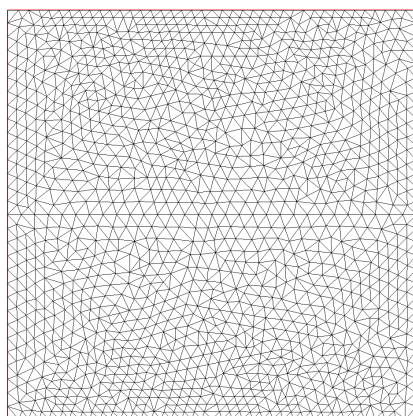
$$\begin{aligned} -\kappa(\mu)\Delta u_\mu + u_\mu &= 0 && \text{dans } \Omega \\ u_\mu &= 0 && \text{sur } \Gamma_1, \\ u_\mu &= f(\mu, \cdot) && \text{sur } \Gamma_2, \end{aligned}$$

Avec $f(\mu, y) = e^{-16(y-y_0(\mu))^2}$

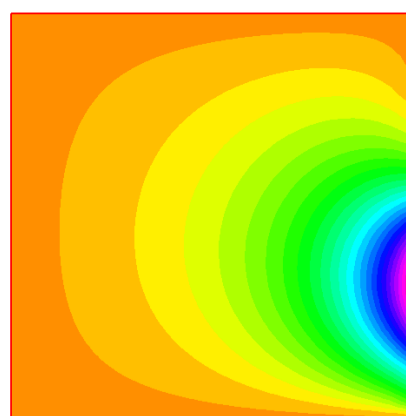
L'expérience a été menée avec le logiciel **FreeFem++**, avec lequel nous avons exécuté un premier programme résolvant le problème sous forme éléments finis.

Le code proposé permet en effet de construire le maillage et la formulation variationnelle pour $\Omega = [0, 1]^2$ et pour κ constante.

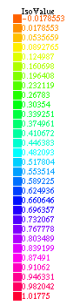
Ci-dessous nous présentont la solution obtenue ainsi que le maillage :



Maillage



Première solution



2014 - 2015

2 Plan d'expérience numérique

Maintenant, nous souhaitons trouver un maximum de solutions pour différentes valeurs de μ , en prenant κ et y_0 non constant dépendant de $\mu = (\mu_1, \mu_2) \in [0, 1]^2$:

$$\kappa = \kappa(\mu_1) = \kappa_{min} + \mu_1(\kappa_{max} - \kappa_{min})$$

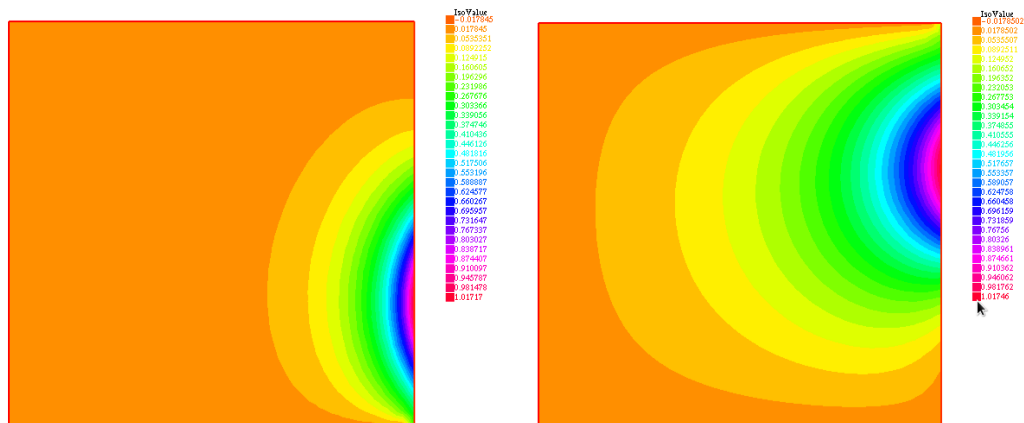
$$y_0 = y_0(\mu_2) = y_{min} + \mu_2(y_{max} - y_{min})$$

Ne pouvant pas réaliser les calculs pour un nombre infini de μ nous allons quadriller le pavé $[0, 1]^2$ en N^2 points et stocker ces valeurs dans un tableau de taille $2N^2$:

$$\text{mui} = \begin{pmatrix} \mu_{1,1} & \mu_{2,1} \\ \vdots & \vdots \\ \mu_{1,1} & \mu_{2,N} \\ \mu_{1,2} & \mu_{2,1} \\ \vdots & \vdots \\ \vdots & \vdots \\ \mu_{1,N} & \mu_{2,N} \end{pmatrix}$$

Ainsi, pour chaque ligne nous récupérons les deux valeurs μ_1 et μ_2 et nous résolvons le problème éléments finis qui en découle.

On peut alors afficher les solutions (approchées) obtenues pour différentes valeurs :



Il est aussi intéressant de stocker ces N^2 solutions lors de la programmation dans un vecteur uhi de taille N^2 .

3 Composantes principales

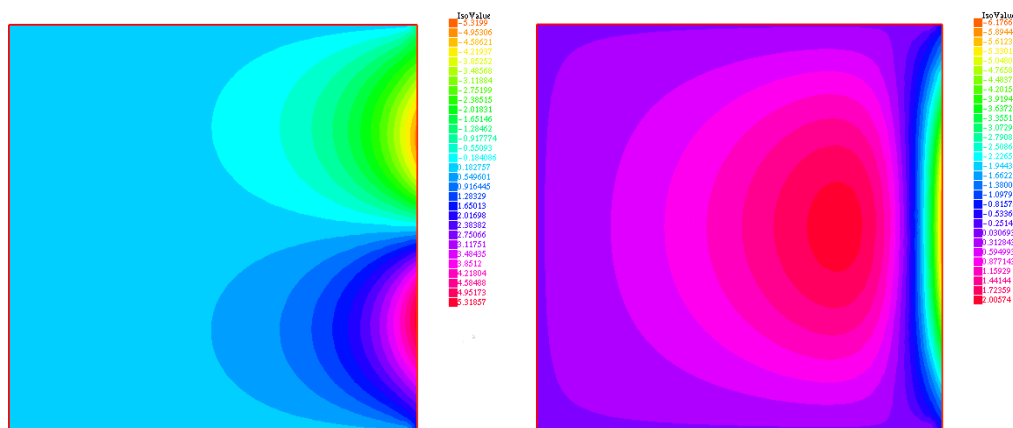
Nous allons à présent réduire la dimension grâce à la méthode ACP (Analyse en Composantes Principales). Ainsi nous calculons la matrice de corrélation déduite de la matrice des colonnes des (u_i^h) . De cette matrice on peut déduire les valeurs propres et ne considérer que les 10 plus grandes en valeur absolue. Nous avons également besoin des vecteurs propres associés. Ceci a été fourni grâce à la fonction `EigenValue`.

Nous avons donc les éléments nécessaires aux calculs des φ^k avec la formule :

$$\varphi^k = \frac{1}{\sqrt{\lambda_k}} \sum_i (v^k)_i u_i^h$$

Où les λ_k sont les valeurs propres et les v^k sont les vecteurs propres associés.

On peut tracer les fonctions propres, en voici quelques exemples :



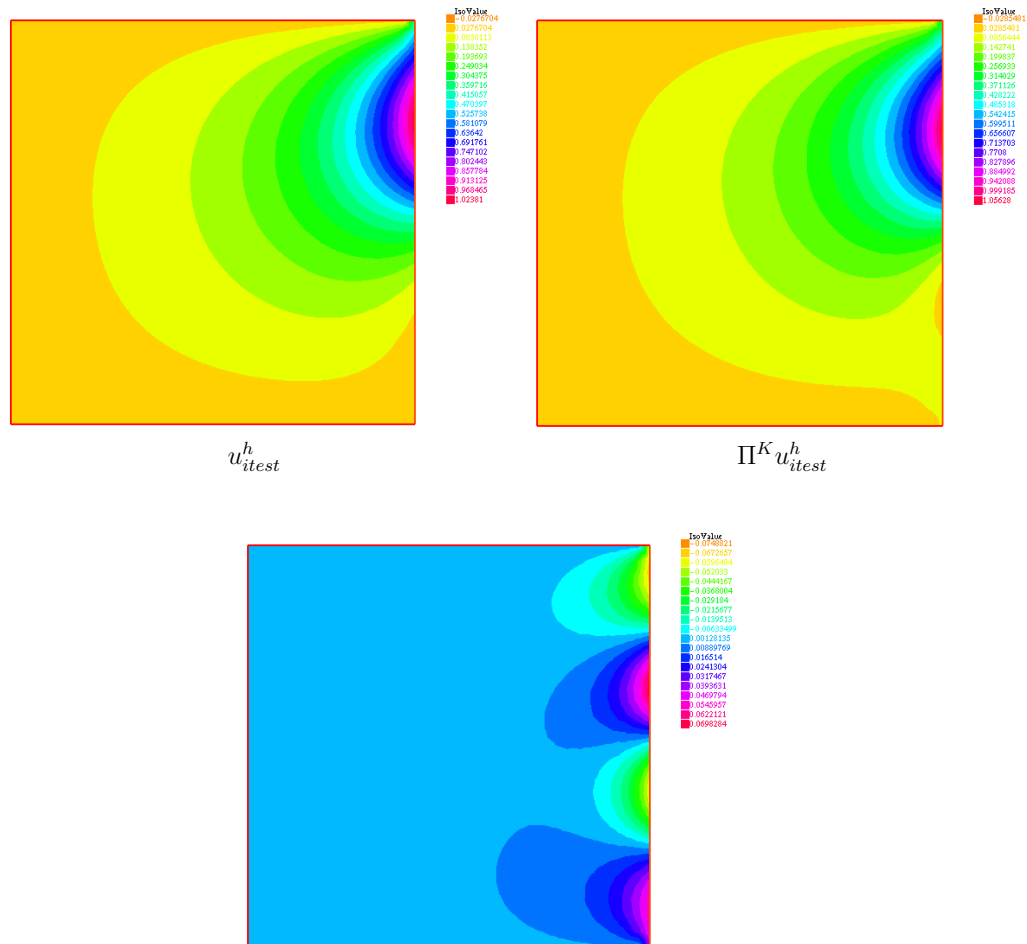
4 Vérification, calcul de $\Pi^K u$

La formule suivante nous permet d'obtenir la projection des u_i^k sur les $(\varphi^k)_{k < K}$ avec $K = 10$.

$$\Pi^K u_i^h = \sum_{k=1}^K \langle u_i^h, \varphi^k \rangle_{L^2} \varphi^k$$

On peut alors comparer la valeur exacte avec sa projection et tracer l'erreur de projection :

2014 - 2015



5 Données moins régulières

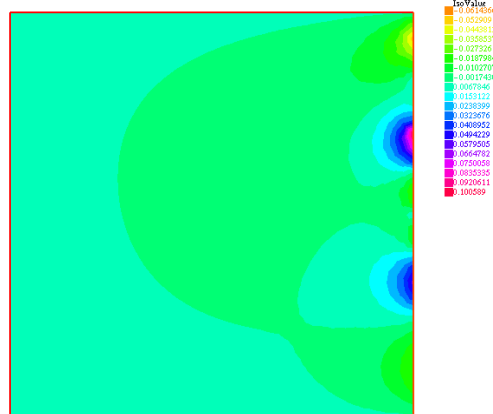
On va maintenant changer notre fonction f . On remarque que l'erreur a augmenté avec cette fonction moins régulière, même si elle est toujours située au même endroit dans Ω . $f(\mu, y) = \max(0, 1 - 4|y - y_0(\lambda)|)$.

Ci dessous l'erreur obtenue avec la nouvelle fonction f :

6 Maillage et interpolation dans l'espace paramétrique

A présent, on souhaite que les valeurs de μ choisies découlent des sommets d'un maillage éléments finis. Le maillage ainsi effectué comporte N_s points, ainsi notre matric mui contiendra $2N_s$ coefficients.

Page 4



7 Modèle réduit complete (espace+paramètres)

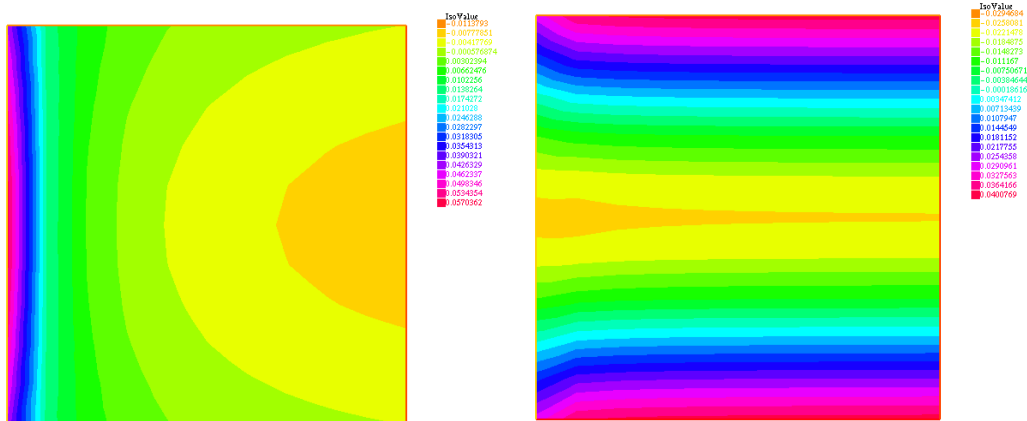
Maintenant, l'objectif est de déterminer les fonctions $a^k(\mu)$ tels que les solutions du problème (projetées dans $vect_k(\varphi^k)$) s'écrivent :

$$\tilde{u}^h(x, \mu) = \sum_{k=1}^K a^k(\mu) \varphi^k(x)$$

On va alors utiliser les u_i^h obtenus précédemment afin de déterminer leur projection sur $vect_k(\varphi^k)$: $\Pi^K u_i^h(x) = \sum_{k=1}^K a_i^k \varphi^k(x)$.

On cherche donc les coefficients $a_i^k = \langle u_i^h, \varphi^k \rangle_{L^2}$, calculés grâce à la fonction `int2d`.

Les vecteurs $(a_i^k)_i$ représentent la fonction $a^k(\mu)$, que l'on peut représenter :



Un premier k

Un deuxième k

2014 - 2015

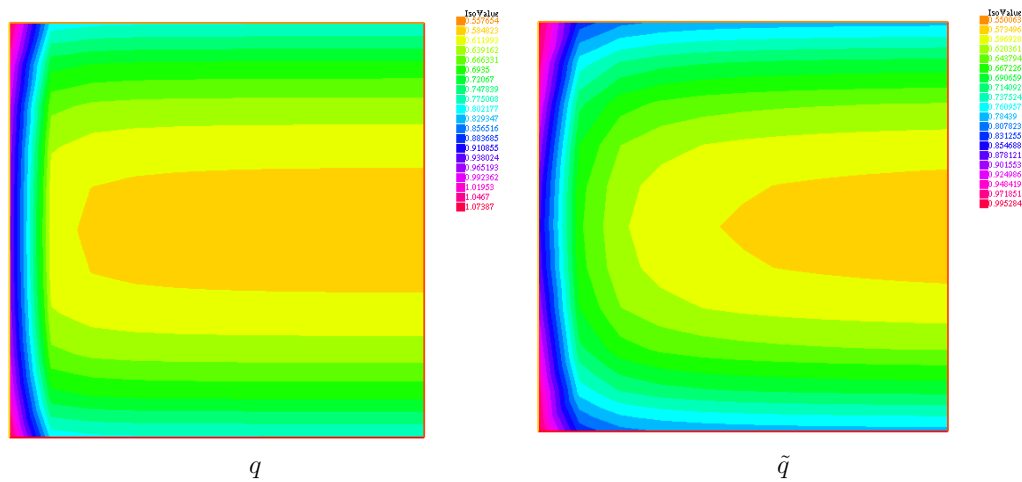
On peut remarquer que lorsque μ_1 est proche de 0, $\kappa(\mu)$ l'est également. Ceci permet d'expliquer les singularités obtenues lors de l'affichage des solutions, puisque l'on se rapproche d'un pôle de l'équation ($u = 0$).

8 Quantité d'intérêt

Nous allons maintenant considérer la quantité d'intérêt associée au problème :

$$q(\mu) = \frac{1}{2} \int_{\Omega} |\nabla u_{\mu}|^2 dx$$

Toujours avec `int2d`, on calcule cette quantité pour tous les u_i^h afin de se représenter q . On peut aussi calculer une approximation de q en utilisant la projection des u_i^h et la formule $\tilde{q}(\mu) = \frac{1}{2} \sum_{k=1}^K \sum_{l=1}^K a^k(\mu) a^l(\mu) \int_{\Omega} \nabla \varphi^k \cdot \nabla \varphi^l$



On remarque toujours que pour μ_1 proche de 0, il y a des singularités dans le calcul de solutions approchées.

Page 6

2014 - 2015

9 Conclusion

L'objectif attendu, de résoudre numériquement l'équation sur le domaine Ω , a été atteint. Pour cela il a été nécessaire de la résoudre par éléments finis en un nombre fini de points et d'en déduire des "composantes principales". Ces dernières nous ont permises, par interpolation, d'obtenir une expression approchée de la solution.

On ne parle pas ici d'erreur d'approximation, mais cette dernière peut être intéressante à calculer afin de vérifier (au moins numériquement) que le travail effectué rend bien compte de la réalité des solutions. Cette erreur est mathématiquement exprimable en fonction de différents paramètres calculés lors des différentes approximations.

Page 7

3.3 Projet M2 - Résolution des problèmes de diffraction par équations intégrales

TP TA02 : Résolution des problèmes de diffraction par équations intégrales

BERA Antoine et TEXIER Aurore

10 novembre 2015



Introduction

Le but de la partie I est d'implémenter une méthode de calcul rapide des valeurs singulières d'une matrice, puis de développer une méthode pour réduire les coûts de stockage d'une matrice pleine en la découpant par blocs et en utilisant sa décomposition en valeurs singulières, afin de calculer un produit matrice-vecteur rapide.

Dans la partie II, on essaye de coder une méthode pour réduire le temps de calcul du produit matrice pleine-vecteur et réduire le coût de stockage. Pour ce faire, on utilisera un développement en ondes planes de la fonction de Green et un découpage en boîte de l'espace de travail.

Table des matières

1	Partie I : Approximations par des matrices de rang faible	3
1.1	Question 1 : Génération du nuage de points	3
1.2	Question 2 : Etude des valeurs singulières de la matrice de Green	3
1.3	Question 3 : ACA avec pivotage complet	4
1.4	Question 4 : ACA avec pivotage partiel	5
1.5	Question 5 : Partitionnement de la matrice en 4 sous-blocs	7
1.6	Question 6 : Partitionnement récursif de la matrice	8
1.7	Question 7 : Algorithme rapide basé sur ce partitionnement	10
2	Partie II : Méthode multipôle rapide pour un nuage de points	11
2.1	Question 1 : Nuage de points	11
2.2	Question 2 : Produit matrice-vecteur classique	11
2.3	Question 3 : Quadrature de Gauss-Legendre	12
2.4	Question 4 : Intégration des harmoniques sphériques	12
2.5	Question 5 : Vérification du développement en ondes planes	13
2.6	Question 6 : Méthode multipôle rapide mono-niveau	14
2.7	Question 7 : Complexité de la FMM	15
2.8	Question 8 : Comparaison méthode classique/méthode FMM	15
3	Conclusion	16

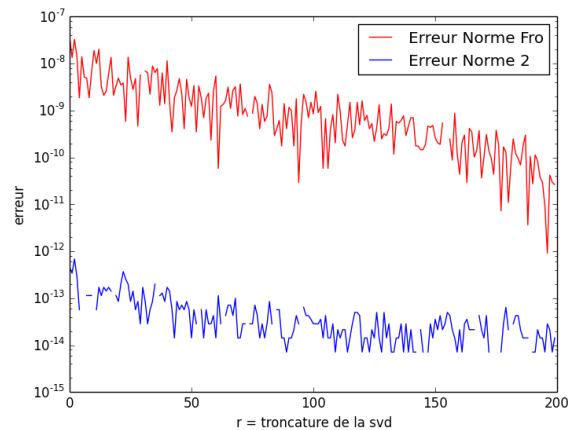
1 Partie I : Approximations par des matrices de rang faible

1.1 Question 1 : Génération du nuage de points

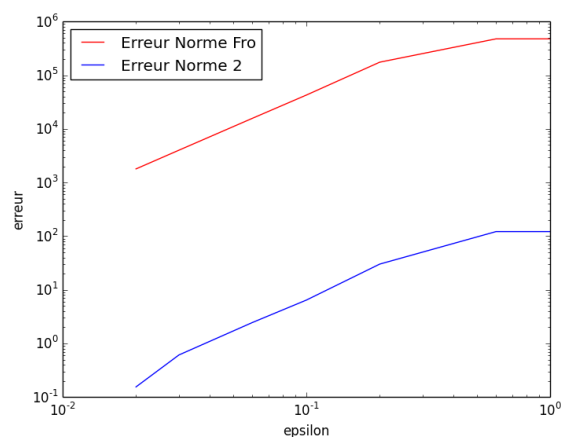
On a généré le nuage de points X défini sur $[-1; 1]$ par : $x_i = -1 + i * h$ où $h = 2/N$ est le pas d'espace et $i \in [0; N]$.

1.2 Question 2 : Etude des valeurs singulières de la matrice de Green

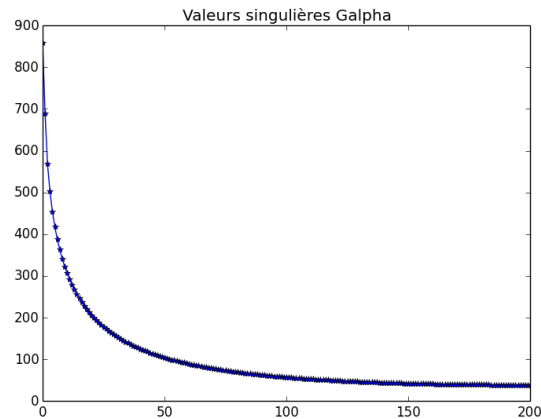
A l'aide d'un `meshgrid`, on définit la matrice $(G_\alpha)_{i,j} = \frac{\exp(ik|x_i - x_j|)}{|x_i - x_j| + \alpha}$, $\forall (x_i, x_j) \in X^2$. Ensuite, on calcule G_r la SVD de G_α tronquée aux r plus grandes valeurs singulières en utilisant le paramètre de troncature ϵ_{SVD} tel que $\|G_\alpha - G_r\|_2 < \epsilon_{SVD} \|G_\alpha\|_2$. Pour choisir un ϵ_{SVD} convenable, on vérifie si $\|G_\alpha - G_r\|_F = \sum_{i=r+1}^{N+1} \sigma_i^2$ et $\|G_\alpha - G_r\|_2 = \sigma_{k+1}$.



On voit sur la figure précédente que plus on rajoute de termes dans la SVD tronquée, plus $\|G_\alpha - G_r\|_F = \sum_{i=r+1}^{N+1} \sigma_i^2$ et $\|G_\alpha - G_r\|_2 = \sigma_{k+1}$ se rapprochent de 0.



En faisant varier ϵ_{SVD} , on voit alors que pour $\epsilon_{SVD} < 10^{-2}$, les égalités précédentes sont bien vérifiées. Dans ce cas, le rang de G_r , la SVD tronquée, est égale au rang de G_α .



D'après ce qui précède et comme les valeurs singulières de G_α ne sont pas très petites à partir d'un certain rang, on ne peut pas approcher G_α par une matrice de rang faible. De plus, on ne va pas pouvoir utiliser directement la SVD pour calculer le produit matrice-vecteur rapide car le nombre d'opérations serait en $O(N^2)$.

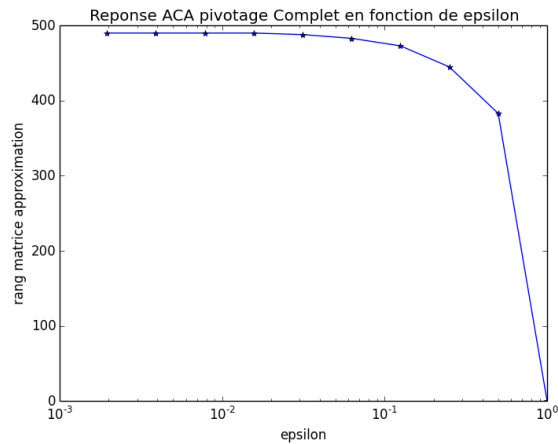
1.3 Question 3 : ACA avec pivotage complet

Une décomposition ACA est une méthode itérative qui a pour but de déterminer le rang d'une matrice A . Pour ce faire on trouve le plus grand élément de la matrice, on "annule" la ligne et la colonne correspondantes et on ajoute cette "annulation" dans une matrice initialement nulle B_k . Cette matrice approche donc la matrice initiale au fur et à mesure que l'algorithme avance.

Le critère d'arrêt qu'on utilise est : $\frac{\|A-B_k\|_F}{\|A\|_F} < \epsilon_f$ où ϵ_f est donné par l'utilisateur.

Pour des ϵ_f assez petits (on verra après ce que signifie "assez petit"), on peut vérifier que la décomposition d'une matrice avec l'ACA complet est obtenue en un nombre d'itérations égal au rang de cette matrice.

Maintenant, traçons le nombre d'itérations en fonction de ϵ_f pour une matrice de taille $500 * 500$ de rang 490 :



On voit que pour $\epsilon_f < 10^{-2}$, l'approximation obtenue est correcte.

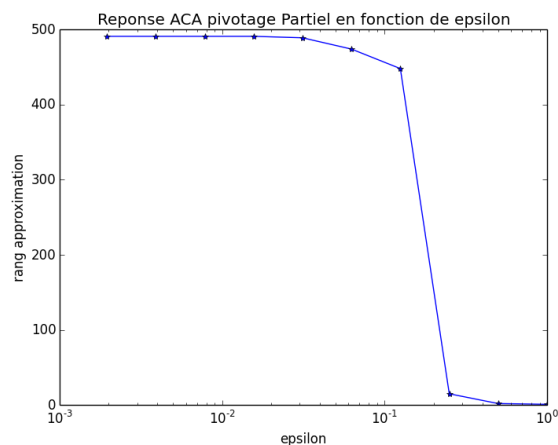
1.4 Question 4 : ACA avec pivotage partiel

L'algorithme ACA avec pivotage partiel repose sur la même idée que le ACA pivotage complet. Néanmoins on cherche à optimiser le nombre d'opérations en cherchant le pivot maximal (en valeur absolue) dans la ligne sur laquelle on se trouve, en démarrant arbitrairement sur la première. Ensuite on va réaliser les mêmes opérations que le précédent algorithme pour annuler la ligne et la colonne correspondantes. Enfin on ne recherche plus le pivot dans les lignes et les colonnes déjà parcourues.

La décomposition d'une matrice avec l'ACA partiel est obtenue en un nombre d'itérations égale au rang de cette matrice.

On utilise le critère d'arrêt : $\frac{\|u_k\|_2 \|v_k\|_2}{\|B_k\|_F} < \epsilon_p$.

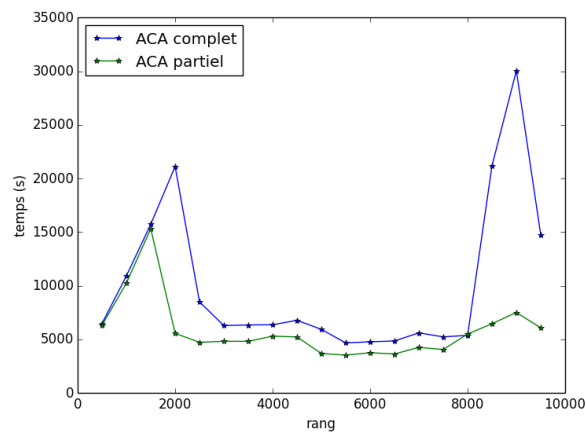
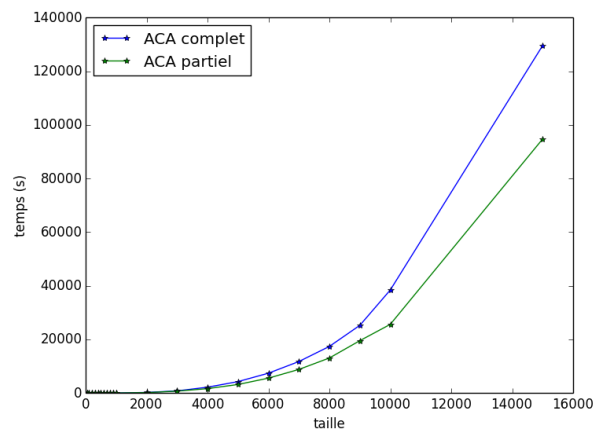
Traçons le nombre d'itérations en fonction de ϵ_p :



On remarque que pour $\epsilon_p < 10^{-2}$, l'approximation obtenue est correcte.

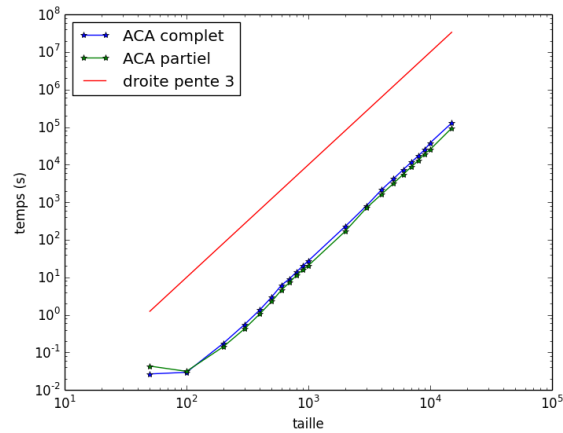
Dans l'algorithme ACA avec pivotage complet, l'ordinateur recherche à chaque itération le maximum parmi tous les éléments de la matrice. Cependant, on peut réduire cette recherche uniquement sur la ligne à laquelle on se trouve pour réduire le temps de calcul ; C'est ainsi qu'est programmé l'ACA avec pivotage partiel.

En effet si on note (m, n) la taille de la matrice et r son rang, on aura une complexité en $O(rmn)$ pour ACA complet et $O(r^2(m+n))$ pour ACA partiel. Ceci donne un fort avantage pour l'algorithme ACA partiel lorsque la taille est assez grande. On peut retrouver ces résultats en traçant le temps de calcul en fonction de la taille de la matrice, puis en fonction du rang :



On voit sur la première figure que l'ACA partiel est plus rapide que l'ACA complet surtout pour des matrices de grande taille.

On peut également tracer la courbe de variation de la taille en \log/\log :

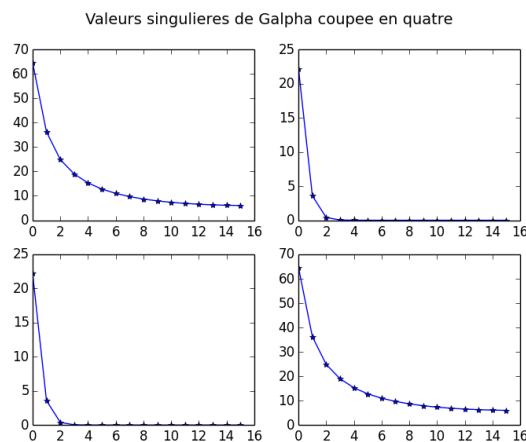


Dans la figure tracée en *log/log* on remarque que les courbes sont presque des droites parallèles. Nous pouvons alors interpréter les coefficients de leurs régression linéaire. Nous avons pris un rang égal à la taille de la matrice, on s'attend donc à trouver un temps de calcul en $O(n^3)$, ce qui est le cas puisque les pentes sont respectivement de 2.96 pour ACA Complet et 2.88 pour ACA partiel. (Ceci correspond bien à une complexité en $O(n^3)$).

1.5 Question 5 : Partitionnement de la matrice en 4 sous-blocs

Comme supposé dans le titre de cette partie, nous allons couper notre matrice en 4 matrices de même taille (ou presque de même taille si la matrice ne le permet pas). Pour des questions de simplicité, nous supposons dans la suite que les matrices sont de taille 2^k .

Pour chaque sous-bloc, nous allons regarder les valeurs singulières et les tracer en ordre décroissant :



On observe que les blocs extra-diagonaux n'ont qu'un nombre faible de valeurs singulières non-négligeables.

On peut effectuer une approximation de rang faible lorsque le nombre de valeurs singulières non-négligeables est assez petit devant la taille du sous-bloc. La valeur “assez petite” doit être choisie judicieusement en fonction de la taille, ce point sera développé par la suite.

Enfin pour obtenir une approximation précise le paramètre ϵ_{SVD} doit être choisit avec les remarques de la question 2, c'est à dire $\epsilon_{SVD} < 10^{-2}$.

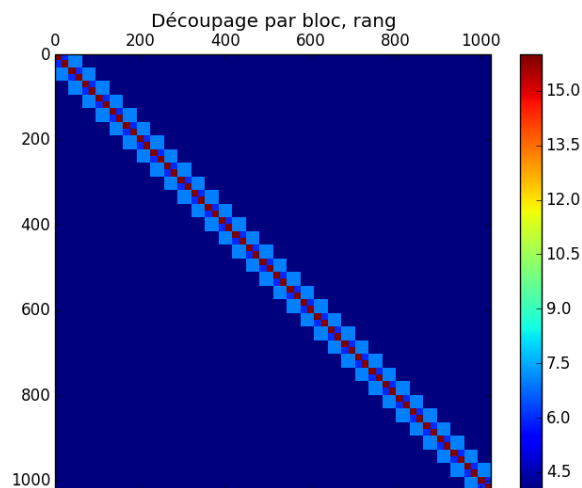
1.6 Question 6 : Partitionnement récursif de la matrice

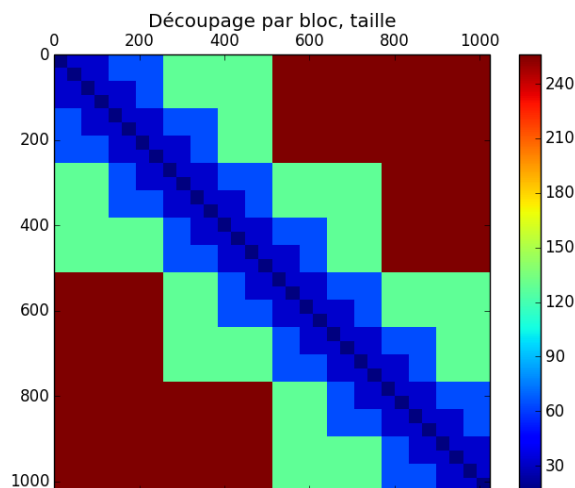
Le but est maintenant de continuer le partitionnement et donner une condition de non-subdivision d'un bloc permettant d'optimiser les coûts de stockage. En effet, si on décompose en tout petits morceaux, l'approximation de rang faible ne permet pas de gagner beaucoup en termes de coût de stockage ; et si on ne décompose pas un gros bloc qui n'est pas de rang faible, le stockage reste le même.

Ainsi on utilise dans la fonction `partitionnement` des paramètres `rang_faible` et `niveau` qui correspondent, respectivement, au rang limite permettant de considérer une matrice comme étant de rang faible et le nombre maximum d'itérations exécutées (le niveau de descente maximal).

Pour simplifier les choses, on a choisi de travailler avec des matrices de taille 2^n . L'algorithme est un processus récursif : on partitionne chaque matrice en 4 blocs et si on est pas de rang faible (le rang est calculé avec l'ACA partiel), on réapplique ce processus en diminuant de 1 le paramètre `niveau`.

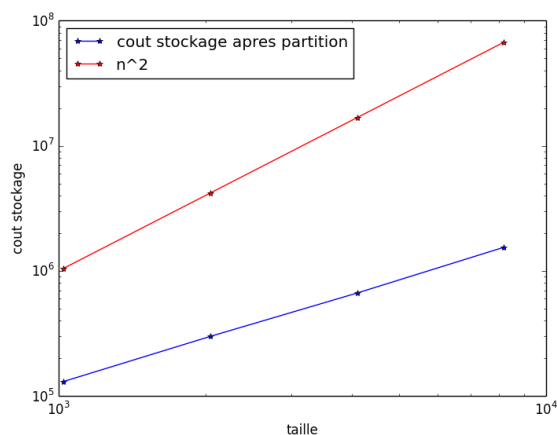
On peut alors afficher la matrice G_α de taille 2^{10} après lui avoir appliqué la fonction `partitionnement` avec les paramètres `rang_faible = 7` et `niveau = 6` : tout d'abord on peut représenter, avec un code couleur, la valeur du rang ; puis on représente la taille de chaque sous-bloc.





On remarque sur ces figures que quasiment la totalité des blocs se sont arrêtés car ils sont de rang faible hormis les blocs diagonaux qui se sont arrêtés à cause du niveau de descente. Sur la deuxième figure, on peut observer que plus on est loin de la diagonale, plus les blocs sont grands. Ceci peut être expliqué par la décomposition en valeurs singulières de G : comme vu à la question 5, les blocs extra-diagonaux sont de rang faible contrairement aux blocs diagonaux.

Enfin, on peut regarder le gain en terme de coût de stockage. Sur la figure suivante, on remarque que le coût de stockage après partitionnement est bien inférieur au coût habituel de stockage. En effet, lorsqu'un bloc est de rang faible, son coût de stockage est réduit à $(2n + 1)r$ (où n est la taille du bloc et r son rang) en comparaison avec n^2 (coût habituel d'un bloc de taille n).



On a réussi à diminuer drastiquement le coût de stockage de notre matrice. Ainsi, on peut utiliser des matrices de tailles plus grande qu'avant. Dans la question suivante, on s'intéresse à l'utilisation du partitionnement dans le cadre d'un produit matrice-vecteur.

1.7 Question 7 : Algorithme rapide basé sur ce partitionnement

Pour calculer le produit matrice pleine-vecteur, on partitionne la matrice comme expliqué à la question précédente, en utilisant l'ACA partiel pour calculer le rang de chaque bloc. Ensuite pour chaque bloc, s'il est de rang faible, on utilise sa décomposition en valeurs singulières tronquée pour calculer le produit, on a alors $(2n + 1)r$ opérations où n est la taille du bloc et r son rang ; sinon si le bloc n'est pas de rang faible alors on fait le produit classiquement et on a n^2 opérations avec n la taille du bloc.

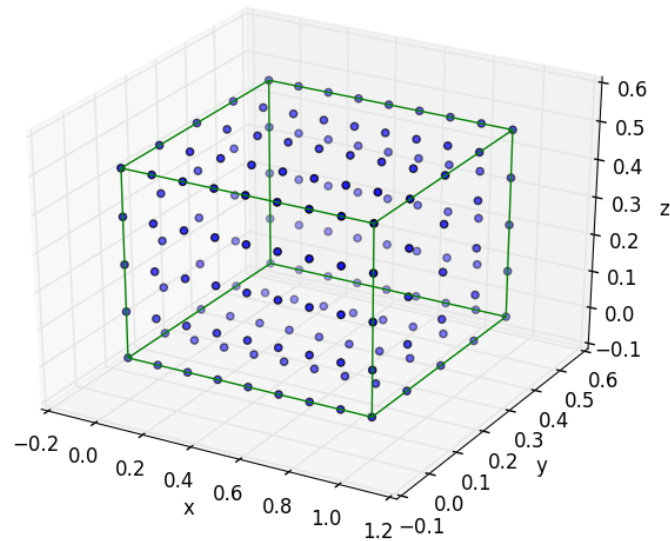
Pour G_α de taille 2^{10} , l'erreur entre le calcul classique et le calcul utilisant le partitionnement est de $2.4 * 10^{-4}$.

Donc cette méthode permet de calculer une approximation d'un produit matrice-vecteur plus rapidement que le produit classique mais en étant tout de même assez précis.

2 Partie II : Méthode multipôle rapide pour un nuage de points

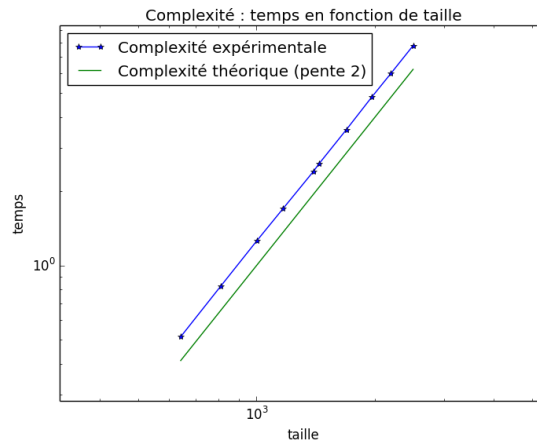
2.1 Question 1 : Nuage de points

On a généré le nuage de points X définissant sur le bord du pavé $[0; L] * [0; l] * [0; l]$. Pour cela, on utilise la fonction `meshgrid` pour chaque face du pavé, puis on rassemble tous les points mais en faisant attention de ne pas en prendre en double.



2.2 Question 2 : Produit matrice-vecteur classique

Après construction de la matrice de Green G , nous avons effectué le produit matrice-vecteur classique entre G et un vecteur aléatoire ρ . On remarque que la complexité en temps de calcul est en n^2 . En effet on a tracé le temps de calcul en fonction de la taille de la matrice en *loglog* qui permet de constater que la droite est de pente environ 2. Nous avons tracé une droite de pente exactement 2 pour comparer et nous avons fait une régression linéaire qui nous donne une droite de pente 1.99.



Le plus grand N traitable par nos ordinateurs personnels est 10000.

2.3 Question 3 : Quadrature de Gauss-Legendre

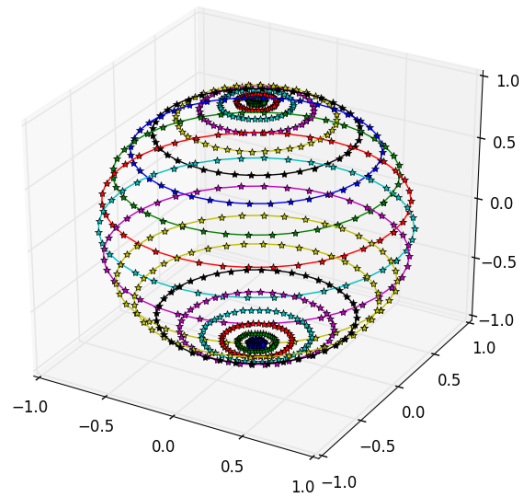
Pour calculer les poids et les points de la quadrature de Gauss Legendre à L points, on calcule les valeurs propres et vecteurs propres de la matrice T ayant pour seuls coefficients non nuls les $T_{i,i+1} = T_{i+1,i} = \frac{i}{\sqrt{4i^2-1}}$. Alors les points sont les valeurs propres de T et les poids sont le double du carré de la première coordonnée des vecteurs propres.

On a pu constater que pour les 4 premières quadratures, les points et les poids correspondent bien aux valeurs théoriques.

2.4 Question 4 : Intégration des harmoniques sphériques

Pour intégrer ultérieurement les harmoniques sphériques il nous faut une quadrature de la sphère unité. Pour ce faire il faut utiliser les coordonnées sphériques (φ, θ) . On utilise alors une quadrature uniforme sur $[0; 2\pi]$ à $2L + 1$ points pour φ et une quadrature de Gauss-Legendre à $L + 1$ points pour θ . Cette quadrature est correcte puisque une quadrature uniforme sur $[0; 2\pi]$ donne un poids total de 2π et une quadrature de Gauss-Legendre donne un poids total de 2 ; ainsi on a un poids total de 4π qui correspond à la surface de la sphère unité.

Pour chaque point du maillage le poids associé est le produit des deux poids associés aux quadratures de φ et θ . On représente sur la figure ci-après le maillage de la sphère unité.



On a vérifié que cette quadrature est exacte pour les harmoniques sphériques.

2.5 Question 5 : Vérification du développement en ondes planes

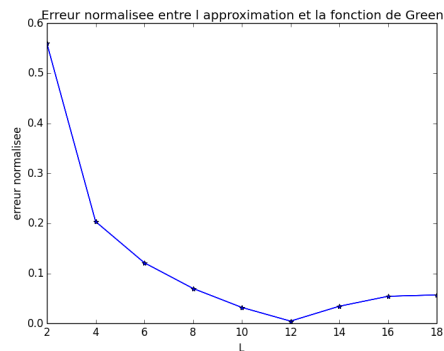
Pour coder le développement en ondes planes de la fonction de Green, on définit les polynômes de Legendre P_p et la fonction de Hankel sphérique du premier ordre $h_p^{(1)}$. Ensuite l'approximation est donnée par :

$$\frac{\exp(i k |x - y|)}{4\pi |x - y|} = \lim_{L \rightarrow +\infty} \int_{s \in S^2} e^{i k s \cdot r} \mathcal{G}_L(s, r_0)$$

où

$$\mathcal{G}_L(s, r_0) = \frac{i k}{16\pi^2} \sum_{p=0}^L (2p+1) i^p h_p^{(1)}(k |r_0|) P_p\left(\frac{s \cdot r_0}{\|r_0\|}\right)$$

et $x - y = r_0 + r$. Comme on ne peut pas prendre une limite numériquement, il faut chercher pour quel L l'intégrale est la plus proche de la fonction de Green.



On voit ici que $L = 12$ est l'approximation la plus précise. En effet, si on utilise la formule vue en cours pour déterminer L :

$$L = \sqrt{3}kd + 7.5\log(\sqrt{3}kd + \pi)$$

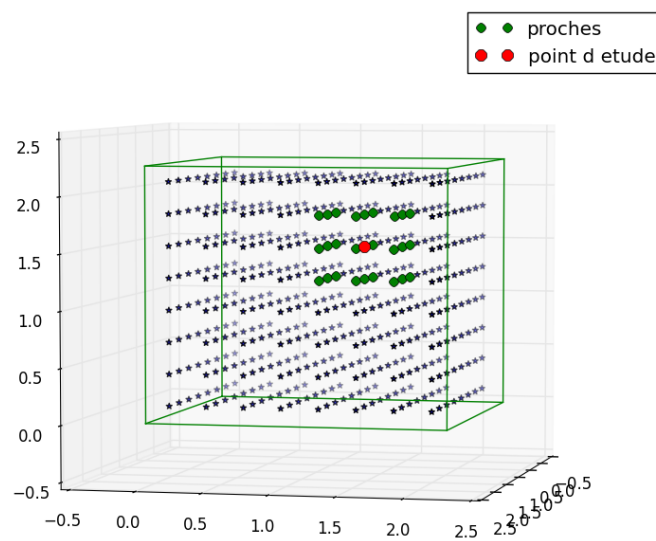
avec $kd = 0.5 * 2\pi$ comme choisi pour la figure (car on doit avoir $d \geq 0.3\lambda = 0.3 * 2\pi/k$), on trouve bien $L = 12$.

2.6 Question 6 : Méthode multipôle rapide mono-niveau

Pour découper en boîte, on part initialement d'un cube de taille d , puis on le découpe en 8 cubes de taille $d/2$. On répète ce procédé jusqu'à la précision voulue. Pour cette fonction, on a choisi de retenir l'emplacement du centre de chaque cube et de leur taille.

A partir du calcul de ces centres, on définit une fonction qui détermine si deux boîtes sont proches ou lointaines. On dit que deux boîtes sont proches lorsqu'elles se touchent (même par un seul sommet).

On obtient par exemple la figure qui suit, représentant les centres des boîtes. Le point rouge est le centre de la boîte étudiée, les points verts sont les centres des boîtes proches et les autres sont les centres des boîtes lointaines.



Ce découpage en boîte va être utile pour calculer le produit matrice-vecteur. En effet, on va séparer les calculs du produit en fonction de la distance entre les points du domaine. Ainsi, pour deux points éloignés, on approchera leur interaction par l'interaction des centres des boîtes les contenant. On considère que deux points sont éloignés lorsqu'ils sont dans des boîtes éloignées. Pour deux points proches, on utilisera la méthode définie en I pour calculer le produit.

Détaillons le calcul pour des points éloignés : d'abord, pour chaque boîte, on calcule et on stocke le moment multipôle ; ensuite on le transfère à toutes les boîtes lointaines ; enfin lorsqu'une boîte a récupéré l'information de toutes ses boîtes lointaines, elle peut propager les contributions correspondantes à l'ensemble de ses points.

On a réussi à coder la partie pour les points éloignés, en revanche, on n'a pas pu ajouter la contribution des proches car comme précisé dans la partie I, on a codé la BEM pour des matrices carrées de taille 2^n , mais pour le calcul des proches, la matrice de Green correspondante n'est pas forcément carrée.

2.7 Question 7 : Complexité de la FMM

Lors de l'estimation de la complexité, l'étape de calcul des moments multipôles ne doit pas être pris en compte. En effet, c'est une initialisation du problème contrairement aux transferts, aux calculs (pour les lointains et les proches) et à la redistribution de l'information.

N'ayant pas pu coder entièrement la FMM, on ne peut pas tracer la courbe de complexité.

2.8 Question 8 : Comparaison méthode classique/méthode FMM

En théorie, la FMM est plus rapide que le calcul classique : la FMM a une complexité en $O(N^{3/2})$ comparée à celle du produit classique en $O(N^2)$.

3 Conclusion

En conclusion, ces deux études nous ont permis de réduire la complexité (et donc le temps de calcul) du produit matrice-vecteur et le coût de stockage d'une matrice pleine (comme la matrice de Green).

Nous avons pour cela utilisé des décompositions en valeurs singulières afin de trouver le rang d'une matrice et déterminer si elle est de rang faible. Nous avons utilisé ceci au travers d'un algorithme récursif qui divise la matrice en blocs ; si un bloc est de rang faible le coût de stockage et le temps de calcul matrice-vecteur sont diminués.

Dans un deuxième temps, nous avons réalisé que lors du produit matrice-vecteur, deux points éloignés auront la même contribution que ces mêmes points légèrement déplacés. Ainsi on a regroupé les points en boîte et donc diminué le temps de calcul en l'effectuant par boîte et non par point lorsque les boîtes sont éloignées.

Ces deux méthodes sont théoriquement et numériquement avantageuses lorsqu'il s'agit de matrice à grande échelle (le travail effectué hors complexité est en effet trop important pour améliorer le calcul pour des petites matrices).

3.4 Projet M2 - Création de la classe des matrices creuses

I04 - Matrices creuses

Bera-Paolantoni

30 janvier 2016

1 Introduction

Quand on veut utiliser des matrices de grande taille, il peut être coûteux en mémoire d'en stocker tous les éléments. Heureusement, dans certains problèmes, les matrices utilisées comportent un grand nombre de zéros : ce sont des matrices creuses. Il est donc pratique de créer une nouvelle classe pour stocker les matrices : `matsparse`, afin d'utiliser librement des matrices sans en stocker les éléments inutiles (les éléments nuls). Pour cela, il va falloir être capable de construire des `matsparse` à partir de matrices classiques, et d'effectuer toutes les opérations de base directement sur cette nouvelle classe. Ensuite, il faudra trouver un moyen de les afficher de manière classique. Enfin, à partir de ces fonctionnalités, on pourra utiliser cette classe dans un vrai problème de calcul scientifique : ici, il s'agira de la résolution d'un système linéaire avec la méthode de gradient conjugué.

2 La classe `matsparse`

Tout d'abord, on s'intéresse à la création de la classe `matsparse`, c'est-à-dire à la manière de stocker les matrices creuses, à la façon de les construire, et aux opérations élémentaires que l'on peut faire dessus. Pour cela, on utilise trois fichiers :

1. `matsparse.h` pour la définition de la classe, où l'on inclut un opérateur d'extraction d'un coefficient,
2. `matsparse.cpp` pour les constructeurs, l'extraction des différentes informations de la matrice et un opérateur de copie,
3. `fonctions_pour_matsparse.cpp` où l'on construit les opérations élémentaires sur les matrices.

2.1 La classe

Stockage des matrices creuses

Pour stocker les matrices creuses, nous choisissons de créer trois vecteurs privés : A , I et J . Dans I sont stockés les *indices des lignes* des coefficients non nuls, dans J les *indices des colonnes*, et dans A les *valeurs* des coefficients non nuls. De plus, le premier élément de I représente le *nombre de lignes* de la matrice, le premier élément de J le *nombre de colonnes*, et le premier élément de A sera arbitrairement fixé à 0, afin de ne pas décaler les indices.

Ainsi, si la matrice creuse possède p coefficients non nuls, Les vecteurs A , I et J seront de taille $p + 1$. Et avec ces trois vecteurs, on a toutes les informations concernant la taille de la matrice et les coefficients non nuls de la matrice.

La donnée d'un triplet $(I(p), J(p), A(p))$ pour $p \geq 1$ donne donc la valeur d'un coefficient non nul de la matrice ainsi que sa position dans la matrice. Le couplet $(I(0), J(0))$ donne lui les dimensions de la matrice au sens classique : si $(I(0), J(0)) = (m, n)$, la matrice correspondante sera dans $M_{m,n}(\mathbb{R})$.

Dans toute la suite, pour construire une matrice creuse, on considèrera qu'un de ses coefficients est nul si sa valeur est inférieure à 10^{-15} . De plus, en ce qui concerne les notations, quand on parlera de la `mat` `sparse` M , on notera I_M , J_M et A_M les trois vecteurs qui la représentent.

Accès aux informations sur la matrice

Ces trois vecteurs étant stockés en privé, on ne peut pas modifier une matrice déjà créée, mais il est utile d'avoir accès aux données stockées dedans. C'est pourquoi nous avons créé dans un fichier `mat` `sparse` `.cpp` des fonctions `lignes`, `colonnes` et `valeurs`, qui récupèrent les informations contenues respectivement dans I , J et A .

Nous avons également créé une fonction `[]`, directement dans la classe, qui permet de récupérer la valeur d'un élément de la matrice creuse à partir de ses indices classiques (indices apparaissant dans le stockage classique).

Cette fonction récupère la i -ième ligne `Li=M[i]` de la matrice M , on peut donc extraire l'élément d'indice (i, j) de M en appelant `M[i][j]`.

Pour extraire la ligne i , on crée d'abord un vecteur rempli de 0 que l'on va ensuite remplir avec les éléments de la ligne voulue : `chat`. Pour cela, on utilise le nombre de colonnes de la matrices, stocké dans `J[0]`, afin d'allouer le vecteur `chat`. Ensuite, on va

parcourir le vecteur I en sélectionnant les éléments valant i . Quand on se trouve sur un tel élément : si $I[p]=i$ on récupère alors les valeurs de $J[p]$ et de $A[p]$, pour fixer alors $chat[J[p]]=A[p]$.

Constructeurs

Pour construire des `matparse`, on fait trois constructeurs :

1. un premier qui crée la matrice nulle d'une taille donnée de type `matparse`,
2. un deuxième qui crée une matrice de type `matparse` à partir d'une matrice classique stockée sous forme d'un vecteur de vecteur,
3. un troisième qui crée une matrice de type `matparse` à partir de la donnée de trois vecteurs qui représentent les mêmes données que les vecteurs A , I et J dans la classe.

Pour le premier, on crée une `matparse` en créant son vecteur I qui ne contient qu'un élément : le nombre de lignes voulu, son vecteur J qui ne contient qu'un élément : le nombre de colonnes voulu, et son vecteur A qui ne contient qu'un élément : le 0 fixé arbitrairement.

Pour le deuxième : on fixe comme dans le premier le premier élément de I , J et A correspondant à la taille de la matrice. Puis on fait une double boucle pour parcourir tous les éléments de la matrice classique, et à un élément donné, on fait un test : si sa valeur est supérieure à 10^{-15} , on ajoute au vecteur I l'indice de ligne de l'élément, au vecteur J son indice de colonne, et au vecteur A sa valeur.

Enfin, le dernier constructeur est presque immédiat puisqu'il n'y a qu'à récupérer les trois vecteurs d'entrée. On suppose que les données en argument sont *bien construites*. En effet, on ne vérifie pas qu'il n'y ait pas deux couples d'indices donnant des valeurs distinctes. On ne vérifie pas non plus que le nombre de lignes placé en $I[0]$ est compatibles avec les indices stockés dans $I[i]$ pour $i \neq 0$.

2.2 Opérations sur la classe

2014 - 2015

```

Constructeur matrice nulle
matrice :
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

dans matsparse :
0
5
5

Constructeur a partir d'une classique
matrice :
0 0 12 75
0 0 12 75

dans matsparse :
0 12 75 12 75
2 0 0 1 1
4 2 3 2 3

Constructeur avec A, I et J
Matrice Laplacien
2 -1 0 0 0 0 0 0 0 0
-1 2 -1 0 0 0 0 0 0 0
0 -1 2 -1 0 0 0 0 0 0
0 0 -1 2 -1 0 0 0 0 0
0 0 0 -1 2 -1 0 0 0 0
0 0 0 0 -1 2 -1 0 0 0
0 0 0 0 0 -1 2 -1 0 0
0 0 0 0 0 0 -1 2 -1 0
0 0 0 0 0 0 0 -1 2 -1
0 0 0 0 0 0 0 0 -1 2

dans matsparse :
0 2 2 2 2 2 2 2 2 2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
10 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8
10 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 9

```

FIGURE 1 – Test des constructeurs

2.2 Opérations sur la classe

Toutes les opérations suivantes sont dans le fichier `operations_sur_matsparse.cpp`.

Transposition

Cette opération prend en entrée une `matsparse` M et retourne une `matsparse` N .

Pour cette opération, on récupère les vecteur I_M , J_M et A_M qui représentent M en `matsparse`. Et on construit les trois vecteurs qui vont représenter N en `matsparse` : $I_N = J_M$, $J_N = I_M$ et $A_N = A_M$.

2.2 Opérations sur la classe

2014 - 2015

Somme et différence

L'opération de différence étant similaire à l'opération de somme, nous ne détaillons ici que la somme.

Cette opération prend en entrée deux `mat sparse` P et Q et retourne une `mat sparse` S . Elle va consister à construire trois vecteurs I_S , J_S et A_S à partir de I_P , J_P , A_P , I_Q , J_Q et A_Q .

Elle commence par un test sur la taille des matrices d'entrée car la somme n'est possible que si elles ont la même taille.

Ensuite, on fait une double boucle sur les indices "classiques" de la matrice : si les matrices d'entrée P et Q sont de taille $m \times n$, on fait varier un indice i de 1 à m et un indice j de 1 à n , et en s'aidant de l'opérateur `[]` construit dans `mat sparse.h`, on récupère les valeurs $P_{i,j}$ et $Q_{i,j}$. Si leur somme est supérieure à 10^{-15} , on ajoute l'élément i à I_S , j à J_S et $P_{i,j} + Q_{i,j}$ à A_S .

Produit scalaire-matrice

Cette opération prend en entrée un `double` λ et une `mat sparse` M , et retourne une `mat sparse` N .

On construit d'abord le premier élément de I_N , J_N et A_N qui sont les mêmes que ceux de I_M , J_M et A_M , car la matrice d'entrée et celle de sortie sont de même dimension.

On fait alors un test : si $\lambda = 0$, on ne fait rien de plus, car les vecteurs construits correspondent à la représentation de la `mat sparse` nulle.

Sinon, on fait une boucle pour parcourir les éléments de la matrice creuse : si le vecteur I_M est de taille $n + 1$ (ce qui revient à dire que M a n éléments non nuls), on fait varier un indice p de 1 à n et si $\lambda * A_M[p] \geq 10^{-15}$, on ajoute respectivement les éléments : $I_M[p]$ à I_N , $J_M[p]$ à J_N et $\lambda * A_M[p]$ à A_N .

Produit matrice-vecteur

Cette opération prend en entrée une `mat sparse` M et un `vector` U contenant des `double`, et retourne un `vector` V contenant des `double`.

Tout d'abord, on fait un test pour vérifier que le produit est bien faisable selon la taille de la matrice et du vecteur d'entrée.

2.2 *Opérations sur la classe*

2014 - 2015

S'il n'y a pas de problème avec la taille, on commence par allouer un vecteur V de la même taille que U .

On fait ensuite une boucle pour modifier chaque coefficient de V . On fait ici varier un indice i de 0 à $J_M[0] - 1$. Chaque coefficient V_i doit en effet récupérer la valeur d'une somme, qui est calculée dans une deuxième boucle.

Cette deuxième boucle porte sur les éléments de I_M , pour utiliser au mieux la structure creuse de la matrice : à l'itération p (si I_M est de taille $n + 1$, p varie de 1 à n), on teste si on est sur la bonne ligne ; si c'est le cas, grâce à $J_M[p]$, on peut ajouter à la somme le terme $A_M[p] * U_{J[p]}$, qui correspond à $M_{I_M[p], J_M[p]} * U_{J[p]}$.

Grâce à cette somme, on a récupéré le coefficient V_i .

A l'issue de la grande boucle, on a donc rempli tous les coefficients de V .

Produit matrice-matrice

Cette opération prend en entrée deux `mat sparse` B_1 et B_2 , de données respectives (A_1, I_1, J_1) et (A_2, I_2, J_2) , et retourne une `mat sparse` B de données (A, I, J) .

Tout d'abord, on fait un test sur la dimension des matrices d'entrée afin de vérifier que leurs tailles sont compatibles pour le produit matriciel.

S'il n'y a pas de problème de taille, on crée trois vecteurs A , I et J en mettant à chacun un premier élément, correspondant à la taille de la matrice.

Ensuite, pour remplir ces trois vecteurs, on fait deux boucles `for` afin de parcourir les indices classiques (i, j) de la matrice produit.

Pour un couple d'indices (i, j) donné, on veut calculer $B_{i,j}$ qui est somme de coefficients des deux matrices de départ :

$$B_{i,j} = \sum_{k=1}^n B_{1\ i,k} B_{2\ k,j}$$

Pour cela, on parcourt I_1 et J_2 avec deux boucles `for` afin d'identifier avec un `if` lorsque $I_1[p] = i$, $J_2[q] = j$ et $J_1[p] = I_2[q]$, et quand c'est le cas, on ajoute $A_1[p]A_2[q]$ à la somme, qui correspond à $B_{i,J_1[p]} B_{I_2[q],j}$.

Ceci permet de faire deux boucles de taille le nombre d'éléments non nuls de chaque matrice, qui a toutes les chances d'être bien inférieur au nombre de colonnes de B_1 si les matrices sont creuses. Ceci permet de réduire considérablement le temps de calcul lorsque le nombre d'éléments non nuls est petit. De plus, si on avait fait la somme grâce à une

2.3 Comparaison des différents produits matriciels

2014 - 2015

seule boucle, il aurait fallu calculer $B_1[i][k]$ et $B_2[k][j]$, qui sont plus longs à calculer, car ils utilisent eux même des boucles dans la fonction []. Cette méthode a également été codée et est laissée en commentaire pour permettre des comparaisons de temps de calcul.

Une fois la somme (`valsum`) complète (les deux dernières boucles finies, à (i, j) donné), on fait un test pour vérifier que cette somme est supérieure en valeur absolue à 10^{-15} , et si c'est le cas, on agrandit les vecteurs A , I , et J : avec respectivement les éléments `valsum`, i et j .

A la fin de toutes ces boucles, on construit enfin la matsparse B à partir de A , I et J .

2.3 Comparaison des différents produits matriciels

Cette section est consacrée à développer les différents tests effectués. Les tests de somme, différence, produit, transposition ne sont pas détaillés dans le rapport, mais ont été effectués sur des matrices de différentes tailles, avant de poursuivre le projet.

Lors de la programmation des produits matrice-matrice et matrice-vecteur, on a utilisé une méthode utilisant la structure creuse de la matrice. On a également utilisé les algorithmes "naïf" des différents produits. En effet, on a créé des fonctions permettant de récupérer $A_{i,j}$ et donc de faire une boucle simple pour calculer $\sum_{k=0}^n a_{i,k}b_{k,j}$ et $\sum_{k=0}^n a_{i,k}u_k$. Cette méthode ne fait pas la différence entre un coefficient nul ou non donc semble plus coûteuse.

On va alors comparer les temps de calcul de ces différentes versions :

Considérons une matrice de taille $100 * 100$ avec 13 coefficients non nuls (non représentée ici pour des raisons évidentes de mise en page). Avec la méthode que l'on pense la plus rapide (celle développée dans le paragraphe précédent), le programme prend 0.601 secondes tandis que pour l'autre, il prend 25.019 secondes. Ceci permet de confirmer nos attentes concernant le temps de calcul, lorsque la matrice est réellement creuse.

Maintenant, nous pouvons utiliser la même matrice pour tester le produit matrice-vecteur. Dans le premier cas, l'algorithme prend met 0.261 secondes tandis que le second, lui, met 0.933 secondes. Là encore nos attentes sont confirmées.

2014 - 2015

3 Affichage

Pour procéder à l'affichage des fichiers `.rsa`, on a utilisé le paquet `iohb` et on s'est inspiré du fichier `sample.c`.

1. La fonction `readHB_info` permet de lire les informations sur la matrice contenue dans le fichier `.rsa` (sa taille, son nombre d'éléments non nuls, son type et si un membre de droite est disponible).
2. La fonction `readHB_newmat_double` permet de stocker la matrice dans les variables déclarées précédemment.
3. La fonction `writeHB_mat_double` permet d'écrire dans un fichier `.rsa` spécifié en argument une matrice sparse définie précédemment.

Dans notre cas, nous n'avons pas de membre de droite spécifié, donc nous n'avons pas eu à utiliser la fonction `readHB_newaux_double`.

Notons aussi une syntaxe un peu déroutante au premier abord de la programmation en C. On a déclaré des *pointeurs* vers des entiers (comme avec `int *L`), des tableaux etc et on a ensuite lancé la fonction `readHB_info` par exemple avec pour arguments ces pointeurs (avec `&L`). Cela permet d'écrire dans les différentes adresses mémoire spécifiées par les pointeurs grâce à la fonction `readHB_info` et de rendre un entier (correspondant à un potentiel signalement d'erreur) à la manière du fonctionnement habituel de la fonction `main`. Cela rompt avec nos habitudes de programmation où l'on aurait plutôt demandé en argument le nom du fichier et rendu en sortie un quintuplet par exemple. Il est clair que la méthode choisie par les auteurs du paquet `iohb` est bien plus facile d'emploi. Elle permet aussi de palier à des problèmes d'allocation de la mémoire (pour la matrice par exemple) avant d'avoir consulté le fichier `.rsa`.

4 Résolution d'équation avec l'algorithme de gradient conjugué

On souhaite ici résoudre un système linéaire à l'aide de la méthode du gradient conjugué dans lequel on va utiliser des matrices en tant que `matsparse`.

4.1 L'algorithme

On s'intéresse ici au système linéaire $AX = f$ où A est symétrique définie positive. Cela revient à minimiser la fonctionnelle suivante :

$$J : v \in \mathbb{R}^n \mapsto J(v) = \frac{1}{2} (Av, v) - (f, v)$$

Page 8

4.1 L'algorithme

2014 - 2015

Pour résoudre ce problème d'optimisation, on utilise l'algorithme de gradient conjugué. Partant d'un vecteur initial x_0 , arbitraire, on pose :

$$r_0 = -\nabla J(x_0) = f - Ax_0$$

et :

$$p_0 = r_0$$

Ceci est l'étape d'initialisation.

Si $\nabla J(x_k) = 0$, l'algorithme est terminé.

Si on a construit $x_1, p_1, r_1, \dots, x_k, p_k, r_k$ (ce qui sous-entend que $\nabla J(x_l)$ est non nul pour $0 \leq l \leq k-1$ car la construction d'un x_l nécessite une division par $\|\nabla J(x_{l-1})\|$), et si $\nabla J(x_k)$ est non nul, on définit alors :

$$a_{k+1} = \frac{r_k^2}{(Ap_k, p_k)}$$

puis les vecteurs :

$$x_{k+1} = x_k + a_{k+1}p_k$$

et :

$$r_{k+1} = -\nabla J(x_{k+1}) = f - Ax_{k+1}$$

puis le nombre :

$$b_{k+1} = \frac{\|\nabla J(x_{k+1})\|^2}{\|\nabla J(x_k)\|^2} = \frac{\|r_{k+1}\|^2}{\|r_k\|^2}$$

qui permet enfin de construire le vecteur :

$$p_{k+1} = r_{k+1} + b_{k+1} * p_{k+1}$$

Dans notre algorithme, nous mettons donc trois tests d'arrêt : $\|Ax - f\| < \epsilon \|Ax_0 - f\|$, qui est le véritable test d'arrêt de l'algorithme (ça revient à dire $\|\nabla J(x)\| < \epsilon \|\nabla J(x_0)\|$), $iter > itmax$, qui permet d'arrêter l'algorithme au bout d'un nombre d'itérations fixé en cas de non convergence, et $(Ap, p) < \epsilon$, pour éviter de faire une éventuelle division par 0 dans le calcul de a . Si la matrice A est de taille n , on sait que l'algorithme de gradient conjugué converge en au plus n itérations. On peut donc fixer $itmax$ à la taille de A , mais on peut l'ajuster manuellement : si on veut une meilleure précision, on peut faire un peu plus d'itérations, mais si la matrice est très grande, on peut s'arrêter arbitrairement avant d'avoir atteint la taille de A .

4.2 Test de l'algorithme

Pour tester l'algorithme du gradient conjugué il nous faut utiliser une matrice symétrique définie positive afin de se garantir la convergence. La matrice du Laplacien en est une qui est très souvent utilisée en analyse numérique, notamment dans la discrétisation d'équations aux dérivées partielles..

Nous avons donc testé l'algorithme sur cette matrice en obtenant des résultats attendus. En effet lors du calcul de la norme $\|AX - f\|$ (notations théoriques, pas celles de l'algorithme), on trouve $\|AX - f\| \approx 10^{-14}$.

```

 2  -1  0  0  0  0  0  0  0  0
-1  2 -1  0  0  0  0  0  0  0
 0 -1  2 -1  0  0  0  0  0  0
 0  0 -1  2 -1  0  0  0  0  0
 0  0  0 -1  2 -1  0  0  0  0
 0  0  0  0 -1  2 -1  0  0  0
 0  0  0  0  0 -1  2 -1  0  0
 0  0  0  0  0  0 -1  2 -1  0
 0  0  0  0  0  0  0 -1  2 -1
 0  0  0  0  0  0  0  0 -1  2

Affichage de l'erreur pour le gradient conjugué :
Nb iterations = 20
err/err0 =8.931174e-016
A*p*p = 5.806016e-028

Affichage de V
 15 30 44 56 65 70 70 64 51 30

Affichage de norme(B1*V-L)

1.464821e-014

Fin du programme main

Process returned 0 (0x0)   execution time : 0.623 s
Press any key to continue.
```

FIGURE 2 – Test gradient conjugué

5 Conclusion

Finalement, nous avons construit une nouvelle classe pour stocker des matrices, que l'on peut utiliser librement comme des matrices classiques puisque nous avons défini tous les opérateurs classiques sur cette classe. Ces opérateurs prennent en compte la nouvelle structure des matrices utilisées, et effectuent donc les diverses opérations de manière différente.

Il est évident que si les matrices que l'on veut utiliser n'ont pas de structure creuse, il sera très contreproductif en terme de temps de calcul d'utiliser pour elles la classe `matparse`, car les opérateurs définis sur cette classe ne seront pas adaptés.

En revanche, sur des matrices qui ont réellement une structure creuse, les tests ont montré qu'il pouvait être beaucoup plus rapide d'utiliser la classe `matparse`, notamment pour le produit matriciel.

Enfin, pour valider l'utilisation pratique de cette nouvelle classe, nous l'avons utilisée pour résoudre un problème de calcul scientifique : un système linéaire par méthode de gradient conjugué.

On se rend compte en lisant l'algorithme de gradient conjugué lui-même (dans le fichier `gradient_conjugué.cpp`) que l'utilisation des `matparse` n'est pas visible dans le code, grâce aux opérateurs sur la classe construits précédemment, ce qui rend l'utilisation de la classe aussi pratique que celle des matrices classiques.

3.5 Stage M2 - Conception d'obstacles invisibles à fort contraste dans un guide d'ondes

Conception d'obstacles invisibles à fort contraste dans un guide d'ondes

BERA Antoine

mars 2016 - août 2016



Stage effectué au laboratoire *POEMS*, sous l'encadrement d'Anne-Sophie Bonnet Ben-Dhia et de Lucas Chesnel



Table des matières

Table des matières	1
1 Introduction	2
2 Description de la méthode sur un cas modèle simplifié	5
2.1 Le théorème	5
2.2 Illustrations - exemples	9
2.2.1 Exemple du parcours d'un cercle	9
2.2.2 Exemple de parcours sur une courbe à courbure non constante	11
3 Un problème d'invisibilité dans un guide d'onde	14
3.1 Position du problème	14
3.2 Cas du guide non perturbé	14
3.3 Le problème de diffraction	16
3.4 Conservation d'énergie	18
3.5 Les coefficients de scattering	19
3.6 Un premier déplacement	20
3.7 Choix de μ_0 , μ_1 et μ_2	21
3.8 Résolution par méthode du point fixe	23
3.9 Itération de la méthode	23
3.10 Invisibilité totale	26
4 Programmation de l'algorithme	27
4.1 Écriture de l'algorithme	27
4.2 <i>Xlife++</i>	28
4.3 Non réflexion	28
4.3.1 Initialisation de l'algorithme	29
4.3.2 Itération de l'algorithme	33
4.4 Invisibilité	35
5 Conclusion et ouvertures	37
Références	39

1 Introduction

On a tous déjà entendu parler d'invisibilité, ou même rêvé de devenir invisible... Mais que signifie réellement ce terme "invisible" ?

Pour fixer les idées, parlons d'abord de l'invisibilité lumineuse. La perception que j'ai d'un objet est traduite par les ondes lumineuses pénétrant dans mes yeux. On parle alors d'invisibilité dans la direction de mes yeux lorsque les ondes y pénétrant sont les mêmes avec ou sans objet. Mais les ondes lumineuses arrivant sur l'objet ne sont pas dirigées uniquement vers mes yeux. C'est pourquoi on parlera d'invisibilité lorsque, dans toutes les directions, les ondes se propagent de la même manière avec ou sans objet.

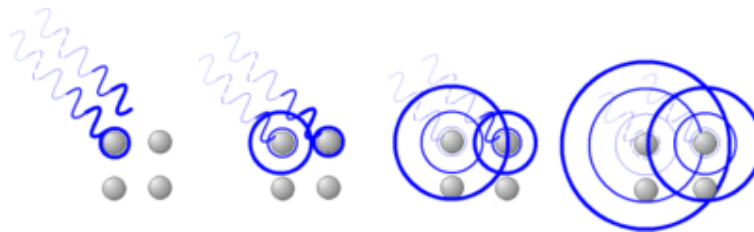


FIGURE 1 – Onde incidente et onde diffractée.

Dans notre étude, nous allons nous fixer un champ incident, et nous intéresser à l'invisibilité d'un objet dans un nombre fini de directions. Cette restriction est effectuée puisque dans la pratique, lorsque l'on observe quelque chose, on n'utilise qu'un nombre fini d'éléments de mesure. Ici, nous nous intéresserons plus précisément au cas d'un obstacle placé dans un guide d'ondes, ce qui restreint à 2 directions d'invisibilité. Nous enverrons une *onde incidente* provenant de la gauche, se dispersant sur l'obstacle en produisant une *onde réfléchie* (repartant vers la gauche) et une *onde transmise* (partant vers la droite).

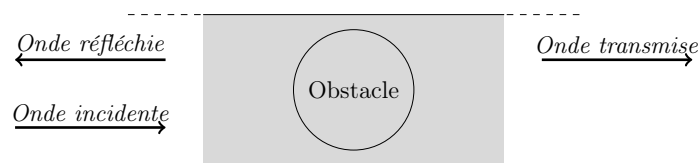


FIGURE 2 – Guide d'ondes.

L'objectif de cette étude est de parvenir à construire un obstacle pénétrable, invisible dans un guide d'onde. L'obstacle sera alors une inconnue dans toute notre démarche.

Fixons donc un champ incident, u_i , provenant de $-\infty$. Nous savons qu'à fréquence donnée,

le champ diffracté se décompose en somme d'ondes propagatives (se propageant à l'infini) et d'ondes évanescentes (exponentiellement décroissantes en $\pm\infty$). On prendra ici le cas où la fréquence est suffisamment petite de telle sorte qu'il n'y ait qu'une seule onde propagative.

On va pouvoir ainsi définir les coefficients de réflexion et de transmission (R et T) représentant l'amplitude du champ total (champ incident + champ diffracté) dans les deux directions possibles du guide. L'objectif est d'obtenir $R = 0$ (non réflexion) et $T = 1$ (transmission parfaite). L'obstacle sera caractérisé par l'indice ρ du milieu en chaque point, qui est donc une inconnue du problème. En effet, ces deux coefficients R et T dépendent de l'indice dans tout le milieu, donc du défaut présent dans le guide d'ondes.

On remarque déjà que l'objectif de ces travaux est de trouver des zéros non triviaux d'une fonction donnée : par exemple, si on cherche à résoudre uniquement la non réflexion, il nous faut trouver ρ tel que $R(\rho) = 0$ (on peut également chercher à résoudre $R(\rho) = 0$ et $T(\rho) - 1 = 0$ simultanément afin d'avoir invisibilité totale). Intuitivement on imagine que pour annuler un nombre fini de valeurs avec un paramètre qui vit dans un espace de dimension infinie (une fonction), il y aura probablement de multiples façon d'en trouver. Nous ne devons pas seulement en trouver, mais plutôt construire des zéros on donnant une forme souhaitée.

Un principe important de la physique est la conservation de l'énergie. Ce principe affirme que la variation d'énergie d'un système est égale à la somme des énergies entrantes et sortantes du système. Dans notre cas, l'énergie entrant est transportée par le champ incident dans le guide d'ondes. L'énergie transportée par un champ u à travers une surface S est $Im(\int_S \bar{u} \frac{\partial u}{\partial \mathbf{n}} d\sigma)$. On peut alors calculer l'énergie entrant dans le guide d'onde (via le champ incident) ainsi que l'énergie sortant du guide d'onde (via le champ diffracté). Ce principe nous permet donc de donner une relation entre les coefficients R et T : $|R|^2 + |T|^2 = 1$. Nous détaillerons plus loin les calculs permettant d'obtenir ce résultat.

Si il n'y a pas d'obstacle, l'onde passe sans être diffractée, c'est une situation d'invisibilité connue, permettant d'annuler notre fonction : $\rho = 0$ est un point d'annulation de R et de $T - 1$. Nous allons alors utiliser une méthode de continuation, consistant à partir d'un point d'annulation connu (ici aucun obstacle, $\rho = 0$) afin de trouver un autre point d'annulation dans un voisinage du premier. Le voisinage dans lequel on a trouvé ce point d'annulation peut être petit, mais l'objectif est d'itérer le processus afin de s'éloigner le plus possible du point de départ. Il a déjà été prouvé que l'on peut construire des perturbations non triviales invisibles, et mon travail consiste en la compréhension et la réalisation informatique de cette première étape afin de pouvoir itérer la méthode ; c'est à dire partir de l'obstacle invisible trouvé pour en construire un nouveau etc. Cette démarche sera effectuée en gardant pour objectif de construire des obstacles invisibles pouvant éventuellement être réalisés physiquement.

Je développerai dans un premier temps, sur un cas simple, un exemple de méthode de conti-

2016

uation permettant de comprendre les démonstrations futures. C'est un paragraphe à visées pédagogiques, aidant à se fixer les idées pour la suite du document. Ensuite j'expliquerai comment s'écrivent les équations du problème et donnerai la fonction que l'on cherche à annuler. Enfin, avant de passer à la programmation et à l'étude des résultats, nous expliquerons le processus de continuation dans le cas d'un obstacle pénétrable dans un guide d'ondes.

Page 4

2 Description de la méthode sur un cas modèle simplifié

Avant d'entrer dans le vif du sujet, nous allons commencer par un paragraphe à visées pédagogiques. On se place en effet dans un cadre qui est une simplification du modèle. La méthode développée ici sera réutilisée ultérieurement. Rappelons que l'un de nos objectifs est de trouver, dans le guide d'onde, des défauts ne créant aucune réflexion. Pour cela nous cherchons à annuler le coefficient de réflexion $R \in \mathbb{C}$ (aucune réflexion) dépendant d'un défaut représenté par une fonction contraste $\rho \in L^\infty(\mathcal{O})$, où \mathcal{O} est une partie bornée de \mathbb{R}^2 : mathématiquement, cela revient à étudier les zéros de la fonction

$$F : L^\infty(\mathcal{O}) \rightarrow \mathbb{R}^2 \\ \rho \mapsto (\operatorname{Re}(R(\rho)) , \operatorname{Im}(R(\rho))),$$

définie sur L^∞ (de dimension infinie). Dans cette optique, nous utiliserons le fait que $F(0) = 0$ (lorsqu'il n'y a pas d'obstacle, il n'y a pas de réflexion).

Dans ce paragraphe, nous allons tenter de trouver les zéros d'une fonction $F : \mathbb{R}^2 \rightarrow \mathbb{R}$ qui est définie sur un espace de dimension finie 2, et nous supposons connu un couple (x_i, y_i) annulant F . Définissons l'ensemble $\mathcal{C} = \{(x, y) \in \mathbb{R}^2 \mid F(x, y) = 0\}$, lieu des zéros de F , qui sera généralement une courbe de \mathbb{R}^2 .

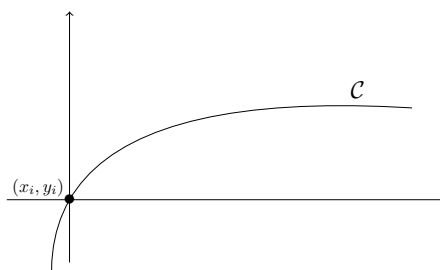


FIGURE 3 – Courbe $F = 0$.

2.1 Le théorème

Supposons que F soit de classe \mathcal{C}^1 et notons $dF(x_i, y_i)$ sa différentielle en (x_i, y_i) qui est une application linéaire de \mathbb{R}^2 dans \mathbb{R} . Nous utiliserons ici parfois le vecteur gradient de F , noté $\nabla F(x_i, y_i)$, de sorte que $dF(x_i, y_i)(x, y) = \nabla F \begin{pmatrix} x_i \\ y_i \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$. Nous cherchons à construire $(x_f, y_f) \neq (x_i, y_i)$ tel que $F(x_f, y_f) = 0$. Puisque l'on connaît un zéro (x_i, y_i) , nous allons essayer de trouver (x_f, y_f) dans son voisinage. Pour ce faire, écrivons un développement limité à l'ordre 2 de F en (x_i, y_i) . Soient $(x, y) \in \mathbb{R}^2$ et $\varepsilon > 0$, on peut écrire :

$$\begin{aligned} F\begin{pmatrix} x_i + \varepsilon x \\ y_i + \varepsilon y \end{pmatrix} &\underset{\varepsilon \rightarrow 0}{=} F\begin{pmatrix} x_i \\ y_i \end{pmatrix} + \varepsilon \nabla F\begin{pmatrix} x_i \\ y_i \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \mathcal{O}(\varepsilon^2) \\ &\underset{\varepsilon \rightarrow 0}{=} \varepsilon \nabla F\begin{pmatrix} x_i \\ y_i \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \mathcal{O}(\varepsilon^2). \end{aligned}$$

D'après le théorème du rang, $\dim(\ker dF(x_i, y_i)) \geq 1$. Si dans l'expression précédente on prend $(x, y) = (x_0, y_0) \in \ker dF(x_i, y_i)$, on obtient pour $\varepsilon > 0$ assez petit un nouveau couple $(x_i + \varepsilon x_0, y_i + \varepsilon y_0)$ proche de \mathcal{C} au sens où :

$$F\begin{pmatrix} x_i + \varepsilon x_0 \\ y_i + \varepsilon y_0 \end{pmatrix} \underset{\varepsilon \rightarrow 0}{=} \mathcal{O}(\varepsilon^2)$$

Comme l'illustre la Figure 4, on s'est quand même "éloigné" de la courbe et on aimerait s'en rapprocher. Pour cela partons de ce point et dirigeons nous dans la direction d'un vecteur (x_1, y_1) de manière à essayer d'annuler le terme en $\mathcal{O}(\varepsilon^2)$. On voit sur la figure que si l'on part orthogonalement à la direction (x_0, y_0) , on peut espérer se trouver sur la courbe, pour peu que l'on ait choisit ε suffisamment petit.

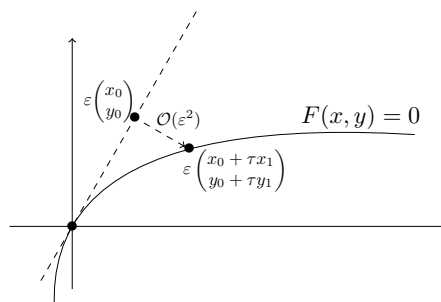


FIGURE 4 – Courbe $F = 0$ avec $(x_i, y_i) = (0, 0)$.

Soit $(x_1, y_1) \in \mathbb{R}^2$ quelconque, on cherche à trouver $\varepsilon > 0$ et $\tau \in \mathbb{R}$ tels que :

$$F\begin{pmatrix} x_i + \varepsilon(x_0 + \tau x_1) \\ y_i + \varepsilon(y_0 + \tau y_1) \end{pmatrix} = 0. \quad (1)$$

Or,

$$F\begin{pmatrix} x_i + \varepsilon x_0 + \varepsilon \tau x_1 \\ y_i + \varepsilon y_0 + \varepsilon \tau y_1 \end{pmatrix} \underset{\varepsilon \rightarrow 0}{=} \varepsilon \tau \nabla F\begin{pmatrix} x_i \\ y_i \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \mathcal{O}(\varepsilon^2).$$

Afin de pouvoir revenir sur la courbe, il nous faut une direction (x_1, y_1) qui ne soit pas dans le noyau de $dF(x_i, y_i)$. C'est pourquoi nous allons à présent supposer que $\nabla F(x_i, y_i)$ est non nul.

Cette hypothèse induit en particulier que l'ensemble \mathcal{C} est, au moins localement, une courbe de \mathbb{R}^2 . En effet les hypothèses du théorème des fonctions implicites sont vérifiées : la non nullité du gradient en un point assure que l'une de ses deux composantes, au moins, n'est pas nulle (par

exemple celle suivant x) et le lieu des zéros est donc localement une courbe.

Puisque $\nabla F(x_i, y_i) \neq 0$, nous pouvons trouver (x_1, y_1) tel que $\nabla F(x_i, y_i) \cdot (x_1, y_1) = 1$. Dans la pratique, si l'on connaît ∇F , on peut prendre $(x_1, y_1) = \frac{\nabla F(x_i, y_i)}{\|\nabla F(x_i, y_i)\|^2}$.

Finalement on obtient, en remplaçant :

$$0 = F \begin{pmatrix} x_i + \varepsilon x_0 + \varepsilon \tau x_1 \\ y_i + \varepsilon y_0 + \varepsilon \tau y_1 \end{pmatrix} = \varepsilon \tau + \varepsilon^2 \tilde{F}_\varepsilon(\tau).$$

où l'on a posé

$$\tilde{F}_\varepsilon(\tau) = \frac{-1}{\varepsilon^2} \left(\varepsilon \tau - F \begin{pmatrix} x_i + \varepsilon x_0 + \varepsilon \tau x_1 \\ y_i + \varepsilon y_0 + \varepsilon \tau y_1 \end{pmatrix} \right).$$

Ainsi τ est recherché comme un éventuel point fixe de la fonction $t \mapsto -\varepsilon F_\varepsilon(t)$:

$$\varphi_\varepsilon(t) = t - \frac{1}{\varepsilon} F \begin{pmatrix} x_i + \varepsilon x_0 + \varepsilon t x_1 \\ y_i + \varepsilon y_0 + \varepsilon t y_1 \end{pmatrix}$$

Avant de montrer que φ_ε vérifie les conditions du théorème de point fixe, donnons l'expression de sa dérivée :

$$\varphi'_\varepsilon(t) = 1 - \nabla F \begin{pmatrix} x_i + \varepsilon x_0 + \varepsilon t x_1 \\ y_i + \varepsilon y_0 + \varepsilon t y_1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}.$$

Nous cherchons à montrer qu'il existe $\varepsilon_0 > 0$ et un fermé $U \subset \mathbb{R}$ tels que pour tout $\varepsilon < \varepsilon_0$, $\varphi_\varepsilon(U) \subset U$ et φ_ε soit une contraction sur U .

Remarquons que $\varphi'_0(0) = 0$ et que $(\varepsilon, t) \mapsto \varphi'_\varepsilon(t)$ est continue au voisinage de $(0, 0)$ par composition de fonctions continues (puisque F est \mathcal{C}^1). Ainsi, il existe $r > 0$ tel que pour tout $\varepsilon < r$ et tout $|t| \leq r$, $|\varphi'_\varepsilon(t)| < \frac{1}{2}$. Notons $U = [-r, r]$.

Prenons $t \in U$ et $\varepsilon < r$. On a ,

$$|\varphi_\varepsilon(t)| \leq |\varphi_\varepsilon(0)| + |\varphi_\varepsilon(t) - \varphi_\varepsilon(0)| \leq \frac{1}{\varepsilon} \left| F \begin{pmatrix} x_i + \varepsilon x_0 \\ y_i + \varepsilon y_0 \end{pmatrix} \right| + \frac{1}{2} |t - 0|.$$

Or $\left| F \begin{pmatrix} x_i + \varepsilon x_0 \\ y_i + \varepsilon y_0 \end{pmatrix} \right| = \mathcal{O}(\varepsilon^2)$, donc il existe $\varepsilon_0 < r$ tel que pour tout $\varepsilon < \varepsilon_0$ on ait :

$$\frac{1}{\varepsilon} \left| F \begin{pmatrix} x_i + \varepsilon x_0 \\ y_i + \varepsilon y_0 \end{pmatrix} \right| \leq \frac{1}{\varepsilon} \frac{r\varepsilon}{2}.$$

Ainsi, pour tout $\varepsilon < \varepsilon_0$ et tout $t \in U$,

$$|\varphi_\varepsilon(t)| \leq \frac{r}{2} + \frac{r}{2} = r.$$

Donc pour tout $\varepsilon < \varepsilon_0$, $\varphi_\varepsilon(U) \subset U$ et φ_ε est contractante sur U (par inégalité des accroissements finis). Ainsi, pour tout $\varepsilon < \varepsilon_0$, il existe un unique $\tau \in U$ (avec $0 \in U$), tel que

$$F \begin{pmatrix} x_i + \varepsilon x_0 + \varepsilon \tau x_1 \\ y_i + \varepsilon y_0 + \varepsilon \tau y_1 \end{pmatrix} = 0.$$

Rappel du théorème de point fixe : Si U est un espace métrique complet non vide, toute fonction contractante $f : U \mapsto U$ admet une unique point fixe dans U .

Le fait que 0 appartienne à U est important informatiquement puisque cela nous donne un point de départ à l'algorithme de point fixe. En effet, cette démonstration et son application dans la suite reposent principalement sur le théorème de point fixe qui nous fournit un algorithme.

Afin de terminer la démonstration, il faut justifier du fait que le zéro de F trouvé grâce au théorème de point fixe est différent de (x_i, y_i) . Si ils étaient égaux, en divisant par $\varepsilon > 0$, on aurait $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \tau \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ ce qui est impossible puisque ces vecteurs sont linéairement indépendants (l'un est dans le noyau de $dF(x_i, y_i)$, l'autre ne l'est pas).

En conclusion, nous avons démontré le résultat suivant :

Théorème 1 Soit $F : \mathbb{R}^2 \rightarrow \mathbb{R}$ de classe \mathcal{C}^1 . Soit (x_i, y_i) un zéro de F . Supposons que $\nabla F(x_i, y_i)$ soit non nul, notons (x_0, y_0) un élément de son noyau et (x_1, y_1) un antécédent de 1 par $\nabla F(x_i, y_i)$. Alors il existe $\varepsilon_0 > 0$ tel que pour tout $0 < \varepsilon < \varepsilon_0$, il existe $\tau \in \mathbb{R}$ tel que :

$$F \begin{pmatrix} x_i + \varepsilon x_0 + \varepsilon \tau x_1 \\ y_i + \varepsilon y_0 + \varepsilon \tau y_1 \end{pmatrix} = 0.$$

De plus, τ s'obtient grâce à l'algorithme de point fixe.

Remarque 1 Dans ce théorème nous avons imposé la non nullité de $\nabla F(x_i, y_i)$. Mais que peut-il se passer si $\nabla F(x_i, y_i) = 0$? Quels problèmes peut-on rencontrer dans ce cas précis? Le lieu des zéros peut-il être localement une courbe? Une première idée est de penser à une fonction localement constante nulle, ce qui, pour nous, n'a que peu d'intérêt. Le cas de la fonction $F(x, y) = x^2 + y^2$ est aussi un cas particulier : elle n'admet que $(0, 0)$ comme point d'annulation, qui est donc un zéro isolé. Malheureusement il y a aussi une troisième possibilité, qui serait que F n'est pas adaptée au problème. En effet, considérons une fonction F vérifiant les hypothèses du théorème en chaque point de \mathcal{C} , qui est alors une courbe de \mathbb{R}^2 . En posant $\tilde{F} = F^2$, on obtient la même courbe d'annulation que celle de F .

Cependant, calculons $\nabla \tilde{F}(x, y)$ pour $(x, y) \in \mathcal{C}$:

$$\nabla \tilde{F} \begin{pmatrix} x \\ y \end{pmatrix} = 2 \underbrace{F(x,y)}_{=0} \nabla \tilde{F} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

On ne peut donc pas imaginer, avec cette méthode, parcourir la courbe \mathcal{C} en utilisant la fonction \tilde{F} .

2.2 Illustrations - exemples

Nous allons maintenant donner quelques applications de ce théorème et présenter également des situations dans lesquelles il ne s'applique pas. Sous python, nous avons réalisé l'algorithme consistant à itérer le théorème précédent : en démarrant, par exemple, avec $(x_i, y_i) = (0, 0)$ puis en prenant $(x_i, y_i) = (x_f, y_f)$ comme nouveau point de départ à chaque étape.

Comme on l'a vu, il existe ε assez petit pour lequel l'algorithme va converger. Pour en trouver un, nous démarrerons avec une valeur de ε et tant que l'algorithme de point fixe n'a pas convergé (on dira ici qu'il n'y a pas convergence si on a effectué plus 100 itérations), on prendra ε de plus en plus petit.

2.2.1 Exemple du parcours d'un cercle

Considérons la fonction F :

$$F(x, y) = (x - 1)^2 + (y - 1)^2 - 2.$$

La courbe d'annulation \mathcal{C} , de F , passe par $(0, 0)$ comme on peut le voir sur la Figure 5 :

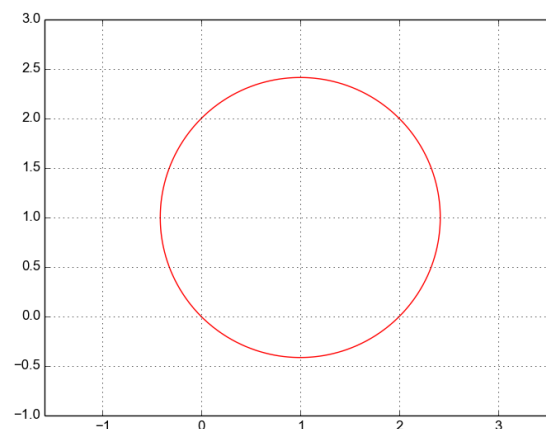


FIGURE 5 – Courbe \mathcal{C} .

Nous pouvons à présent procéder à l'écriture de l'algorithme de point fixe. A chaque étape, il faut avoir accès à la différentielle afin de déterminer les vecteurs (x_0, y_0) et (x_1, y_1) . Le vecteur gradient, caractérisant la différentielle, s'écrit :

$$\nabla F \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} 2(x_i - 1) \\ 2(y_i - 1) \end{pmatrix}.$$

Ainsi on prendra, par exemple, $(x_0, y_0) = \frac{(-2(y_i-1), 2(x_i-1))}{\|\nabla F(x_i, y_i)\|}$ et $(x_1, y_1) = \frac{\nabla F(x_i, y_i)}{\|\nabla F(x_i, y_i)\|^2}$. Remarquons que nous avons volontairement normalisé le vecteur (x_0, y_0) de manière à ce que la norme du déplacement $\varepsilon(x_0, y_0)$ soit de longueur ε . Ceci s'observera graphiquement. On a lancé l'algorithme avec $\varepsilon = 2$, qui n'a pas fonctionné. En effet, on remarque sur la Figure 6 que si l'on part orthogonalement au trait bleu (c'est à dire orthogonalement à la direction du noyau), on ne croquera jamais la courbe. C'est pourquoi nous lui avons demandé de recommencer avec $\varepsilon = 0.5$, et nous avons obtenu la Figure 7 après 15 étapes.

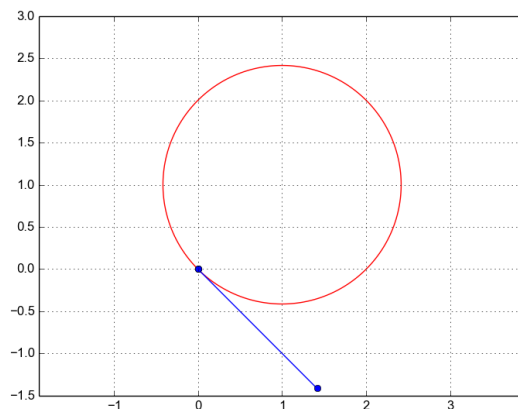
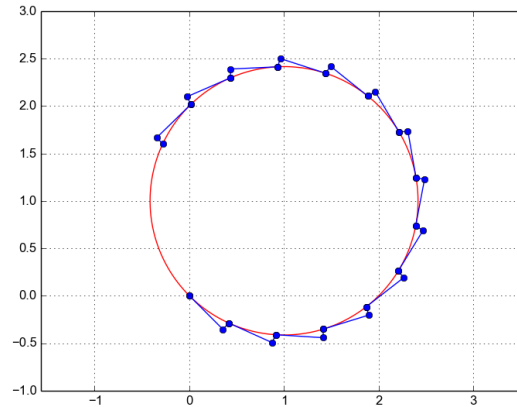


FIGURE 6 – Déplacement sur le cercle - $\varepsilon = 2$.

Sur cette figure nous avons, à chaque étape, représenté (x_i, y_i) , $(x_i, y_i) + \varepsilon(x_0, y_0)$ et (x_f, y_f) . On peut remarquer la régularité du déplacement, liée à la régularité de la courbure du cercle. Nous verrons que sur les exemples de courbure moins régulière, le déplacement (lié à la valeur ε) sera moins régulier.

Nous avons donc réussi, sur un cas simple, à trouver des zéros d'une fonction F en partant d'une solution connue. La satisfaction est d'autant plus grande que les zéros sont éloignés du premier et que ε n'est pas aussi petit que ce que l'on pouvait imaginer avec une méthode utilisant un développement limité.

FIGURE 7 – Déplacement sur le cercle - $\varepsilon = 0.5$.

À présent se pose une question : peut-on trouver, à chaque étape, le ε optimal (c'est à dire celui pour lequel le point fixe converge, mais si on l'augmente légèrement, le point fixe ne converge plus) ?

La recherche d'un ε optimal, a été réussie informatiquement. En effet il suffit de procéder par méthode de dichotomie : partons du principe que l'on connaît ε_1 pour lequel l'algorithme de point fixe converge et ε_2 pour lequel il ne converge pas. Regardons ce qu'il se passe pour $\varepsilon = \frac{\varepsilon_1 + \varepsilon_2}{2}$ et ainsi de suite.

Pour se trouver dans la situation initiale précédente, il faut se fixer un ε et regarder s'il permet la convergence ou non. S'il y a convergence, nous étudions la convergence avec 10ε , etc. jusqu'à trouver un cas pour lequel il n'y a pas convergence. Inversement, s'il n'y a pas convergence avec le ε initial, on le divise par 10 jusqu'à convergence.

Nous avons codé cette méthode sur le cercle. En partant de $\varepsilon = 0.5$ ou $\varepsilon = 2$ nous avons obtenu la même figure, présentée en figure 8 :

2.2.2 Exemple de parcours sur une courbe à courbure non constante

Considérons la fonction F suivante :

$$F(x, y) = (x^2 + y^2)^3 - x^2 - y$$

La courbe \mathcal{C} des zéros de F passe toujours par $(0, 0)$:

On remarque ici que la première étape a convergé avec $\varepsilon = 0.5$, mais qu'ensuite cela n'a pas

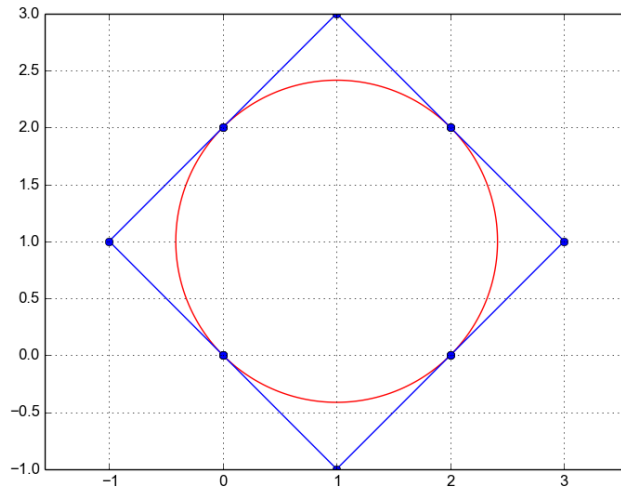


FIGURE 8 – Déplacement optimal sur le cercle.

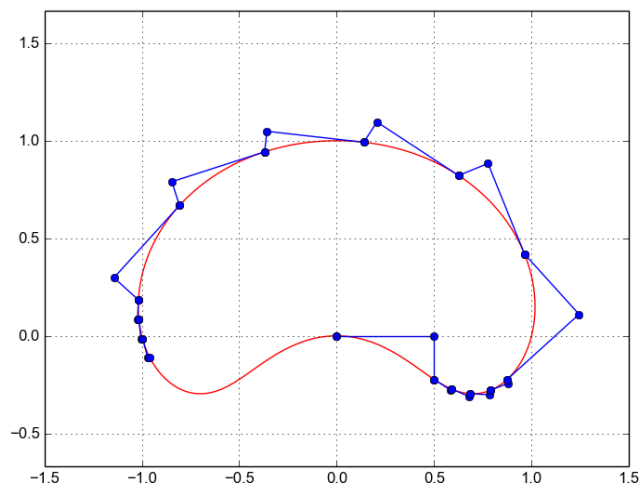


FIGURE 9 – Courbe \mathcal{C} .

convergé. En prenant, par exemple, $\varepsilon = 0.1$ nous avons pu continuer le parcours. Cette remarque va avec l'intuition que l'on peut avoir concernant le lien entre la courbure et la valeur de ε : Si la courbe tourne "très fort" (faible rayon de courbure), on s'attend à un faible ε , alors que si la

2.2 Illustrations - exemples

2016

courbe a un fort rayon de courbure, on peut espérer que l'algorithme de point fixe converge pour de grandes valeurs de ε .

Stoppons ici la description en dimension finie et revenons à présent au problème d'invisibilité dans le guide d'onde.

3 Un problème d'invisibilité dans un guide d'onde

3.1 Position du problème



FIGURE 10 – Guide d'ondes de référence / Guide d'ondes perturbé.

On s'intéresse, maintenant, à la propagation des ondes acoustiques en régime harmonique dans le guide d'ondes $\Omega = \mathbb{R} \times]0, 1[$. Ce guide d'ondes sera perturbé sur un domaine \mathcal{O} par un obstacle pénétrable, autrement dit, l'indice du milieu n'est pas constant égal à 1 sur \mathcal{O} . Le vecteur normal au bord $\partial\Omega$, dirigé vers l'extérieur sera noté \mathbf{n} .

On va donc s'intéresser ici au cas où l'indice, $1 + \rho$, du milieu n'est pas constant sur Ω ; en particulier supposons qu'il est constant égal à 1 en dehors du domaine \mathcal{O} , autrement dit que ρ est une fonction supportée sur $\overline{\mathcal{O}}$. Le domaine \mathcal{O} est considéré comme un obstacle sur lequel une onde incidente u_i va arriver et induire un champ diffracté u_s . Notons alors $u = u_i + u_s$ le champ total dans le guide d'ondes, somme du champ incident et du champ diffracté.

Afin de modéliser l'expérience précédente, on considère les problèmes (2) (problème de Dirichlet) et (3) (problème de Neumann) :

$$\begin{cases} \Delta u + k^2(1 + \rho)u = 0 & \text{dans } \Omega \\ u = 0 & \text{sur } \partial\Omega \end{cases} \quad (2)$$

$$\begin{cases} \Delta u + k^2(1 + \rho)u = 0 & \text{dans } \Omega \\ \partial_{\mathbf{n}} u = 0 & \text{sur } \partial\Omega. \end{cases} \quad (3)$$

Dans ces équations, u représente la pression dans le guide d'ondes, k est le nombre d'onde proportionnel à la fréquence ω des oscillations harmoniques et Δ est l'opérateur laplacien. Nous allons à présent, dans tout le rapport, nous intéresser au cas de Neumann.

3.2 Cas du guide non perturbé

Avant de regarder le problème (3), on va étudier le cas du guide de référence, sans perturbation ($\rho = 0$), appelé problème de Helmholtz et vérifiant l'équation (4). Pour ce problème simple, on peut calculer analytiquement les solutions à variables séparées. Ces solutions permettront de comparer le champ diffracté (dans le cas du guide perturbé), avec le champ issu d'un guide non perturbé.

$$\begin{cases} \Delta u + k^2 u = 0 & \text{dans } \Omega \\ \partial_{\mathbf{n}} u = 0 & \text{sur } \partial\Omega. \end{cases} \quad (4)$$

Effectuons une séparation de variables afin de résoudre ce problème : cherchons u de la forme $u(x, y) = \tilde{u}(y)e^{i\beta x}$. Notons (λ_n, φ_n) les éléments propres de l'opérateur Laplacien sur $]0, 1[$ avec conditions de Neumann (trouver $(\lambda, \varphi) \in \mathbb{R} \times L^2(]0, 1[)$ tel que $-\Delta\varphi = \lambda\varphi$ et $\varphi'(0) = \varphi'(1) = 0$). Ces éléments vérifient :

$$\begin{aligned} 0 &= \lambda_0 < \lambda_1 \leq \dots \leq \lambda_n \leq \dots \\ \varphi_n \in H^1(]0, 1[) \quad (\varphi_m, \varphi_n)_{L^2(]0, 1[)} &= \delta_{n,m} \quad \text{pour } m, n \in \mathbb{N} \end{aligned}$$

Ici, on peut donner l'expression des éléments propres (λ_n, φ_n) : $\lambda_n = (n\pi)^2$ et $\varphi_n(y) = \cos(n\pi y)$.

Notons que la famille (φ_n) forme une base hilbertienne de $L^2(]0, 1[)$. Ainsi, une solution u du problème de Helmholtz s'écrit :

$$u(x, y) = \sum_{n=0}^{\infty} a_n(x) \varphi_n(y), \quad (5)$$

avec

$$a_n(x) = \int_{]0, 1[} u(x, y) \varphi_n(y) dy.$$

Par dérivation sous le signe intégrale, on vérifie que chaque fonction a_n est solution de l'équation différentielle $\frac{d^2 a_n}{dx^2} = (-k^2 + \lambda_n) a_n(x)$ qui s'écrit donc

$$a_n(x) = A_n^+ e^{i\beta_n x} + A_n^- e^{-i\beta_n x} \quad \text{où } \beta_n = \sqrt{k^2 - \lambda_n} \quad \text{et } A_n^\pm \in \mathbb{R}. \quad (6)$$

Ainsi, en regroupant (5) et (6), une solution se décomposera selon les modes

$$w_n^\pm(x, y) = \frac{1}{\sqrt{2|\beta_n|}} \varphi_n(y) e^{\pm i\beta_n x}.$$

Remarquons que l'on a artificiellement introduit le terme $\frac{1}{\sqrt{2|\beta_n|}}$. C'est un coefficient utile dans le cas d'une résolution multimodale, ce qui ne sera pas notre cas. Pour respecter le cas plus général, nous le conserverons. On prendra comme convention que la racine complexe, si elle n'est pas réelle, est calculée en prenant la racine complexe de partie imaginaire positive.

Rappelons que $\lambda_n = (n\pi)^2$ et $\varphi_n(y) = \cos(n\pi y)$. Nous allons maintenant fixer k tel que $0 < k < \pi = \sqrt{\lambda_1}$. Dans cette situation, on remarque que pour $n > 0$, β_n est imaginaire pur et donc que w_n^+ décroît exponentiellement en $+\infty$ et croît exponentiellement en $-\infty$ (et inversement pour w_n^-). Ces modes sont généralement appelés modes évanescents.

On remarquera que $\lambda_0 = 0$ et que φ_0 est la fonction constante égale à 1, qui donnera la solution appelée mode plan. Pour $n = 0$, $\beta_0 = k$ est strictement positif, ce qui implique que les

3.3 Le problème de diffraction

2016

fonctions w_0^\pm sont oscillantes. Ces modes sont appelés modes propagatifs.

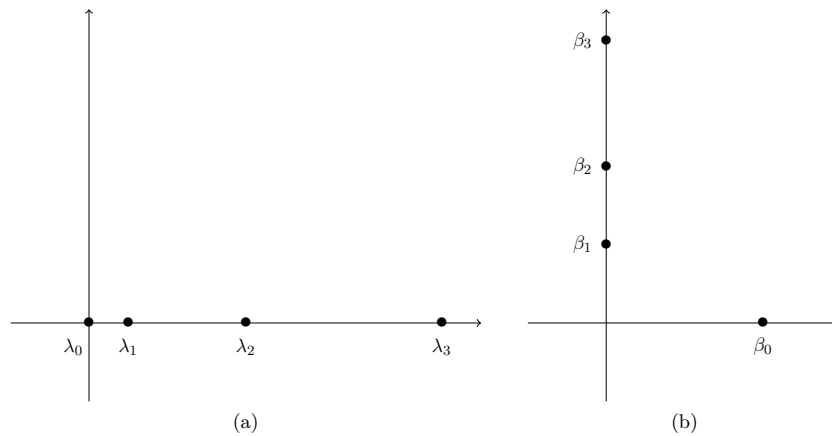


FIGURE 11 - (a) λ_n et (b) β_n avec $0 < k < \pi = \sqrt{\lambda_1}$.

3.3 Le problème de diffraction

Maintenant que l'on a résolu le cas du guide sans perturbation, nous allons envoyer une onde provenant de $-\infty$ sur une perturbation caractérisée par un contraste ρ et tenter d'obtenir le champ diffracté. On va donc considérer comme champ incident le mode w_0^+ que l'on notera u_i ou w_+ (selon les circonstances) dans la suite : $w_+(x, y) = \frac{1}{\sqrt{2k}} e^{ikx}$. Remarquons que $\overline{w_+} = w_-$.

Fixons $L > 0$ et notons $\Omega_L =]-L, L[\times]0, 1[$ et $\Sigma_{\pm L} = \{\pm L\} \times]0, 1[$. Nous nous intéresserons à des perturbations supportées par Ω_L , c'est à dire que nous considérons $\mathcal{O} \subset \Omega_L$:

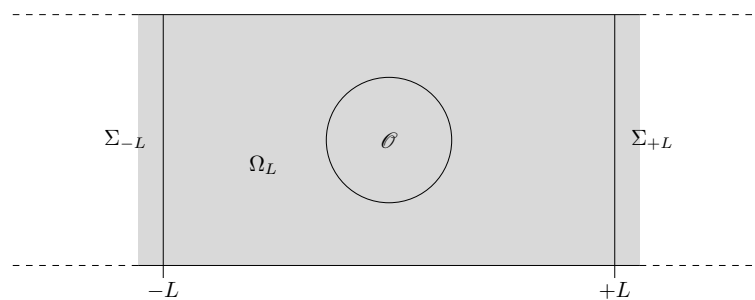


FIGURE 12 - Domaine Ω_L .

Rappelons le problème que l'on cherche à résoudre :

$$\begin{cases} \Delta u + k^2(1 + \rho)u = 0 & \text{dans } \Omega \\ \partial_{\mathbf{n}} u = 0 & \text{sur } \partial\Omega. \end{cases} \quad (7)$$

On dira qu'un champ $u \in H^1_{loc}(\Omega)$ est sortant s'il s'écrit de forme

$$u = s_+\chi_+w_+ + s_-\chi_-w_- + \tilde{u}, \quad (8)$$

où $s_{\pm} \in \mathbb{C}$, $\tilde{u} \in H^1(\Omega)$ est exponentiellement décroissant en $\pm\infty$ et $\chi_+ \in \mathcal{C}^\infty(\Omega)$ (resp. χ_-) est une fonction de troncature constante nulle sur $] -\infty, -L[$ (resp. sur $]L, +\infty[$), constante égales à 1 sur $]L, +\infty[$ (resp. sur $] -\infty, -L[$) et quelconque sur $[-L, L]$. Par exemple :

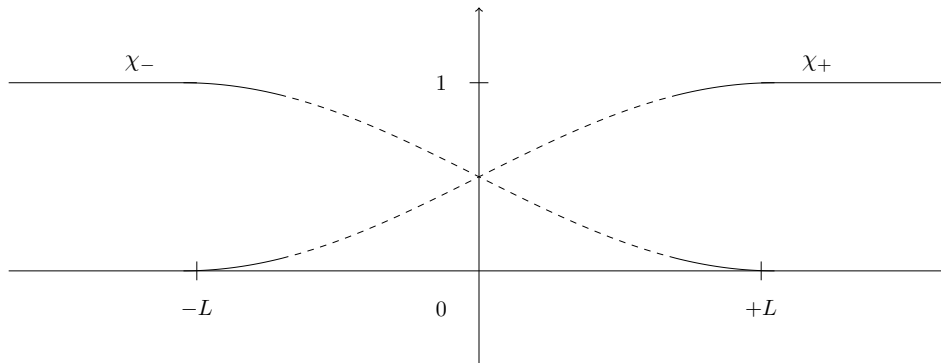


FIGURE 13 – Exemples de fonctions χ_{\pm} .

On va maintenant chercher $u \in H^1_{loc}(\Omega)$ tel que $u - u_i = u - w_+$ soit sortant et vérifiant le problème (7). Ce problème admet une unique solution $u \in H^1_{loc}(\Omega)$. La preuve de ce résultat est effectuée dans le polycopié [1].

En particulier, si u est la solution, on peut écrire $u - w_+ = s_+\chi_+w_+ + s_-\chi_-w_- + \tilde{u}$ avec les conventions précédentes. A présent, nous allons nous intéresser à la valeur du champ $u - w_+$ en $-\infty$ et de celle de u en $+\infty$. Ainsi, il est naturel de calculer $u - \chi_-w_+$:

$$\begin{aligned} u - \chi_-w_+ &= (1 - \chi_-)w_+ + s_+\chi_+w_+ + s_-\chi_-w_- + \tilde{u} \\ &= \underbrace{(1 - \chi_- - \chi_+)}_{= 0 \text{ en dehors de } [-L, L]} w_+ + (1 + s_+)\chi_+w_+ + s_-\chi_-w_- + \tilde{u} \\ &= (1 + s_+)\chi_+w_+ + s_-\chi_-w_- + \hat{u} \quad \text{avec } \hat{u} \text{ exponentiellement décroissant} \end{aligned}$$

On notera classiquement $T = 1 + s_+$ le coefficient de transmission et $R = s_-$ le coefficient de réflexion. En effet, avec la définition de χ_{\pm} , en $+\infty$ on a $u = Tw_+ + \hat{u}$ et en $-\infty$ on a $u = u_i + Rw_- + \hat{u}$.

3.4 Conservation d'énergie

Nous allons à présent trouver une relation liant R et T . Pour cela appliquons la conservation d'énergie : Par formule de Green, on a :

$$\begin{aligned} \operatorname{Im} \left(\int_{\partial\Omega_L} \bar{u} \frac{\partial u}{\partial \mathbf{n}} d\sigma \right) &= \operatorname{Im} \left(\int_{\Omega_L} \bar{u} \Delta u d\sigma + \int_{\Omega_L} \nabla \bar{u} \nabla u d\sigma \right) \\ &= \operatorname{Im} \left(\int_{\Omega_L} (1 + \rho) |u|^2 d\sigma + \int_{\Omega_L} |\nabla u|^2 d\sigma \right) \\ &= 0. \end{aligned} \quad (9)$$

Calculons donc

$$\int_{\partial\mathcal{E}} \bar{u} \frac{\partial u}{\partial \mathbf{n}} d\sigma = \int_{\Sigma_{-L}} \bar{u} \frac{\partial u}{\partial \mathbf{n}} d\sigma + \int_{\Sigma_L} \bar{u} \frac{\partial u}{\partial \mathbf{n}} d\sigma.$$

Plaçons nous sur $[L, +\infty]$. On rappelle que l'on a $u = Tw_+ + \hat{u}$. De plus, on sait que \hat{u} est exponentiellement décroissante donc se décompose selon les modes évanescents exponentiellement décroissants en $+\infty$. Ainsi on peut écrire, pour simplifier les calculs suivants,

$$u = \sum_{n=0}^{+\infty} A_n^+ e^{i\beta_n x} \varphi_n(y). \quad (10)$$

En particulier, on a $A_0^+ = \frac{T}{\sqrt{2k}}$.

On a alors :

$$\frac{\partial u}{\partial \mathbf{n}} = \frac{\partial u}{\partial x} = \sum_{n=0}^{+\infty} i\beta_n A_n^+ e^{i\beta_n x} \varphi_n(y). \quad (11)$$

Ainsi, par orthogonalité des modes on obtient :

$$\begin{aligned} \int_{\Sigma_L} \bar{u} \frac{\partial u}{\partial \mathbf{n}} d\sigma &= \int_{\Sigma_+} \left(\sum_{n=0}^{+\infty} i\beta_n |A_n^+|^2 |\varphi_n(y)|^2 \right) d\sigma \\ &= \sum_{n=0}^{+\infty} i\beta_n |A_n^+|^2. \end{aligned} \quad (12)$$

Puisque β_n est imaginaire pur pour $n > 0$, on a :

$$\operatorname{Im} \left(\int_{\Sigma_L} \bar{u} \frac{\partial u}{\partial \mathbf{n}} d\sigma \right) = \beta_0 |A_0^+|^2 = \frac{|T|^2}{2k}. \quad (13)$$

De la même manière, en $-\infty$, on obtient :

$$\operatorname{Im} \left(\int_{\Sigma_{-L}} \bar{u} \frac{\partial u}{\partial \mathbf{n}} d\sigma \right) = \frac{|R|^2 - 1}{2k}. \quad (14)$$

Ainsi, on peut conclure :

$$|R|^2 + |T|^2 = 1. \quad (15)$$

Cette égalité va être extrêmement importante! En effet, on se rend déjà compte que si nous

réussissons à créer un défaut ne produisant aucune réflexion ($R = 0$), on pourra affirmer que $|T| = 1$, autrement dit, en $+\infty$ on aura $u = Tw_+ + \tilde{u} = e^{i\theta}w_+ + \tilde{u} = \frac{1}{\sqrt{2k}}e^{i(kx+\theta)} + \tilde{u}$ où $\theta \in \mathbb{R}$. Si $\theta \neq 0[2\pi]\mathbb{R}$, on parlera de déphasage entre l'onde incidente et le champ total en $+\infty$. Attention, on n'a pas nécessairement $T = 1$! Cependant, si on réussit à trouver un défaut invisible ($T = 1$), on sait qu'il ne produit aucune réflexion puisque le coefficient de réflexion R est nul.

3.5 Les coefficients de scattering

Notre objectif est maintenant de trouver un défaut invisible, c'est-à-dire de construire un coefficient physique ρ tel que les coefficients s_- et s_+ dans la décomposition (8) vérifient $s_- = 0$ (non réflexion) et $s_+ = 0$ (transmission parfaite). Nous allons commencer par exprimer s_{\pm} en fonction de (u, ρ) :

Calculons, de deux manières différentes, l'expression

$$\int_{\Sigma_{-L} \cup \Sigma_L} \left(\frac{\partial u_s}{\partial \mathbf{n}} \overline{w_+} - u_s \frac{\partial \overline{w_+}}{\partial \mathbf{n}} \right) d\sigma.$$

Comme $\Delta u + k^2(1 + \rho)u = 0$ et $\Delta \overline{w_+} + k^2 \overline{w_+} = 0$, on remarque que $\Delta u_s + k^2 u_s = -k^2 \rho u$. Ainsi, en utilisant la formule de Green sur Ω_L , on obtient :

$$\begin{aligned} \int_{\Sigma_{-L} \cup \Sigma_L} \left(\frac{\partial u_s}{\partial \mathbf{n}} \overline{w_+} - u_s \frac{\partial \overline{w_+}}{\partial \mathbf{n}} \right) d\sigma &= \int_{\Omega_L} \Delta u_s \overline{w_+} - u_s \Delta \overline{w_+} d\Omega \\ &= \int_{\Omega_L} \Delta u_s \overline{w_+} + k^2 u_s \overline{w_+} d\Omega \\ &= -k^2 \int_{\Omega_L} \rho u \overline{w_+} d\Omega. \end{aligned} \quad (16)$$

Nous allons maintenant calculer le terme de l'équation (16) en utilisant l'expression de u_s en $\pm\infty$. Sur Σ_L , on a $u_s = s_+ w_+ + \hat{u}$, et sur Σ_{-L} , $u_s = s_- w_- + \hat{u}$ avec \hat{u} exponentiellement décroissant. Rappelons que les modes sont orthogonaux deux à deux. En particulier, w_{\pm} est orthogonal à \hat{u} . Ainsi on a par exemple, $\int_{\Sigma_{-L}} \frac{\partial \hat{u}}{\partial \mathbf{n}} \overline{w_+} d\sigma = 0$. On obtient donc :

$$\begin{aligned} &\int_{\Sigma_{-L} \cup \Sigma_L} \left(\frac{\partial u_s}{\partial \mathbf{n}} \overline{w_+} - u_s \frac{\partial \overline{w_+}}{\partial \mathbf{n}} \right) d\sigma \\ &= s_- \int_{\Sigma_{-L}} \left(\frac{\partial w_-}{\partial \mathbf{n}} \overline{w_+} - w_- \frac{\partial \overline{w_+}}{\partial \mathbf{n}} \right) d\sigma + s_+ \int_{\Sigma_L} \left(\frac{\partial w_+}{\partial \mathbf{n}} \overline{w_+} - w_+ \frac{\partial \overline{w_+}}{\partial \mathbf{n}} \right) d\sigma \\ &= s_- \int_{\Sigma_{-L}} \underbrace{\left(\frac{\partial w_+}{\partial \mathbf{n}} \overline{w_+} - w_+ \frac{\partial \overline{w_+}}{\partial \mathbf{n}} \right)}_{=0} d\sigma + s_+ \int_{\Sigma_L} \left(\frac{\partial w_+}{\partial \mathbf{n}} \overline{w_+} - w_+ \frac{\partial \overline{w_+}}{\partial \mathbf{n}} \right) d\sigma \\ &= s_+ \int_0^1 \left(\frac{ik}{\sqrt{2k^2}} - \frac{-ik}{\sqrt{2k^2}} \right) dy \quad (\text{car } \frac{\partial}{\partial \mathbf{n}} = + \frac{\partial}{\partial x} \text{ sur } \Sigma_L) \\ &= i s_+. \end{aligned}$$

On peut reprendre les mêmes calculs avec $\overline{w^-}$ et remarquer que $\text{supp}(\rho) \subset \mathcal{O}$ (permettant de réduire le domaine d'intégration). On a finalement :

$$is_{\pm}(\rho) = -k^2 \int_{\Omega_L} \rho u \overline{w_{\pm}} d\Omega = -k^2 \int_{\mathcal{O}} \rho u \overline{w_{\pm}} d\Omega. \quad (17)$$

Cette expression peut, au premier coup d'oeil, sembler linéaire par rapport à ρ . Cependant, ce n'est pas le cas puisque u dépend de ρ .

Nous allons, dans un premier temps, rechercher un défaut ρ permettant d'annuler s_- .

3.6 Un premier déplacement

Pour reprendre l'idée de la démonstration du Théorème 1, cherchons d'abord à construire un "petit" défaut non réfléchissant ($s_- = 0$). Soit $\varepsilon > 0$, nous allons chercher une perturbation ρ d'amplitude ε : Ceci revient à rechercher une perturbation μ telle que $\rho = \varepsilon\mu$. Définissons alors, à ε fixé, $s_{\pm}^{\varepsilon}(\mu) = s_{\pm}(\varepsilon\mu)$. Notons également u_s^{ε} le champ diffracté par cette perturbation.

Reprenons la formule (17). On peut calculer un développement limité de s_-^{ε} , que l'on admettra dans ce rapport. L'approximation, appelée *approximation de Born*, nous dit qu'une petite perturbation de ρ provoque une petite perturbation de u : $u = w_+ + \mathcal{O}(\varepsilon)$.

$$\begin{aligned} is_-^{\varepsilon}(\mu) &= is_-(\varepsilon\mu) = -k^2 \varepsilon \int_{\mathcal{O}} \mu (u_s^{\varepsilon} + w_+) \overline{w^-} d\Omega \\ &= -k^2 \varepsilon \int_{\mathcal{O}} \mu w_+^2 d\Omega + \mathcal{O}(\varepsilon^2). \end{aligned}$$

Définissons $\widetilde{s_-^{\varepsilon}}$ la quantité telle que

$$s_-^{\varepsilon}(\mu) = s_-(\varepsilon\mu) = ik^2 \varepsilon \int_{\mathcal{O}} \mu w_+^2 d\Omega + \varepsilon^2 \widetilde{s_-^{\varepsilon}}(\mu).$$

Nous pouvons remarquer que le terme $ik^2 \int_{\mathcal{O}} \mu w_+^2 d\Omega$ est exactement la différentielle de s_- en 0 appliquée à μ .

En se rappelant de la démarche suivie dans le paragraphe 2, l'objectif est maintenant d'annuler le terme $ik^2 \int_{\mathcal{O}} \mu w_+^2 d\Omega$, ce que nous supposons réalisé pour une fonction μ_0 (il existe toujours une telle fonction puisque le noyau de la différentielle est de dimension infinie). C'est ici que l'on se rend compte que l'on pourra se permettre de choisir μ_0 . La solution attendue sera proche de μ_0 , donc le choix effectué aura son importance sur la nature du résultat. Nous étudierons dans la suite comment déterminer cette fonction. Le terme $\widetilde{s_-^{\varepsilon}}(\mu_0)$ ne sera lui, a priori, pas nul. L'idée alors est d'ajouter $\tau_1\mu_1 + \tau_2\mu_2$ à μ_0 avec μ_1 et μ_2 tels que :

$$ik^2\varepsilon \int_{\mathcal{O}} \mu_1 w_+^2 d\Omega = 1$$

$$ik^2\varepsilon \int_{\mathcal{O}} \mu_2 w_+^2 d\Omega = i.$$

En notant $\mu = \mu_0 + \tau_1\mu_1 + \tau_2\mu_2$ et $\tau = \tau_1 + i\tau_2$ on obtient $is_-^\varepsilon(\tau) = \varepsilon\tau_1 + i\varepsilon\tau_2 + \varepsilon^2\widetilde{s_-^\varepsilon}(\tau)$.

On remarque dans cette expression qu'il y a deux paramètres réels τ_1 et τ_2 qui vont nous permettre d'annuler le reste. En effet, on souhaite annuler un coefficient complexe ($s_-^\varepsilon(\tau)$), il est donc naturel de jouer avec deux paramètres réels (τ_1 et τ_2).

De plus :

$$s_-^\varepsilon(\tau) = 0 \iff \tau = -\varepsilon\widetilde{s_-^\varepsilon}(\tau).$$

τ est alors solution d'un problème de point fixe, qui admet une unique solution si ε est bien choisi. En effet, on admet ici qu'il existe $\varepsilon_0 > 0$ et un voisinage de 0 tels que la fonction $\tau \mapsto -\widetilde{s_-^\varepsilon}(\tau)$ soit une contraction pour $\varepsilon < \varepsilon_0$ et que ce voisinage soit stable par cette dernière. Le théorème du point fixe assure donc l'existence et l'unicité du point fixe dans un voisinage de 0.

3.7 Choix de μ_0 , μ_1 et μ_2

En reprenant les notations du paragraphe 2, nous cherchons à annuler la fonction F telle que $F(\mu) = (\operatorname{Re}(s_-(\mu)) , \operatorname{Im}(s_-(\mu)))$. Comme nous l'avons vu précédemment, il nous faut trouver des éléments du noyau de $dF(0)$, ainsi que des antécédents d'une base de \mathbb{R}^2 . Nous allons noter G sa différentielle appliquée en 0 :

$$G : L^\infty(\mathcal{O}) \rightarrow \mathbb{R}^2$$

$$\mu \mapsto (\operatorname{Re}(ds_-(0)(\mu)) , \operatorname{Im}(ds_-(0)(\mu))).$$

Rappelons que la fonction $\mu \in L^\infty(\mathcal{O})$ recherchée est à valeur réelle, afin d'avoir une perturbation réalisable en pratique. Si on écrit $w_+^2 = \frac{1}{2k}e^{2ikx} = \frac{1}{2k}(\cos(2kx) + i\sin(2kx))$, on peut réécrire G sous la forme :

$$G(\mu) = \left(-\frac{k}{2} \int_{\mathcal{O}} \mu(x, y) \sin(2kx) d\Omega , +\frac{k}{2} \int_{\mathcal{O}} \mu(x, y) \cos(2kx) d\Omega \right).$$

Nous sommes donc ramenés à trouver μ_0 , μ_1 et μ_2 tels que $G(\mu_0) = (0, 0)$, $G(\mu_1) = (1, 0)$ et $G(\mu_2) = (0, 1)$. Il faut alors, pour μ_1 et μ_2 , s'intéresser à la surjectivité de l'opérateur G .

On peut réécrire ces équations :

$$\begin{aligned} \frac{k}{2} \int_{\mathcal{O}} \mu_1(x, y) \sin(2kx) d\Omega = -1 & \quad \text{et} \quad \frac{k}{2} \int_{\mathcal{O}} \mu_1(x, y) \cos(2kx) d\Omega = 0 \\ \frac{k}{2} \int_{\mathcal{O}} \mu_2(x, y) \sin(2kx) d\Omega = 0 & \quad \text{et} \quad \frac{k}{2k} \int_{\mathcal{O}} \mu_2(x, y) \cos(2kx) d\Omega = 1. \end{aligned}$$

On peut alors généraliser le problème en cherchant à montrer que pour tout couple $(a, b) \in \mathbb{R}^2$ il existe une fonction μ définie sur Ω telle que :

$$\int_{\mathcal{O}} \mu(x, y) \cos(2kx) d\Omega = a \quad \text{et} \quad \int_{\mathcal{O}} \mu(x, y) \sin(2kx) d\Omega = b.$$

Au delà de l'existence, il serait très intéressant de pouvoir également expliciter une solution en vue d'un algorithme.

Pour cela une idée consiste à chercher μ sous la forme $\mu(x, y) = A \cos(2kx) + B \sin(2kx)$. Cela nous conduit à considérer le problème matriciel suivant :

$$\begin{pmatrix} \int_{\mathcal{O}} \cos(2kx) \cos(2kx) d\Omega & \int_{\mathcal{O}} \cos(2kx) \sin(2kx) d\Omega \\ \int_{\mathcal{O}} \sin(2kx) \cos(2kx) d\Omega & \int_{\mathcal{O}} \sin(2kx) \sin(2kx) d\Omega \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}.$$

On reconnaît une matrice de Gram : $\begin{pmatrix} \langle x_1, x_1 \rangle & \langle x_1, x_2 \rangle \\ \langle x_2, x_1 \rangle & \langle x_2, x_2 \rangle \end{pmatrix}$ qui est inversible puisque la famille (\cos, \sin) est libre (lorsque \mathcal{O} est non vide). Ainsi on peut résoudre chacun des 2 systèmes et obtenir les fonctions μ_1 et μ_2 recherchées.

Il faut maintenant trouver un élément du noyau de la différentielle. Pour cela il suffit de prendre n'importe quel élément μ et de lui retrancher ses projections selon (μ_1, μ_2) .

On remarquera que l'on a fait le choix de chercher μ comme combinaison de sin et cos, mais ce n'est pas la seule possibilité. On verra lors de l'itération de cette méthode (cf §3.9), et surtout lors de la programmation, que nous serons amenés à trouver/utiliser d'autres méthodes pour construire μ_1 et μ_2 . En particulier, nous allons chercher à construire nos fonctions constantes par morceaux. Cela semble plus difficile à réaliser, mais nous verrons que cette conditions n'est pas beaucoup plus difficile à satisfaire.

Remarque 2 *Insistons sur l'importance de la surjectivité de G : en effet, en cherchant à résoudre $s_- = 0$ et $s_+ = 0$, on considère les fonctions*

$$F(\mu) = (\operatorname{Re}(s_-(\mu)) , \operatorname{Im}(s_-(\mu)) , \operatorname{Re}(s_+(\mu)) , \operatorname{Im}(s_+(\mu))).$$

$$G(\mu) = (\operatorname{Re}(ds_-(0)(\mu)) , \operatorname{Im}(ds_-(0)(\mu)) , \operatorname{Re}(ds_+(0)(\mu)) , \operatorname{Im}(ds_+(0)(\mu))).$$

$$\text{Mais } ds_+^e(0)(\mu) = ik^2 \varepsilon \int_{\mathcal{O}} \mu w_+ \overline{w_+} d\Omega = ik^2 \varepsilon \int_{\mathcal{O}} \mu d\Omega \in i\mathbb{R}.$$

Donc la troisième composante de G est nulle. Ainsi G ne peut pas être surjective dans \mathbb{R}^4 . Ceci pose donc des problèmes pour appliquer directement la même méthode pour l'invisibilité

totale. Nous verrons plus tard comment contourner cette difficulté.

3.8 Résolution par méthode du point fixe

Comme on vient de le voir, il serait intéressant de trouver ε pour lequel on peut trouver et construire une solution à l'équation de point fixe $\tau = -\varepsilon \widetilde{s}_-^\varepsilon(\tau)$.

On peut aussi montrer que $-\varepsilon \widetilde{s}_-^\varepsilon(\cdot)$ est une fonction contractante, pour peu que l'on choisisse ε assez petit. C'est ainsi que l'on peut appliquer l'algorithme de point fixe du théorème de point fixe et obtenir une perturbation non réfléchissante.

Ce résultat n'est pas démontré ici, mais il peut l'être en utilisant des raisonnements analogues à la démonstration du théorème des fonctions implicites.

Une question mérite d'être posée à ce stade de l'étude : la perturbation ainsi trouvée peut-elle être la perturbation nulle ? Autrement dit, est-on certain d'avoir construit une perturbation invisible non triviale ? Si nous reprenons l'écriture $\rho = \varepsilon(\mu_0 + \tau_1 \mu_1 + \tau_2 \mu_2)$, on remarque que ρ est une combinaison linéaire non triviale ($\varepsilon \neq 0$) de trois fonctions, par construction, linéairement indépendantes. En effet, μ_0 appartient au noyau de G , et μ_1 (resp. μ_2) est d'image $(0, 1)$ (resp. $(1, 0)$).

Nous avons ainsi réussi à créer une perturbation non triviale. Cette perturbation est, a priori, de taille ε petit ; nous allons donc essayer d'itérer la méthode en partant de la perturbation précédemment obtenue afin de grandir le défaut.

3.9 Itération de la méthode

L'idée est donc d'essayer d'appliquer la méthode de continuation présentée au paragraphe 2. Pour cela il va falloir dans un premier temps calculer la différentielle de s_- en n'importe quel point ρ , puis étudier sa surjectivité.

Pour bien comprendre et ne pas se perdre dans les notations, à ρ donné la solution ne sera plus notée u mais u_ρ . Rappelons la formule utile :

$$is_\pm(\rho) = -k^2 \int_{\mathcal{O}} \rho u_\rho \overline{w_\pm} d\Omega.$$

Donc,

$$s_-(\rho) = ik^2 \int_{\mathcal{O}} \rho u_\rho w_+ d\Omega.$$

On admet ensuite que lorsque l'on modifie très peu une perturbation ρ , la solution est aussi très peu modifiée : $u_{\rho+\varepsilon\mu} = u_\rho + \mathcal{O}(\varepsilon)$ (approximation de Born)

Notons ρ_0 une perturbation pour laquelle $s_-(\rho_0) = 0$. On cherche à calculer :

$$\begin{aligned} ds_-(\rho_0)(\mu) &= \lim_{\varepsilon \rightarrow 0} \frac{s_-(\rho_0 + \varepsilon\mu) - s_-(\rho_0)}{\varepsilon} \\ &= \lim_{\varepsilon \rightarrow 0} \frac{ik^2}{\varepsilon} \int_{\mathcal{O}} ((\rho_0 + \varepsilon\mu) \underbrace{u_{\rho_0 + \varepsilon\mu}}_{=u_{\rho_0} + \mathcal{O}(\varepsilon)} - \rho_0 u_{\rho_0}) w_+ d\Omega \\ &= \lim_{\varepsilon \rightarrow 0} \frac{ik^2}{\varepsilon} \int_{\mathcal{O}} (\varepsilon \mu u_{\rho_0} w_+ + \mathcal{O}(\varepsilon^2)) d\Omega \\ &= ik^2 \int_{\mathcal{O}} \mu u_{\rho_0} w_+ d\Omega. \end{aligned}$$

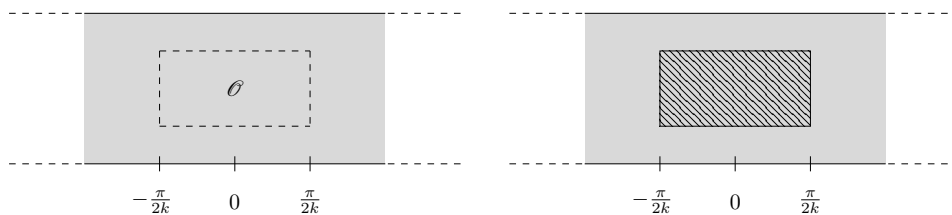
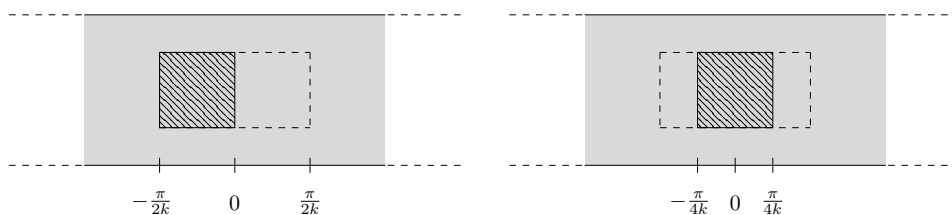
Comme lors de l'initialisation nous aimerions prouver que cette différentielle (d'un espace de dimension infinie dans \mathbb{R}^2) est surjective, et trouver des antécédents de $(1, 0)$ et de $(0, 1)$. On peut imaginer qu'il existe des perturbations ρ qui ne donnent pas de différentielle surjective. Cependant nous allons tout de même, à chaque itération, essayer de trouver des antécédents. Il faudra faire attention aux problèmes éventuels de matrices informatiquement surjectives mais qui ne le sont pas réellement.

L'objectif de cette étude est en premier lieu de construire des défauts non réfléchissants, c'est pourquoi il faut imposer des propriétés sur les perturbations. Lors de l'initialisation il nous avons construit des fonctions qui, multipliées par sin ou cos, sont d'intégrale égale à 0 ou 1. En utilisant des critères de parité et d'imparité de ces dernières il est assez aisé de construire les fonctions μ_0 , μ_1 et μ_2 .

En effet, on souhaite que $\int_{\mathcal{O}} \mu_0(x, y) \sin(2kx) d\Omega = 0$ et $\int_{\mathcal{O}} \mu_0(x, y) \cos(2kx) d\Omega = 0$ donc en prenant μ_0 constante sur une période entière de $x \mapsto \sin(2kx)$, on annulera ces deux intégrales. Maintenant on voudrait n'en annuler qu'une seule pour obtenir μ_1 et μ_2 . Par imparité du sin, en prenant un domaine symétrique (selon la variable x) on annulera l'intégrale avec le sin ; et en remarquant que cos est positive dans un voisinage de 0 on voit que si ce domaine n'est pas trop grand, on pourra ne pas annuler l'intégrale avec le cos. C'est pourquoi prendre μ_2 constante (judicieusement choisie) sur une demi-période centrée de $x \mapsto \sin(2kx)$ est suffisant. On peut effectuer le même raisonnement pour μ_1 en inversant sin et cos.

Concernant la forme du domaine selon la variable y , on a décidé de prendre un domaine rectangulaire centré. Sa position et sa taille selon y ne sont pas importantes pour nos démarches, mais influenceront sur les résultats obtenus et sur la rapidité de convergence de l'algorithme. On résume ce raisonnement par les Figures 14 et 15 :

Il est très important de remarquer qu'ici nous avons eu à faire un choix, mais que bien évidemment il existe bien d'autres possibilités ! Il s'avère que ce cas est aussi très simple à coder et nous permettra de comprendre évolue notre perturbation.


 FIGURE 14 – Domaine de perturbation et domaine de définition de μ_0 .

 FIGURE 15 – Domaines de définition de μ_1 et de μ_2 .

L'idée qui a émergé de cela est de chercher, à chaque itération, les μ_1^n et μ_2^n comme combinaison linéaire de μ_1 et μ_2 . Pour μ_0^n , puisqu'elle est dans le noyau de la différentielle, il nous suffit de choisir un vecteur et de lui retrancher ses projections selon μ_1^n et μ_2^n . Écrivons le calcul de détermination de μ_1^n et μ_2^n :

Définissons a_1^n , a_2^n , b_1^n et b_2^n tels que

$$\begin{cases} \mu_1^n = a_1^n \mu_1 + a_2^n \mu_2 \\ \mu_2^n = a_2^n \mu_1 + a_1^n \mu_2. \end{cases}$$

Comme μ_1^n et μ_2^n doivent vérifier

$$\begin{cases} ik^2 \int_{\mathcal{O}} \mu_1^n u_{\rho^n} w_+ d\Omega = 1 \\ ik^2 \int_{\mathcal{O}} \mu_2^n u_{\rho^n} w_+ d\Omega = i, \end{cases}$$

alors

$$\begin{cases} (ik^2 \int_{\mathcal{O}} \mu_1 u_{\rho^n} w_+ d\Omega) a_1^n + (ik^2 \int_{\mathcal{O}} \mu_2 u_{\rho^n} w_+ d\Omega) a_2^n = 1 \\ (ik^2 \int_{\mathcal{O}} \mu_1 u_{\rho^n} w_+ d\Omega) b_1^n + (ik^2 \int_{\mathcal{O}} \mu_2 u_{\rho^n} w_+ d\Omega) b_2^n = i. \end{cases}$$

En notant

$$M^n = \begin{pmatrix} -k^2 \int_{\mathcal{O}} \mu_1 u_{\rho^n} \sin(kx) d\Omega & -k^2 \int_{\mathcal{O}} \mu_2 u_{\rho^n} \sin(kx) d\Omega \\ +k^2 \int_{\mathcal{O}} \mu_1 u_{\rho^n} \cos(kx) d\Omega & +k^2 \int_{\mathcal{O}} \mu_2 u_{\rho^n} \cos(kx) d\Omega \end{pmatrix},$$

le système revient à résoudre les deux problèmes suivant, et donc à inverser M^n :

$$M^n \cdot \begin{pmatrix} a_1^n \\ a_2^n \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{et} \quad M^n \cdot \begin{pmatrix} b_1^n \\ b_2^n \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Nous ne prouvons pas ici que ces matrices sont toujours inversibles, puisqu'il est possible que ça ne soit pas le cas. Tout comme la surjectivité, il nous faudra donc vérifier numériquement que les différentes matrices sont non-dégénérées ou inversibles en calculant des déterminants adéquats.

3.10 Invisibilité totale

Comme on l'a rapidement expliqué précédemment, lorsqu'on écrit notre fonction G dans le cas de l'invisibilité totale (non réflexion et transmission parfaite) on se rend compte qu'elle ne peut pas être surjective. Cela semble être un problème, mais en se rappelant que la conservation d'énergie nous donne $|R|^2 + |T|^2 = 1$, on a une quatrième équation et l'on peut espérer pouvoir conclure.

En effet, considérons les fonctions F et G à valeur dans \mathbb{R}^3 comme

$$F(\mu) = (\operatorname{Re}(s_-(\mu)) , \operatorname{Im}(s_-(\mu)) , \operatorname{Im}(s_+(\mu))).$$

et

$$G(\mu) = (\operatorname{Re}(ds_-(0)(\mu)) , \operatorname{Im}(ds_-(0)(\mu)) , \operatorname{Im}(ds_+(0)(\mu))).$$

Cette fonction est surjective, et nous permet de trouver μ_0, μ_1, μ_2 et μ_3 étant respectivement un élément du noyau, des antécédents de $(1, 0, 0)$, $(0, 1, 0)$ et $(0, 0, 1)$. Ainsi par la même méthode que précédemment (algorithme de point fixe), on peut trouver une perturbation ρ annulant $s_-^\varepsilon(\rho)$ et $\operatorname{Im}(s_+^\varepsilon(\rho))$.

Maintenant, si l'on réécrit la conservation d'énergie, on a $|s_-^\varepsilon(\rho)|^2 + |1 + s_+^\varepsilon(\rho)|^2 = 1$. Comme $s_-^\varepsilon(\rho) = 0$ et $\operatorname{Im}(s_+^\varepsilon(\rho)) = 0$, on peut écrire $|1 + s_+^\varepsilon(\rho)|^2 = 1$, puis que $(1 + \operatorname{Re}(s_+^\varepsilon(\rho)))^2 + (\operatorname{Im}(s_+^\varepsilon(\rho)))^2 = 1$. Ainsi $1 + \operatorname{Re}(s_+^\varepsilon(\rho)) = \pm 1$ et finalement $\operatorname{Re}(s_+^\varepsilon(\rho)) = 0$ ou 2 . Pour terminer, on remarque que la méthode choisie consistait à faire un développement limité. Ainsi on a $s_\pm^\varepsilon(\rho) = \mathcal{O}(\varepsilon^2)$, donc en choisissant ε assez petit on aura réussi à construire une perturbation ρ telle que $s_\pm^\varepsilon(\rho) = 0$, c'est à dire une perturbation parfaitement invisible !

4 Programmation de l'algorithme

Après avoir étudié mathématiquement l'algorithme nécessaire à la résolution de notre problème, passons à la réalisation pratique du code à l'aide du logiciel *Xlife++*.

Pour ce faire nous allons d'abord écrire le schéma de l'algorithme, avant de se lancer dans l'analyse des résultats.

4.1 Écriture de l'algorithme

Suite à l'étude mathématique effectuée précédemment, le principe général nous donne (en notant (ρ_k) la famille des perturbations successives) :

Algorithme général :

- Initialisation :
 1. Construire la fonction $\rho_0 = 0$
- Tant que le point fixe converge, pour tout $k \geq 0$:
 1. Trouver la solution u_{ρ_k} associée
 2. Algorithme de point fixe afin de trouver ρ_{k+1}

Cet algorithme semble assez simple écrit comme cela. En effet, il ne demande que deux boîtes noires permettant de calculer u_ρ à partir de ρ , et l'algorithme de point fixe. La première étape utilise une résolution éléments finis, qui est une utilisation classique de la librairie *Xlife++*. La deuxième demande à être un peu détaillée, au moins dans la formulation de l'algorithme :

Algorithme du point fixe :

- Initialisation :
 1. Fixer $\varepsilon > 0$ et $N > 0$
 2. Construire μ_0, μ_1 et μ_2
 3. Poser $t = t_1 + i t_2 = 0$ et $n = 0$
 4. Construire $\rho = \varepsilon(\mu_0 + t_1\mu_1 + t_2\mu_2)$
- Tant que $|s_-(\rho)| > 10^{-6}$ et $n < N$:
 1. $n \leftarrow n + 1$
 2. $t = t_1 + i t_2 \leftarrow t - \frac{s_-(\rho)}{\varepsilon}$
 3. $\rho \leftarrow \varepsilon(\mu_0 + t_1\mu_1 + t_2\mu_2)$
 4. Calculer $s_-(\rho)$

On pourra remarquer que cet algorithme de point fixe est écrit à ε fixé. Dans la pratique, dans l'algorithme général on prendra par exemple $\varepsilon = 0.1$; si le point fixe ne converge pas on recommencera avec $\varepsilon = 0.01$ par exemple. En effet, on sait que le point fixe converge pour ε assez petit, il serait dommage de s'arrêter à cause d'un ε non idéalement choisi. Bien entendu, on ne va pas ε prendre trop petit, cela n'aurait que peu d'intérêt d'avancer avec un pas de 10^{-5} .

4.2 *Xlife++*

Xlife++ est une librairie, développée en C++, permettant entre autres de résoudre des problèmes par des méthodes éléments finis. *Xlife++* est développé au laboratoire POEMS de l'ENSTA, c'est pourquoi j'ai été amené à l'utiliser dans le cadre de mon stage.

Pour travailler avec *Xlife++* il est nécessaire de connaître les bases du C++. C'est une librairie simple d'utilisation : par exemple, pour résoudre le problème sans défaut il m'a fallu définir le domaine, le maillage associé, l'espace fonctionnel correspondant et l'équation.

4.3 Non réflexion

Avant de coder la totalité du code, il est intéressant de programmer une première itération, correspondant à un algorithme de point fixe. Ce bout de code sera ensuite généralisé pour donner la méthode de continuation générale.

Pour calculer numériquement une approximation de la solution des problèmes de diffraction, il est nécessaire de borner le domaine de résolution, d'écrire la formulation variationnelle du problème et d'en trouver une équivalent dans le domaine borné. On se restreindra au domaine Ω_L , mais il nous faut donner des conditions aux bords afin que le nouveau problème soit équivalent au premier. Pour cela nous pouvons, par exemple, utiliser la méthode de DtN (Dirichlet-to-Neumann) qui est facilement implémentable avec *Xlife++*.

Nous allons définir les opérateurs T_L et T_{-L} :

$$T_L : H^{\frac{1}{2}}(\Sigma_L) \rightarrow H^{-\frac{1}{2}}(\Sigma_L)$$

$$u \mapsto \sum_{n=0}^{+\infty} i\beta_n(u|\varphi_n)_{L^2(\Sigma_L)}\varphi_n \quad (18)$$

$$T_{-L} : H^{\frac{1}{2}}(\Sigma_{-L}) \rightarrow H^{-\frac{1}{2}}(\Sigma_{-L})$$

$$u \mapsto \sum_{n=0}^{+\infty} i\beta_n(u|\varphi_n)_{L^2(\Sigma_{-L})}\varphi_n. \quad (19)$$

Nous admettons ici que le Problème (7) est équivalent au Problème (20) suivant, avec $u = u_i + u_s$:

$$\begin{cases} \Delta u + k^2(1 + \rho)u = 0 & \text{dans } \Omega_L \\ \partial_{\mathbf{n}}u = 0 & \text{sur } \partial\Omega_L \setminus (\Sigma_{-L} \cup \Sigma_L) \\ \partial_{\mathbf{n}}u_s = T_{-L}u_s & \text{sur } \Sigma_{-L} \\ \partial_{\mathbf{n}}u_s = T_Lu_s & \text{sur } \Sigma_L. \end{cases} \quad (20)$$

De plus, les opérateurs définis précédemment mettent en jeu une somme infinie de termes. En pratique, nous allons tronquer ces sommes. Définissons les opérateurs $T_{L,N}$ et $T_{-L,N}$ comme en (18) et (19) en stoppant la série à $n = N \in \mathbb{N}$:

$$\begin{cases} \Delta u + k^2(1 + \rho)u = 0 & \text{dans } \Omega_L \\ \partial_{\mathbf{n}}u = 0 & \text{sur } \partial\Omega_L \setminus (\Sigma_{-L} \cup \Sigma_L) \\ \partial_{\mathbf{n}}u_s = T_{-L,N}u_s & \text{sur } \Sigma_{-L} \\ \partial_{\mathbf{n}}u_s = T_{L,N}u_s & \text{sur } \Sigma_L. \end{cases} \quad (21)$$

On peut montrer que la solution u_N de ce problème converge exponentiellement dans $H^1(\Omega_L)$ vers la solution u du Problème (20) lorsque N tend vers $+\infty$. On pourra trouver la démonstration de ce résultat dans le polycopié [1]. Pour les expériences numériques, nous prendrons $N = 10$.

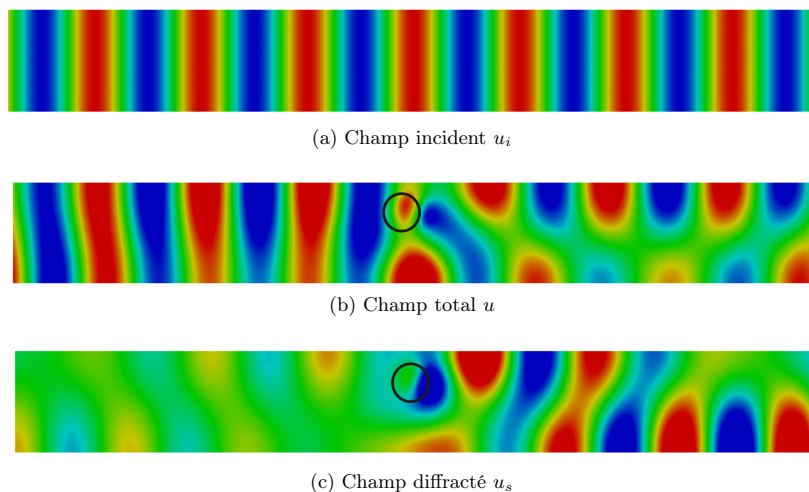
Afin de pouvoir résoudre le Problème (21), il nous faut transformer les conditions de Dirichlet-to-Neumann sur u_s en conditions sur u . Sur $\Sigma_{\pm L}$, on a $u_s = u - u_i$, donc $\partial_{\mathbf{n}}u = \partial_{\mathbf{n}}u_i + T_{\pm L,N}u - T_{\pm L,N}u_i$. Comme $u_i = \frac{1}{\sqrt{2k}}\varphi_0(y)e^{ikx}$, on a $T_{-L,N}u_i = i\beta_0\varphi_0(y)e^{ikx}$. Ainsi, sur Σ_{-L} , par orthogonalité des modes (φ_n) et puisque la normale est dirigée selon $-x$, on a $T_{-L,N}u_i = -\partial_{\mathbf{n}}u_i$ et donc $\partial_{\mathbf{n}}u = 2\partial_{\mathbf{n}}u_i + T_{-L,N}u$. Et, sur Σ_L , on a $T_{-L,N}u_i = \partial_{\mathbf{n}}u_i$ donc $\partial_{\mathbf{n}}u = T_{-L,N}u$. Le Problème (21) est donc équivalent au Problème (22) :

$$\begin{cases} \Delta u + k^2(1 + \rho)u = 0 & \text{dans } \Omega_L \\ \partial_{\mathbf{n}}u = 0 & \text{sur } \partial\Omega_L \setminus (\Sigma_{-L} \cup \Sigma_L) \\ \partial_{\mathbf{n}}u = 2\partial_{\mathbf{n}}u_i + T_{-L,N}u & \text{sur } \Sigma_{-L} \\ \partial_{\mathbf{n}}u = T_{L,N}u & \text{sur } \Sigma_L. \end{cases} \quad (22)$$

Prenons par exemple, le cas où le support de la perturbation serait un disque non centré (on prend ρ constante sur le disque) ; avec cette formulation, nous pouvons calculer puis tracer sous *Paraview* le champ incident u_i , le champ total u et le champ diffracté $u_s = u - u_i$. La Figure 16 représente ces résultats.

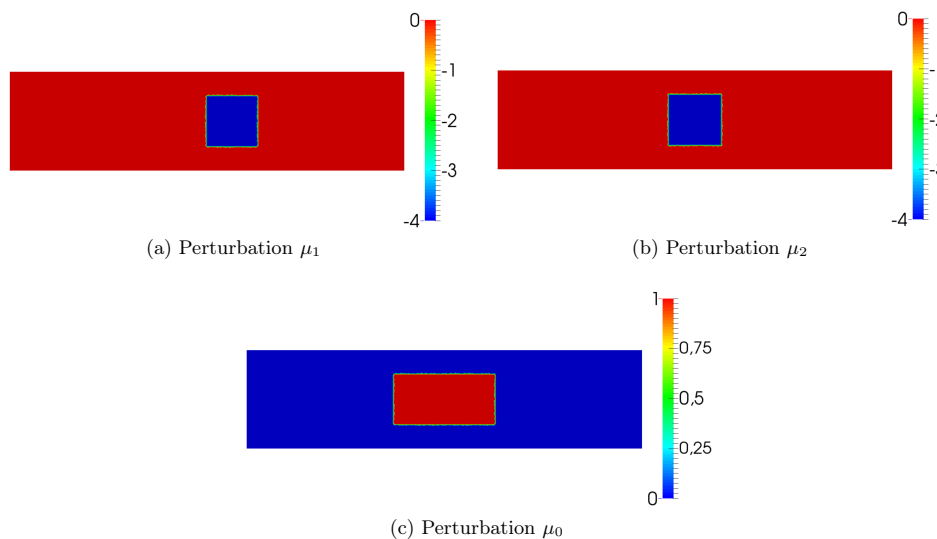
4.3.1 Initialisation de l'algorithme

Dans la suite, nous prendrons $L = 4$.

FIGURE 16 – Diffraction d'une onde sur un défaut - $L = 4$ et $k = 3$.

Nous avons, en premier lieu, défini les fonctions μ_0 , μ_1 et μ_2 afin de créer notre algorithme du point fixe. Dans l'algorithme, les fonctions μ_0 , μ_1 et μ_2 changeront à chaque étage.

Les Figures 17c, 17a et 17b représentent ces trois fonctions à l'itération 0.

FIGURE 17 – (a) μ_1 , (b) μ_2 et (c) μ_0 .

Une fois l'algorithme du point fixe programmé, nous avons pu l'utiliser afin d'obtenir un premier défaut, non trivial, ne produisant aucune réflexion. Dans cette simulation, nous rappelons que nous avons fixé $L = 4$ et $k = 3$. De plus, nous avons choisi $\varepsilon = 0.6$. Le point fixe a convergé en 42 itérations, permettant d'obtenir les résultats présentés en Figure 18.

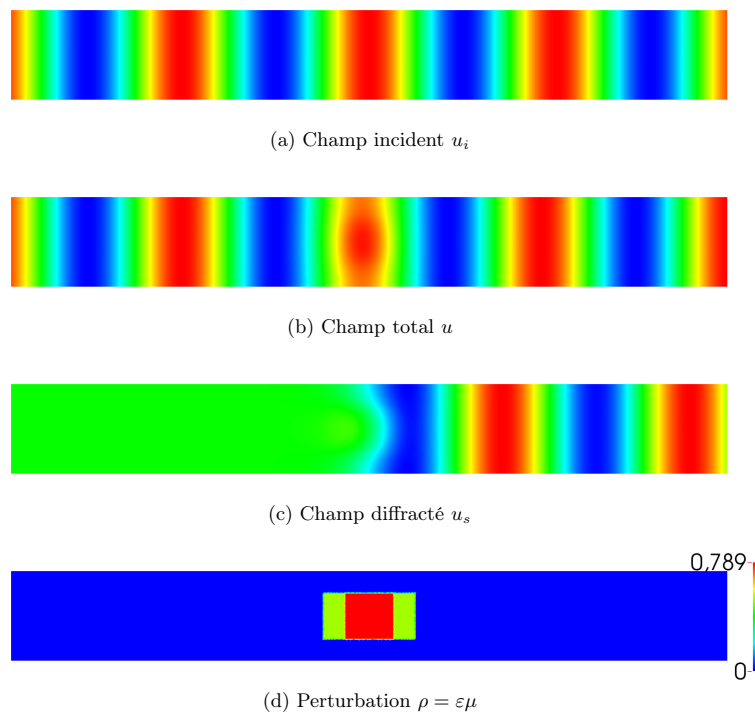


FIGURE 18 – Diffraction d'une onde sur un défaut - $L = 4$, $k = 3$, $\varepsilon = 0.6$.

On peut d'abord observer que le défaut obtenu est loin d'être négligeable. En effet il présente un maximum à 0.789 ce qui fait passer l'indice du milieu de 1 à 1.789. La contrainte géométrique imposant à la perturbation d'être constante sur chacune des quatre briques a été respectée.

Intéressons nous, maintenant, aux différents champs obtenus (Figure 18). Précisons d'abord que l'échelle n'est pas la même pour u_s et pour u , u_i . On remarque sur le champ diffracté que la réflexion est bien nulle ; ce qui signifie que le défaut est non réfléchissant. Concernant la transmission, on peut visualiser un déphasage du champ total par rapport au champ incident (à droite de l'obstacle). Ce déphasage peut également être observé sur le champ diffracté.

Ce résultat est très satisfaisant puisqu'il laisse penser que l'algorithme du point fixe nous permettra de poursuivre notre méthode, et de trouver de nouveaux défauts non réfléchissants.

De plus, on remarque que l'on a réussi à construire le défaut comme on le souhaitait : ici on le cherchait constant par briques rectangulaires, mais on peut espérer obtenir des résultats similaires avec d'autres formes.

Avant de continuer, nous avons souhaité regarder l'évolution de la valeur de $|R|$ (qui est notre critère d'arrêt) lors du parcours de l'algorithme du point fixe, en fonction de la valeur de ε . D'après la théorie présentée dans la partie 3, on s'imagine que le point fixe convergera plus rapidement lorsque ε est petit. (On se rappelle que l'on a effectué un développement limité, donc que plus ε est petit, moins on s'est "éloigné" du lieu des zéros).

La Figure 19 confirme l'intuition précédente. On observe qu'il est de plus en plus difficile de converger lorsque ε grandit. Informatiquement, l'algorithme diverge pour ε supérieur à 1.

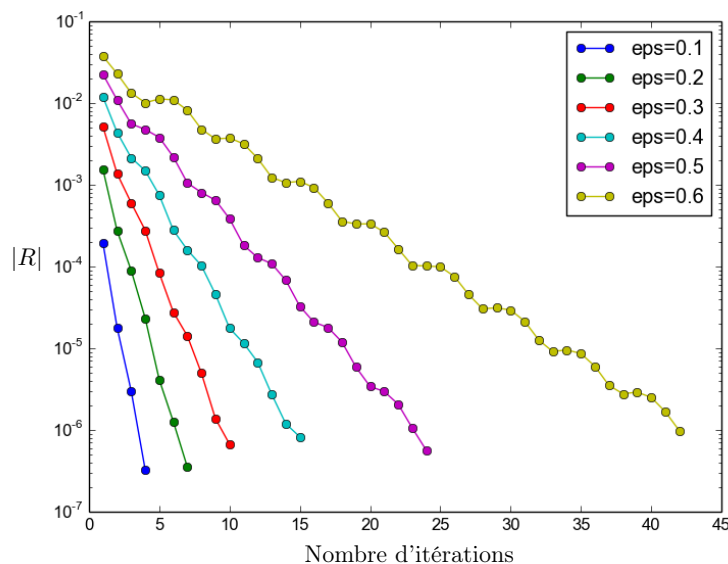


FIGURE 19 – Variation de la valeur de $|R|$ durant l'algorithme de point fixe - $L = 4$, $k = 3$.

Nous avons vu que la largeur de la brique n'est pas importante dans la construction de défauts invisibles, mais a-t-elle une importance dans la rapidité de convergence de l'algorithme de point fixe? Pour étudier cette question nous avons, à ε fixé, fait varier l'épaisseur de la brique et mesuré le nombre d'itération nécessaire afin de faire converger l'algorithme de point fixe. Comme le montre la Figure 20, on remarque qu'il est plus facile de construire un défaut non réfléchissant de faible épaisseur. Il semble en effet assez naturel qu'il soit plus facile de rendre invisible un objet plus fin. Mais on remarque également qu'à partir d'une certaine épaisseur (environ $\varepsilon = 0.7$), le nombre d'itération nécessaire diminue. On peut imaginer que ce phénomène vient du fait que

lorsque le défaut remplit totalement l'épaisseur du guide, le problème se ramène à un problème en unidimensionnelle.

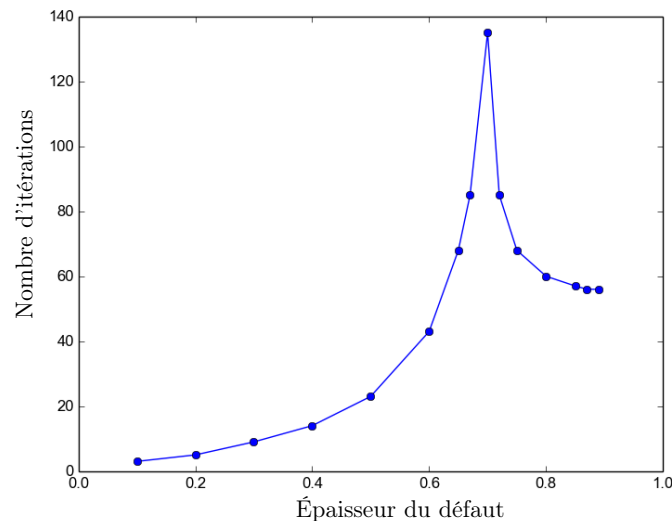


FIGURE 20 – Variation de l'épaisseur du défaut - $L = 4$, $k = 3$, $\varepsilon = 0.1$.

4.3.2 Itération de l'algorithme

A présent, nous allons partir de ce défaut que l'on appellera ρ_1 et construire des nouveaux points d'annulations de la fonction R en partant de ρ_1 . Pour cela il nous faut construire les éléments μ_0^1 , μ_1^1 et μ_2^1 en se servant des calculs effectués dans le paragraphe 3.9, pour ensuite appliquer l'algorithme de point fixe écrit précédemment et trouver ρ_2 . Plus généralement nous allons construire μ_0^n , μ_1^n et μ_2^n afin d'obtenir ρ_n .

Dans le paragraphe 3.9, nous avons décidé d'exprimer μ_1^n et μ_2^n comme combinaison linéaire de μ_1 et de μ_2 . Aussi, μ_0^n est construite en choisissant une fonction et en lui retranchant ses projections selon μ_1^n et μ_2^n (cette fonction sera prise comme combinaison linéaire de μ_0 , μ_1 et μ_2). Chaque fonction ρ_n s'exprime donc en fonction de μ_0 , μ_1 et μ_2 , c'est pourquoi nous pouvons tracer, dans la base (μ_0, μ_1, μ_2) , l'évolution des défauts ρ_n avec $n \in \mathbb{N}$. Le choix de ε à chaque étape est important, nous avons donc décidé de lancer l'algorithme de point fixe avec ε valant successivement les valeurs 0.1, 0.05, 0.01 et 0.001. Si il y a convergence pour $\varepsilon = 1$, par exemple, on ne testera pas les suivants. Nous représentons les 52 premières itérations sur la Figure 21. Au delà, l'algorithme de point fixe ne permet plus de converger pour $\varepsilon = 0.001$. Cet arrêt de la courbe est pour l'instant inexplicé, mais une étude de la différentielle pourrait nous aider à en

comprendre les raisons.

On remarquera, sur la figure 21, que l'espace entre les différents points est plus ou moins grand, selon la valeur de ε permettant la convergence de l'algorithme du point fixe. On pourra également prêter attention aux valeurs des différentes coordonnées. La coordonnée selon μ_1 est, sauf au début de la courbe, négligeable devant les autres composantes. Comme μ_0 et μ_2 sont des fonctions symétriques par rapport à l'axe (Oy) , on s'attend à obtenir un défaut également symétrique.

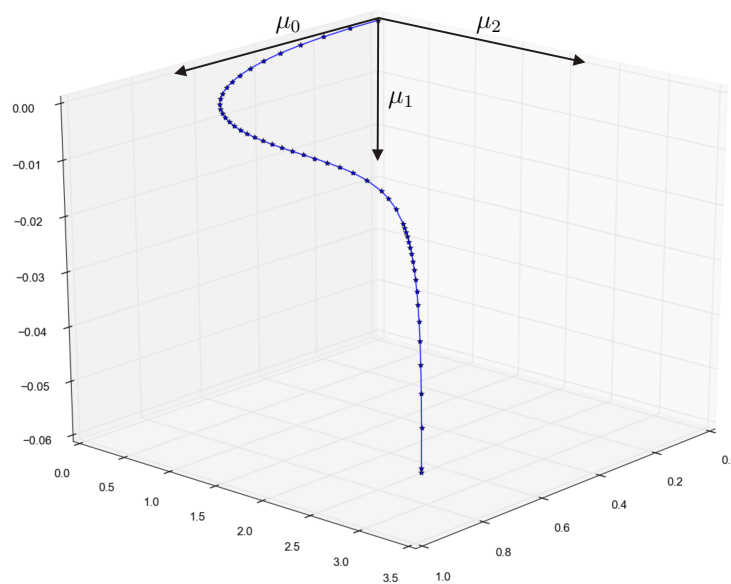
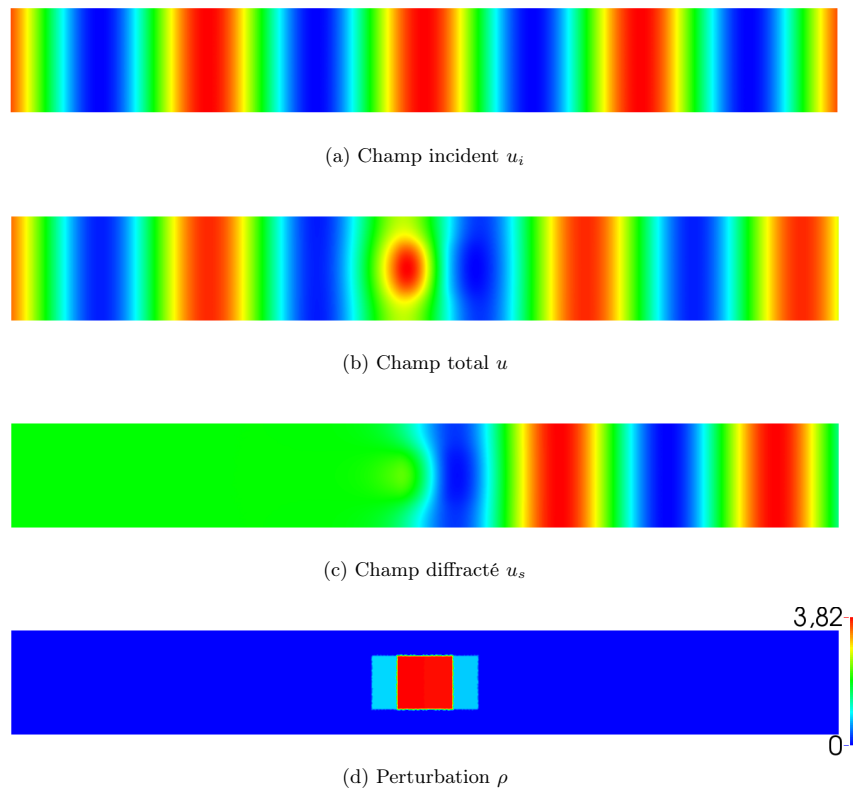


FIGURE 21 – Déplacement le long de la courbe de non réflexion.

Nous avons tracé en Figure 22 les résultats obtenus à la dernière itération. Nous pouvons y voir, à nouveau, le champ incident u_i , le champ total u , le champ diffracté u_s ainsi que le défaut ρ . Comme attendu précédemment, ce défaut semble symétrique en x . De plus, nous pouvons toujours constater en comparant u et u_i qu'il y a un déphasage entre ces deux champs. Mais aussi, sur le champ diffracté, nous observons que l'obstacle est non réfléchissant.

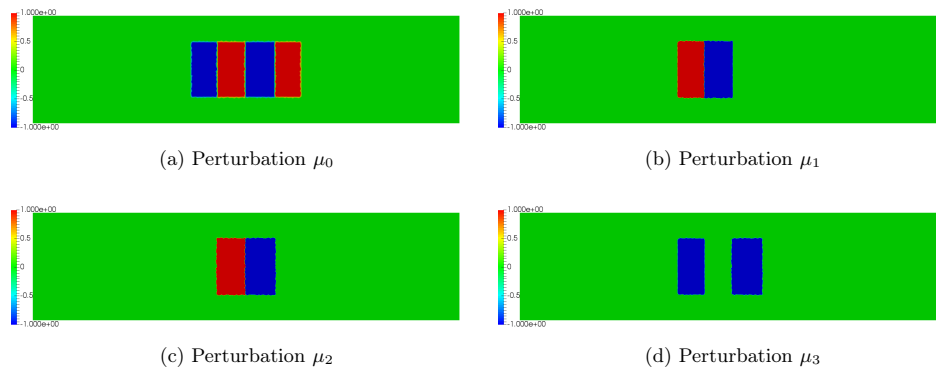
Nous concluons ce paragraphe en remarquant que nous avons réussi à construire un obstacle respectant les contraintes géométriques demandées (ici, dont sa fonction représentative, ρ , est constante par briques) et ayant des valeurs (un maximum à environ 3.82) très loin d'être nulles comparativement à l'indice de référence qui est 1.

FIGURE 22 – Diffraction d'une onde sur un défaut non réfléchissant - $L = 4$, $k = 3$.

4.4 Invisibilité

Passons maintenant à la programmation d'un code permettant de construire un défaut fournissant l'invisibilité totale, à fréquence donnée. Nous prendrons toujours $k = 3$, mais avons effectué les simulations avec $L = 2$. Les fonctions de base μ_0 , μ_1 , μ_2 et μ_3 nécessaires à l'algorithme de point fixe ont besoin d'être modifiées. Un raisonnement analogue au paragraphe 3.9 nous permet de les trouver tels que représentées (à un coefficient multiplicatif près) sur la Figure 23. Notons, que nous cherchons toujours à créer des fonctions ρ constantes par briques (les mêmes quatre briques que précédemment). La seule différence est qu'ici il est nécessaire d'utiliser des perturbations à contraste négatif. Rappelons les équations vérifiées par ces quatre fonctions dans la formule (23).

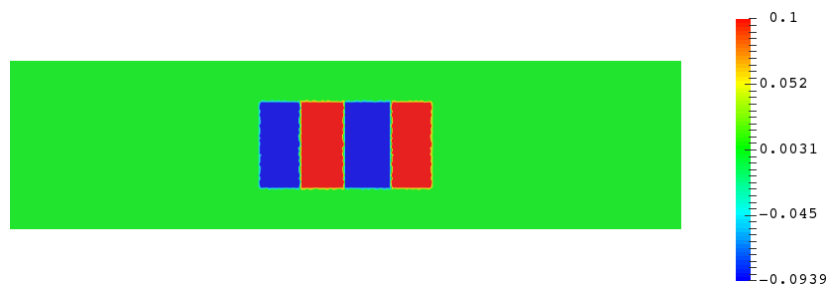
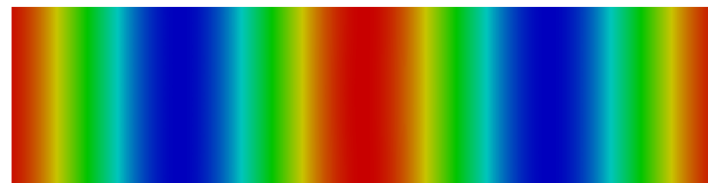
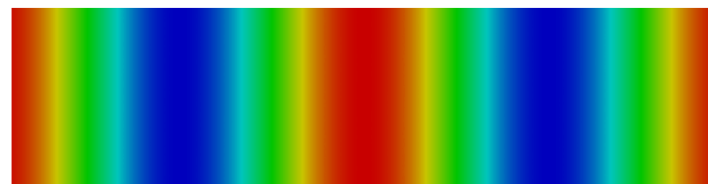
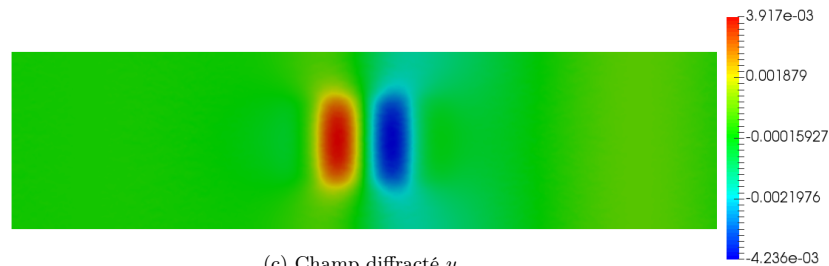
$$\begin{aligned}
 \frac{k^2}{2k} \int_{\mathcal{O}} \mu_0 \sin(2kx) d\Omega &= 0 & \frac{k^2}{2k} \int_{\mathcal{O}} \mu_0 \cos(2kx) d\Omega &= 0 & \frac{k^2}{2k} \int_{\mathcal{O}} \mu_0 d\Omega &= 0 \\
 \frac{k^2}{2k} \int_{\mathcal{O}} \mu_1 \sin(2kx) d\Omega &= -1 & \frac{k^2}{2k} \int_{\mathcal{O}} \mu_1 \cos(2kx) d\Omega &= 0 & \frac{k^2}{2k} \int_{\mathcal{O}} \mu_1 d\Omega &= 0 \\
 \frac{k^2}{2k} \int_{\mathcal{O}} \mu_2 \sin(2kx) d\Omega &= 0 & \frac{k^2}{2k} \int_{\mathcal{O}} \mu_2 \cos(2kx) d\Omega &= 1 & \frac{k^2}{2k} \int_{\mathcal{O}} \mu_2 d\Omega &= 0 \\
 \frac{k^2}{2k} \int_{\mathcal{O}} \mu_3 \sin(2kx) d\Omega &= 0 & \frac{k^2}{2k} \int_{\mathcal{O}} \mu_3 \cos(2kx) d\Omega &= 0 & \frac{k^2}{2k} \int_{\mathcal{O}} \mu_3 d\Omega &= 1.
 \end{aligned} \tag{23}$$


 FIGURE 23 – (c) μ_0 , (a) μ_1 et (b) μ_2 .

Avec ce nouvel algorithme de point fixe, nous n'avons réussi que la première étape. En effet, le programme réalisé n'a pas permis de faire converger le point fixe sur le défaut obtenu à la première étape. Il y a une erreur (remarquée peu avant la fin du stage) dans le programme réalisé qui expliquerait peut-être ce problème. Cette erreur n'a pour l'instant pas été résolue.

Néanmoins nous avons réussi à construire une perturbation totalement invisible à fréquence $k = 3$. Cet obstacle a été obtenu avec $\varepsilon = 0.1$, et on peut remarquer qu'il est très proche de $\varepsilon\mu_0$, comme le laisse imaginer le développement limité.

Le résultat obtenu en Figure 25 est intéressant dans un sens, puisqu'il a permis la construction d'un obstacle invisible. Nous pouvons observer cela avec la champ diffracté qui est nul aux extrémités du guide d'ondes. Cependant, la perturbation créée est relativement faible, et nous aimerions rapidement concrétiser la programmation afin d'obtenir des perturbations d'amplitudes plus importantes.

FIGURE 24 – Obstacle invisible - $L = 2$, $k = 3$.(a) Champ incident u_i (b) Champ total u (c) Champ diffracté u_s FIGURE 25 – Diffraction d'une onde sur un défaut invisible - $L = 2$, $k = 3$.

5 Conclusion et ouvertures

Mes encadrants de stage avaient montré l'existence de perturbations invisibles non triviales, et avaient proposé une méthode numérique permettant d'en construire à partir du guide d'ondes non perturbés. Ces perturbations étant "petites", l'objectif était de reproduire cette démarche afin de s'éloigner encore du guide d'ondes non perturbé.

Dans un premier temps, je me suis approprié leurs études afin de les mettre en application. Le deuxième objectif était de poursuivre l'analyse afin de s'éloigner des "petits" défauts en itérant le procédé présenté dans leurs travaux. Cet objectif a été réalisé en se plaçant à une seule fréquence et en essayant de maîtriser la géométrie de la perturbation.

J'ai alors étudié, dans un cas simplifié, la méthode de continuation que l'on souhaitait mettre en place dans le cas du guide d'ondes. Ensuite, j'ai adapté mathématiquement cette méthode à notre situation, avant de la mettre en place informatiquement.

Les résultats obtenus sont en partie satisfaisants, puisque l'on a réussi à obtenir des obstacles pénétrables ne produisant aucune réflexion (en utilisant le logiciel *Xlife++*). Les obstacles que l'on a construits vérifient bien les contraintes géométriques souhaitées et ne sont pas de petites perturbations. Cependant, des interrogations demeurent, pour l'instant, sans réponse ; notamment concernant l'impossibilité de prolonger la méthode dans le cas de la non réflexion, et concernant la résolution des problèmes existant dans le programme construisant un obstacle invisible.

Plusieurs pistes peuvent être étudiées afin de poursuivre la recherche dans ce domaine. La première idée serait de généraliser la programmation afin de résoudre des problèmes d'invisibilité dans le guide d'ondes, à plusieurs fréquences fixées. Des résultats analogues ont déjà été démontrés sur ce sujet.

Lorsque l'on se donne une perturbation non réfléchissante à fréquence k_0 donnée, nous pouvons représenter la variation de R en fonction de la pulsation ω ($k = \frac{\omega}{c}$). Ainsi, on regarde comment évolue R en fonction de la fréquence. Il peut être intéressant de savoir si R est nul sur une bande de fréquence autour de k_0 ; et donc de chercher à annuler R , mais aussi $\frac{\partial R}{\partial \omega}$, en reprenant une méthode similaire à celle étudiée durant le stage.

Une autre idée peut être de changer totalement de point de vue : ici nous fixons une fréquence et nous cherchons un défaut invisible, mais on pourrait fixer une perturbation et chercher à savoir s'il y a oui ou non des fréquences pour lesquelles la perturbation est invisible, et si oui, comment les calculer.

Références

- [1] A.-S. BONNET BEN-DHIA ET E. LUNÉVILLE, 2015, *Propagation dans les guides d'ondes*.
- [2] A.-S. BONNET-BEN DHIA, L. CHESNEL ET S.A. NAZAROV, 2016, *Complete transmission invisibility for waveguides with sound hard walls*.
- [3] A.-S. BONNET-BEN DHIA ET S.A. NAZAROV, 2013, *Obstacles in acoustic waveguides becoming "invisible" at given frequencies*.
- [4] A.-S. BONNET-BEN DHIA, E. LUNÉVILLE, Y. MBEUTCHA ET S.A. NAZAROV, 2015, *A method to build non-scattering perturbations of two-dimensional acoustic waveguides*.