



Analyse numérique du spectre du Laplacien sur les domaines de surfaces

THÈSE

présentée à la Faculté des Sciences
pour l'obtention du grade de docteur ès sciences par

Fabien Crevoisier

soutenue avec succès le 26 septembre 2012
et acceptée sur proposition du jury

Prof. Olivier Besson	UniNe (directeur de thèse, rapporteur)
Prof. Bruno Colbois	UniNe (directeur de thèse, rapporteur)
Prof. Peter Buser	EPFL (rapporteur)

Université de Neuchâtel, Institut de Mathématiques
Rue Emile-Argant 11, CH-2009 Neuchâtel

IMPRIMATUR POUR THESE DE DOCTORAT

**La Faculté des sciences de l'Université de Neuchâtel
autorise l'impression de la présente thèse soutenue par**

Monsieur Fabien CREVOISIER

**Titre: Analyse numérique du spectre du Laplacien sur les domaines
de surfaces**

sur le rapport des membres du jury:

- Prof. Bruno Colbois, Université de Neuchâtel, co-directeur de thèse
- Prof. Olivier Besson, Université de Neuchâtel, co-directeur de thèse
- Prof. Peter Buser, EPF Lausanne

Neuchâtel, le 15 octobre 2012

Le Doyen, Prof. P. Kropf

RL uf

Résumé

Mots clés : spectre du Laplacien ; domaines de surfaces ; valeurs propres ; fonctions propres ; méthode des éléments finis ; maillage ; ajout d'anses fines ; optimisation de forme ; inégalités isopérimétriques.

Keywords : Laplacian spectrum ; surfaces domains ; eigenvalues ; eigenfunctions ; finite elements method ; meshing ; thin handles adjunction ; shape optimization ; isoperimetric inequalities.

Cette thèse, dont les programmes joints constituent un élément indissociable, traite du spectre du Laplacien sur des domaines de surfaces et plus précisément du calcul numérique des valeurs et fonctions propres pour des conditions au bord de Dirichlet et de Neumann.

Une fois la formulation variationnelle du spectre du Laplacien établie, nous adaptons aux domaines de surfaces la bien connue méthode des éléments finis pour les domaines du plan. La convergence des solutions est démontrée et la mise en œuvre numérique de la méthode est précisée. On aborde en particulier les questions de maillage (par macro-éléments ou de Delaunay) et de traitements de triangulations (renumérotation des nœuds et raffinement local). La construction et la résolution numérique du problème matriciel associé sont également détaillées ; sont notamment abordés la question du mass-lumping et l'algorithme itératif de Lanczos.

Deux applications aux domaines du plan (mais éventuellement généralisables aux surfaces) sont ensuite présentées : premièrement, la démonstration de la convergence des valeurs et fonctions propres d'un domaine à anse fine vers les solutions de deux problèmes limites indépendants, l'un correspondant au problème sur la partie épaisse du domaine, l'autre à un problème associé à l'anse (essentiellement unidimensionnel mais toutefois non trivial selon la géométrie de l'anse) ; deuxièmement, la mise en œuvre d'un algorithme de minimisation des valeurs propres de Dirichlet, basé sur une méthode de variation du bord.

Une série d'exemples d'utilisation de ces algorithmes est proposée, contenant des résultats numériques et des explications concernant leur modélisation : tores, sphères, couples de domaines isospectraux, domaines à anse planes et cylindriques fines, ainsi que les 10 premiers domaines optimaux déterminés par l'algorithme d'optimisation de forme.

Une description du fonctionnement et de l'utilisation des programmes développés pour ce travail sur la base des algorithmes cités plus haut est finalement donnée.

Remerciements

Avant toute autre personne, c'est évidemment à mes directeurs de thèse Olivier Besson et Bruno Colbois que vont mes plus sincères remerciements pour leur patience, leurs encouragements, leurs conseils et bien-sûr pour tout ce qu'il m'ont appris tout au long de ces années de travail. Merci à eux d'avoir toujours cru en moi, particulièrement durant la pénible phase de rédaction, pendant laquelle leur confiance a été très importante pour moi. Un merci particulier à Olivier pour m'avoir initié à la programmation en *Fortran* et pour les exemples d'algorithmes numériques ayant servi de modèles de base à quelques uns de mes programmes.

Mes sincères remerciements à Peter Buser, Professeur à l'Ecole Polytechnique Fédérale de Lausanne, pour m'avoir fait l'honneur de prendre part à mon jury de thèse.

Merci également aux professeurs et collègues de l'institut, avec qui j'ai eu des discussions à la fois passionnantes et instructives. Merci en particulier à mes anciens collègues de bureau Régis Houssou et Souleyemane Doucouré pour les moments de doutes partagés et de rires. Merci également et surtout bonne chance à Régis Straubhaar qui, en quelque sorte, reprend le flambeau.

Merci aux amis qui ont osé me demander de leur expliquer ce que je faisais, et davantage encore à ceux qui ont ensuite su écouter. Finalement, le plus grand des mercis à Mélanie, devenue mon épouse puis la mère de mes filles Niki et Lara pendant la rédaction de cette thèse, pour m'avoir tant encouragé et supporté pendant toutes ces années, même et surtout dans les moments difficiles.

Octobre 2012
Fabien Crevoisier

Table des matières

1	Introduction	5
1.1	Problématique	5
1.2	Plan du travail	8
1.2.1	Spectre du Laplacien (chapitres 2 et 3)	8
1.2.2	Applications (chapitres 4 et 5)	11
1.2.3	Exemples et résultats numériques (chapitre 6)	14
1.2.4	Manuel d'utilisation (chapitre 7)	16
1.3	Notations et prérequis	16
1.3.1	Domaines et surfaces	16
1.3.2	Espaces de Sobolev	17
1.3.3	Théorie spectrale abstraite	18
1.3.4	Spectre du Laplacien	18
2	Analyse numérique du spectre du Laplacien	21
2.1	Formulation variationnelle du problème au valeurs propres	21
2.2	Méthode de Galerkin	23
2.3	Eléments finis	26
2.3.1	Eléments finis de Lagrange	26
2.3.2	Eléments finis simpliciaux	28
2.3.3	Méthode des éléments finis pour les domaines euclidiens	29
2.3.4	Méthode des éléments finis pour les domaines de coques	33
2.3.5	Méthode des éléments finis pour les domaines de surfaces	35
3	Résolution numérique	39
3.1	Maillage par macro-éléments	40
3.1.1	Macro-éléments de type quadrilatère	40
3.1.2	Macro-éléments de type quadrilatères à une face courbe	41
3.1.3	Macro-éléments de type triangle	42
3.1.4	Généralisation à des macro-éléments ayant plusieurs faces courbes	43
3.2	Traitements de triangulation	44
3.2.1	Renumérotation	44
3.2.2	Raffinement	45
3.3	Maillage de Delaunay	47

3.3.1	Appartenance d'un point au cercle circonscrit d'un triangle	50
3.3.2	Indice d'un point par rapport à une courbe	51
3.4	Maillage mixte	55
3.5	Calcul matriciel	55
3.5.1	Construction des matrices de rigidité et de masse	57
3.5.2	Construction des matrices locales	58
3.5.3	Intégration numérique	59
3.6	Résolution du problème aux valeurs propres matriciel	61
3.6.1	Méthode de la puissance	61
3.6.2	Méthodes de projection	63
3.6.3	Méthode des sous-espaces de Krylov	63
3.6.4	Algorithme de Lanczos	64
4	Problèmes d'anses fines	67
4.1	Enoncé du théorème	68
4.2	Preuve du théorème sous hypothèses fortes	72
4.2.1	Convergence d'une sous-suite de valeurs propres	73
4.2.2	Convergence faible d'une sous-suite de fonctions propres	74
4.2.3	Identification du problème limite sur Ω_1	75
4.2.4	Convergence forte des fonctions propres sur Ω_1	77
4.2.5	Identification du problème limite sur Ω_2	77
4.2.6	Convergence forte des fonctions propres sur Ω_2	78
4.2.7	Identification des solutions limites	79
4.2.8	Convergence de la suite entière de valeurs et fonctions propres	82
4.3	Preuve du théorème sous hypothèses faibles	82
4.4	Métrique non euclidienne	84
5	Optimisation de forme	85
5.1	Méthode de variation du bord	86
5.2	Dérivée normale d'une fonction propre de Dirichlet	90
5.2.1	Calcul matriciel	90
5.2.2	Construction des matrices	91
5.3	Changement de topologie des domaines	92
6	Exemples et résultats numériques	95
6.1	Premiers exemples	96
6.1.1	Carré, cylindre et tores	96
6.1.2	Disque et sphères	106
6.2	Vérification d'une propriété de symétrie	115
6.2.1	Demi-tores	115
6.2.2	Hémisphères	117
6.3	Domaines isospectraux	118
6.4	Domaines à anses fines	126

6.4.1	Anses planes brutes	127
6.4.2	Autres anses	130
6.5	Domaines optimaux	133
6.5.1	Présentation générale des domaines optimaux	135
6.5.2	Présentation détaillée des domaines optimaux	138
7	Manuel d'utilisation	151
7.1	Procédures générales	154
7.2	Maillage par macro-él. et traitements de triangulation	155
7.2.1	Définition du domaine	156
7.2.2	Maillage par macro-éléments	157
7.2.3	Fichier de conditions au bord	157
7.2.4	Assemblage de triangulations	157
7.2.5	Renumérotation	158
7.2.6	Raffinement	158
7.2.7	Fichier de visualisation	159
7.3	Maillage de Delaunay	160
7.4	Résolution du problème de Poisson	162
7.5	Résolution du problème aux valeurs propres	162
7.6	Procédures d'utilisation générale	164
7.7	Calcul de la dérivée normale d'une fonction propre de Dirichlet	166
7.8	Optimisation de forme	167
7.9	Quelques remarques géom. concernant la métrique	171

Chapitre 1

Introduction

Cette thèse traite du spectre du Laplacien sur des domaines de surfaces et plus précisément du calcul numérique des valeurs et fonctions propres pour des conditions au bord de Dirichlet et de Neumann. Après un instantané des recherches et résultats dans le domaine, nous donnons un plan du présent travail et précisons les notations et conventions utilisées, ainsi que les prérequis à la compréhension des concepts discutés.

1.1 Problématique

La résolution numérique d'équations aux dérivés partielles sur des domaines de \mathbb{R}^n avec $n \leq 3$ (et plus occasionnellement $n > 3$), en particulier l'estimation de valeurs propres du Laplacien, est devenue courante avec le développement de l'informatique. Parmi les techniques permettant de tels calculs, la plus largement répandue est probablement la méthode des éléments finis. Les origines de cette méthode sont diverses et l'ingénierie a parfois précédé les mathématiques dans la mise au point de la technique. Citons, pour ce qui est de la contribution au formalisme mathématique, Rayleigh ([42]), Ritz ([45]) et Galerkin ([21]), qui introduisent entre 1870 et 1915 le concept de méthode de projection, ainsi que Courant ([17]), qui considère en 1943 des fonctions de base à supports locaux, simplifiant ainsi considérablement leur construction. Il faut toutefois attendre 1973 pour que Strang et Fix ([48]) donnent une forme rigoureuse à cette méthode, dont Raviart et Thomas ([41]) proposent en 1983 une analyse très complète et pratique, détaillant bien des aspects de la résolution numérique de plusieurs problèmes, dont le spectre d'opérateurs elliptiques. Citons encore Babuška et Osborn ([7]) qui, en 1989, affinent les résultats de convergence des valeurs et vecteurs propres obtenus par des méthodes d'éléments finis. Notons toutefois qu'en 1956 déjà les ingénieurs Turner, Clough, Martin et Topp ([52]) utilisent les éléments finis dans l'industrie aéronautique.

Aujourd'hui, les calculs numériques de valeurs propres du Laplacien sur des domaines du plan, en particulier ceux usant des éléments finis, ne sont pas rares et servent notamment à conjecturer des résultats théoriques. Une des utilisations les plus récentes de tels algo-

rithmes est l'optimisation de forme pour les valeurs propres de Dirichlet sous contrainte de volume. Cette question apparaît à la fin du XIX^e siècle déjà, lorsque Rayleigh ([43] et [44]) conjecture que le disque minimise la première valeur propre. En 1923 et 1925, Faber ([20]) et Krahn ([30]) démontrent indépendamment cette conjecture à l'aide d'une technique de réarrangement. Krahn ([31]) et Szegö (cité dans [39]) démontrent ensuite que la réunion disjointe de deux disques identiques minimise la deuxième valeur propre.

Les problèmes de minimisation se corsent pour les valeurs propres suivantes et les résultats théoriques se limitent à l'existence d'un domaine optimal pour la troisième valeur propre et à sa connexité dans \mathbb{R}^2 et \mathbb{R}^3 (Bucur et Henrot [10]; Wolf et Keller [53]). Tout au plus dispose-t-on de quelques conjectures pour les valeurs propres suivantes, notamment par les résultats de Wolf et Keller : la troisième valeur propre, qui est minimisée localement par le disque, aurait pour domaine associé un disque en dimensions 2 et 3, et une réunion disjointe de 3 disques en dimensions supérieures ; la quatrième valeur propre aurait pour domaine associé, en dimensions 2 et 3, une réunion disjointe de deux disques dont le rapport des rayons peut être calculé. La question naturelle de savoir si les domaines optimaux ne sont que des réunions disjointes de disques se pose alors, mais Wolf et Keller remarquent par exemple que la treizième valeur propre du carré est inférieure à celle de n'importe quelle réunion disjointe de disques.

Cette situation conduit naturellement au développement d'algorithmes numériques d'optimisation de forme, permettant d'approcher des valeurs propres optimales ainsi que les domaines associés. En 2004, Oudet ([35]) propose ainsi des candidats pour les dix premières valeurs propres de Dirichlet du plan, validant les conjectures pour les troisième et quatrième valeurs propres et observant au passage que la cinquième valeur propre du disque n'est déjà plus minimisée par une réunion disjointe de disques. En 2012, Antunes et Freitas ([5]) améliorent ces résultats, proposant en particulier un septième domaine optimal différent. Ils présentent de plus des domaines optimaux pour des conditions de Neumann. A noter que les calculs d'Antunes et Freitas ont la particularité de ne pas employer de méthode d'éléments finis, mais une méthode dite MFS (pour *Method of Fundamental Solutions*), qui ne nécessite pas de mailler les domaines mais qui, en contrepartie, ne permet de traiter que des domaines simplement connexes.

Les problèmes d'optimisation de forme pour les valeurs propres du Laplacien ne se limitent toutefois pas à la seule contrainte de volume, ni à la seule condition au bord de Dirichlet. Citons par exemple, en ce qui concerne la première valeur propre de Dirichlet, le cas des polygones, pour lesquels Pòlya ([40]) a montré que les triangles et les quadrilatères réguliers sont optimaux, et les domaines contenu dans une boîte fixée, pour lesquels Buttazo et Dal Maso ([12]) ont montré l'existence d'une solution, avant qu'Henrot et Oudet ([27]) ne démontrent que les parties libres du bord du domaine ne peuvent pas consister en des morceaux de sphères. Citons encore le problème de la minimisation de la deuxième valeur propre de Dirichlet sous contrainte de volume et de convexité dans le plan, pour lequel Cox et Ross ([18]) ont montré l'existence d'une solution. Suite à des simulations

numériques, Troesch ([51]) a conjecturé qu'un tel domaine optimal était le stade, c'est-à-dire l'enveloppe convexe de deux disques tangents de même rayon, mais Henrot et Oudet ont montré que ce n'est pas le cas (entre autres propriété du domaine optimal). Oudet ([35]) propose également des simulations numériques pour approcher ce domaine optimal.

Si, comme on le voit, le calcul numérique de valeurs propres dans le plan constitue un domaine de recherche actif, très peu de travaux portent sur les domaines de surfaces. Citons Strohmaier et Uski ([49]), qui calculent avec un algorithme basé sur la méthode MPS (pour *Method of Particular Solutions*) la première valeur propre non triviale de la surface de Bolza. L'algorithme, quoique donnant un résultat d'une surprenante précision, semble toutefois devoir se limiter à des surfaces bien particulières, en l'occurrence les surfaces hyperboliques. Nous pouvons également citer Maintrot ([33]), qui calcule à l'aide d'une méthode d'éléments finis géodésiques quelques valeurs propres d'une surface de genre 2, vue comme recollement de parties du plan hyperbolique.

Le but premier de ce travail est donc de pallier à l'absence d'algorithme général pour les domaines de surfaces, en proposant une analyse numérique du calcul de telles valeurs propres, ainsi qu'un algorithme fonctionnel, complet et documenté de résolution numérique. Par algorithme général, on entend qu'il soit applicable quels que soient la métrique de la surface, le nombre de cartes nécessaires à sa paramétrisation, leurs recollements ou leurs formes. Les programmes ainsi développés ont donné pleine satisfaction lors de leur utilisation pour de nombreux exemples, dont ceux illustrés dans ce travail, et il est évidemment souhaité qu'ils servent à d'autres recherches à l'avenir.

Est également présenté un algorithme d'optimisation de forme des valeurs propres de Dirichlet sous contrainte de volume qui, s'il ne traite actuellement que de domaines du plan, a été pensé en vue d'une généralisation éventuelle aux domaines d'une surface fixée. Les résultats obtenus dans le plan ont également l'heureuse propriété de coller aux résultats actuels de référence dans le domaine.

Nous considérons finalement un autre problème classique de géométrie spectrale, à savoir le problème des domaines à anses fines. En effet, de tels domaines sont particulièrement délicats à étudier numériquement et il paraît nécessaire d'identifier le spectre des domaines limites, c'est-à-dire après écrasement de l'anse, afin de donner de bonnes estimations numériques des valeurs propres des domaines à anses fines. En 1987, Anné ([4]) a étudié ce problème pour les sous-variétés munies d'anses cylindriques ($X \times \mathbb{S}^1$), montrant que le spectre converge vers la réunion de celui de la partie épaisse et de celui, diminué d'une dimension, de l'anse (X) avec des conditions au bord de Dirichlet. Alors que ce travail ne concernait que les valeurs propres, Anné a de plus démontré une forme de convergence H^1 des fonctions propres lors de communications privées en 2008.

Ayant effectué au même moment un travail théorique sur les domaines à anses planes, avec pour objectif d'obtenir de bons résultats numériques sur de tels exemples, je présente

les résultats obtenus avec des outils essentiellement d'analyse fonctionnelle. Ils montrent une convergence comparable à la situation traitée par Anné, tant du point de vue des valeurs que des fonctions propres, avec toutefois une plus grande liberté donnée au niveau de la géométrie de l'anse et donc des solutions limites générées.

1.2 Plan du travail

Le présent travail peut être divisé en 4 parties relativement distinctes, comme décrit ci-dessous. Ce document serait toutefois incomplet sans les programmes joints, contenant non seulement les procédures de résolution des problèmes présentés dans ce travail, mais également de nombreuses explications sur leur fonctionnement, des commentaires utiles à un développement futur ou à une utilisation des différentes procédures, ainsi que toute une série d'exemples modélisés et résolus que l'utilisateur aura loisir d'étudier, d'interpréter et de modifier.

1.2.1 Spectre du Laplacien (chapitres 2 et 3)

La première partie concerne l'analyse numérique et la résolution numérique du spectre du Laplacien pour les domaines de surfaces, de manière générale. Elle donne un résultat théorique justifiant l'algorithme développé dans ce travail, puis en précise la mise en œuvre.

Analyse numérique

Le chapitre 2 met en place le problème considéré tout au long de ce travail, à savoir celui du spectre du Laplacien sur un domaine Ω d'une surface S , dont la formulation classique est la suivante :

$$(P) \quad \left\{ \begin{array}{l} \text{Trouver } u : \Omega \rightarrow \mathbb{R} \text{ et } \lambda \in \mathbb{R} \text{ tels que :} \\ -\Delta u = \lambda u \quad \text{dans } \Omega \\ u = 0 \quad \text{sur } \partial\Omega^D \\ \frac{\partial u}{\partial n} = 0 \quad \text{sur } \partial\Omega^N, \end{array} \right.$$

où Δ est le Laplacien défini par $\Delta = \text{div} \nabla$ et $\partial\Omega = \partial\Omega^D \amalg \partial\Omega^N$ est une partition du bord de Ω en un nombre fini de composantes connexes.

On en dérive la formulation variationnelle du problème, sur laquelle l'intégralité de ce travail est basée et dont les solutions u , appelées fonctions (ou vecteurs) propres, forment une base hilbertienne de $V(\Omega) = \{v \in H^1(\Omega) : v = 0 \text{ sur } \partial\Omega^D\}$ et λ , appelées valeurs propres, forment une suite positive de valeurs tendant vers l'infini :

$$(P_f) \quad \left\{ \begin{array}{l} \text{Trouver } u \in V(\Omega) \text{ et } \lambda \in \mathbb{R} \text{ tels que, pour tout } v \in V(\Omega) : \\ a(u, v) = \lambda b(u, v) , \end{array} \right.$$

avec

$$\begin{aligned} a(u, v) &= \int_{\Omega} \langle \nabla u, \nabla v \rangle, \\ b(u, v) &= \int_{\Omega} uv. \end{aligned}$$

Nous introduisons ensuite la méthode des éléments finis pour les domaines du plan, en passant par la méthode de Galerkin (aussi appelée méthode de projection, soit la restriction du problème aux valeurs propres à des sous-espaces de dimensions finies H_h convergeant vers $V(\Omega)$) et la définition des éléments finis de Lagrange (en particulier des n -simplexes de type k). La finalité de cette présentation détaillée est le résultat classique suivant :

Théorème 6 *Soit (τ_h) une famille régulière de triangulations de $\bar{\Omega}$ associées à un n -simplexe de type $k \geq 1$, $n \leq 3$, et respectant la partition $\partial\Omega = \partial\Omega^D \amalg \partial\Omega^N$. Alors, si $u \in H^{l+1}(\Omega)$ avec $1 \leq l \leq k$, il existe une constante C_1 telle que*

$$\|u - \Pi_h u\|_{H^1(\Omega)} \leq C_1 h^l \|u\|_{H^{l+1}(\Omega)}.$$

De plus, si $H_m \subset H^{l+1}(\Omega)$ avec $1 \leq l \leq k$ et H_m l'espace engendré par les m premières solutions de (P_f) , il existe des constantes h_0 et $C_2 > 0$ tels que, pour tout $h \leq h_0$,

$$0 \leq \lambda_{m,h} - \lambda_m \leq C_2 h^l \sup_{v \in H_m, \|v\|_{L^2(\Omega)}=1} \|v\|_{H^{l+1}(\Omega)}.$$

Nous développons finalement une méthode généralisée d'éléments finis permettant de traiter les domaines de surfaces via des triangulations dans des atlas. Nous définissons en particulier la notion d'inter-compatibilité de telles triangulations, dont la réunion disjointe forme moralement une triangulation du domaine sur la surface par passage au quotient. Le résultat suivant atteste de la convergence de cette méthode originale, constituant la base de travail de cette thèse :

Théorème 8 *Soit $\{(U_\alpha, f_\alpha)\}_{\alpha=1}^A$ un atlas d'une surface S tel qu'il existe des coques Ω_α*

d'intérieurs disjoints vérifiant $\bar{\Omega}_\alpha \subset f_\alpha(U_\alpha)$ et $\bar{\Omega} = \bigcup_{\alpha=1}^A \bar{\Omega}_\alpha$. Notons alors, pour tout α ,

$D_\alpha = f_\alpha^{-1}(\Omega_\alpha)$. Soient $(\tau_{h,\alpha})$ des familles régulières et inter-compatibles en α de triangulations des \bar{D}_α associées à un n -simplexe de type $k \geq 1$, $n \leq 3$, et respectant les partitions $\partial D_\alpha = \partial D_\alpha^D \amalg \partial D_\alpha^N \amalg \partial D_\alpha^0$ (associées à $\partial\Omega = \partial\Omega^D \amalg \partial\Omega^N$). Alors, si $u \in H^{l+1}(\Omega)$ avec $1 \leq l \leq k$, il existe une constante C_1 telle que

$$\|u - \Pi_h u\|_{H^1(\Omega)} \leq C_1 h^l \|u\|_{H^{l+1}(\Omega)}.$$

De plus, si $H_m \subset H^{l+1}(\Omega)$ avec $1 \leq l \leq k$ et H_m l'espace engendré par les m premières solutions de (P_f) , il existe des constantes h_0 et $C_2 > 0$ tels que, pour tout $h \leq h_0$,

$$0 \leq \lambda_{m,h} - \lambda_m \leq C_2 h^l \sup_{v \in H_m, \|v\|_{L^2(\Omega)}=1} \|v\|_{H^{l+1}(\Omega)}.$$

A noter que les parties ∂D_α^0 dans les cartes correspondent aux parties intérieures au domaine Ω sur lesquelles le recollement des différentes cartes est effectué.

Résolution numérique

Le chapitre 3 précise la mise en œuvre numérique de la méthode des éléments finis décrite dans le chapitre précédent. Deux aspects relativement distincts sont abordés : premièrement, le maillage de domaine et divers traitements de triangulations ; deuxièmement, la résolution numérique du problème aux valeurs propres lui-même, comportant la mise en œuvre du calcul matriciel à partir de la triangulation du domaine, soit la construction des matrices (avec intégration numérique), ainsi que la résolution numérique du problème matriciel aux valeurs propres ainsi obtenu.

La présentation des moyens de maillage débute par un mailleur par macro-éléments, c'est-à-dire par décomposition en sous-domaines. Les macro-éléments considérés ici sont quadrilatères avec ou sans faces courbes, mais aussi triangulaires (sans face courbe). A noter toutefois que les programmes joints n'acceptent qu'une face courbe au maximum, cela ayant suffi à modéliser tous les exemples considérés.

Nous poursuivons avec des procédures de traitement de triangulation : une renumérotation relativement élémentaire des nœuds (indispensable pour le traitement de domaines de surfaces) et un raffinement local de triangulation (avec contrôle de la qualité des éléments) sont détaillés et pallient aux principaux inconvénients du mailleur par macro-éléments, mais aussi de la deuxième méthode de maillage présentée, inspirée de la méthode de Voronoï-Delaunay et basée sur l'insertion successive de nœuds dans la triangulation. Cette méthode comporte deux avantages : premièrement elle permet une bonne automatisation puisqu'il suffit à l'utilisateur de fournir le bord du domaine à mailler (par exemple via une paramétrisation ou une liste de nœuds) ; deuxièmement la qualité de la triangulation est optimale relativement aux nœuds qui la composent, en le sens que la triangulation est de Delaunay. A noter toutefois que le mailleur de Delaunay n'a été implémenté dans les programmes joints que pour les domaines du plan. En effet, un tel maillage pour les domaines de surfaces n'a pas été jugé utile pour la modélisation des exemples considérés. Il s'agirait cependant d'un développement logique des programmes développés dans ce travail et aucun obstacle particulier ne semble empêcher une telle implémentation.

La résolution numérique, une fois la triangulation déterminée, nécessite la construction des matrices de rigidité et de masse du système. Cela passe par le choix d'une base de fonctions admissibles, en l'occurrence les fonctions formes. Une telle fonction est associée à un nœud de la triangulation et est nulle partout sauf sur les éléments contenant ce nœud. Les matrices alors obtenues étant symétriques définies positives, on emploie une décomposition de Cholesky de la matrice de masse pour transformer le problème matriciel aux valeurs propres généralisé en un problème matriciel aux valeurs propres usuel. Est

ensuite présenté le détail de la construction des matrices du système par assemblage de matrices locales, calculées successivement sur chaque élément de la triangulation (en prenant référence sur le 2-simplexe standard). En particulier, l'intégration numérique est discutée et deux options sont proposées concernant les formules de quadrature à utiliser : l'une ayant pour avantage d'être exacte lorsque la métrique est euclidienne, l'autre de rendre la matrice de masse diagonale, facilitant ainsi la résolution numérique du problème matriciel ainsi obtenu. Cette dernière méthode est assimilée à du mass-lumping et on discute de son effet sur les solutions du système.

Enfin, l'algorithme itératif de Lanczos est brièvement présenté. Basé sur les méthodes de la puissance et de projection, il permet le calcul d'un (petit) nombre de valeurs propres d'une partie fixée du spectre, ainsi que des vecteurs propres associés.

1.2.2 Applications (chapitres 4 et 5)

La deuxième partie est constituée de deux applications des algorithmes décrits dans les chapitres précédents, à savoir le traitement des domaines à anses fines et un algorithme de résolution numérique du problème d'optimisation de forme.

Problèmes d'anses fines

Dans le chapitre 4, nous nous intéressons aux domaines à anses fines, qui constituent des exemples a priori peu adaptés au calcul numérique du spectre du Laplacien. Le résultat original démontré ici, qui concerne essentiellement les domaines du plan, montre une convergence des valeurs propres, mais aussi des fonctions propres d'un domaine à anse fine vers les solutions de deux problèmes limites indépendants, l'un correspondant au problème sur la partie épaisse du domaine, l'autre à un problème associé à l'anse, essentiellement unidimensionnel mais toutefois non trivial selon la géométrie de l'anse. Le corollaire est un calcul numérique facilité du spectre de domaines à anses fines, les problèmes limites étant beaucoup plus abordables numériquement. Dans le cas de géométries non triviales de l'anse (épaisseur variable, embranchements, trous), on entrevoit l'utilité du développement futur d'un algorithme de résolution numérique du problème limite sur la partie fine à travers quelques exemples.

Plus précisément, nous considérons le domaine Ω^ε représenté ci-dessous, la partie épaisse du domaine correspondant à Ω_1 et la partie fine à Ω_2^ε . Cette dernière est obtenue par écrasement h^ε d'un domaine initial Ω_2 le long d'une courbe.

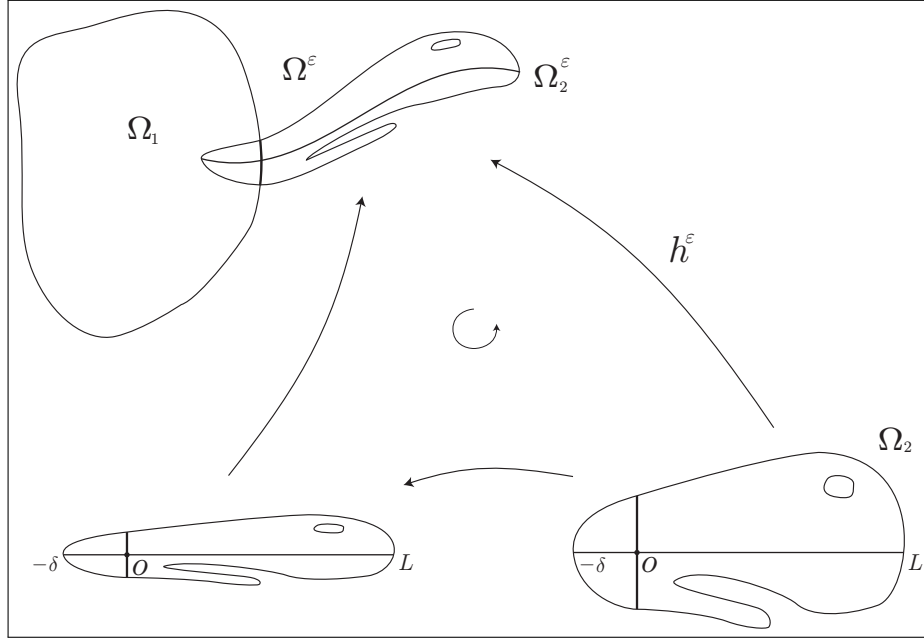


FIGURE 1.1 – Définition de l'anse.

On considère alors les problèmes suivants (pour $\varepsilon > 0$ petit) :

$$(P_\varepsilon) \quad \begin{cases} \text{Trouver } u^\varepsilon \in V(\Omega^\varepsilon) \text{ et } \lambda^\varepsilon \in \mathbb{R} \text{ tels que, pour tout } v^\varepsilon \in V(\Omega^\varepsilon) : \\ \int_{\Omega^\varepsilon} \langle \nabla u^\varepsilon, \nabla v^\varepsilon \rangle = \lambda^\varepsilon \int_{\Omega^\varepsilon} u^\varepsilon v^\varepsilon, \end{cases}$$

$$(P_0) \quad \begin{cases} \text{Trouver } (u_1, u_2) \in V(\Omega_1) \times V_0(\Omega_2) \text{ et } \lambda \in \mathbb{R} \text{ tels que,} \\ \text{pour tout } (v_1, v_2) \in V(\Omega_1) \times V_0(\Omega_2) : \\ \int_{\Omega_1} \langle \nabla u_1, \nabla v_1 \rangle = \lambda \int_{\Omega_1} u_1 v_1 \text{ et } \int_{\Omega_2} \frac{\partial u_2}{\partial x_1} \frac{\partial v_2}{\partial x_1} = \lambda \int_{\Omega_2} u_2 v_2, \end{cases}$$

avec les espaces suivants :

$$V(\Omega^\varepsilon) = \{v \in H^1(\Omega^\varepsilon) : v = 0 \text{ sur } \partial\Omega^{\varepsilon,D}\},$$

$$V(\Omega_1) = \{v \in H^1(\Omega_1) : v = 0 \text{ sur } \partial\Omega_1^D\},$$

$$V_0(\Omega_2) = \left\{ v \in H^1(\Omega_2) : v = 0 \text{ sur } \partial\Omega_2^D \text{ et dans } (\mathbb{R}^- \times \mathbb{R}) \cap \Omega_2, \frac{\partial v}{\partial x_2} = 0 \text{ dans } \Omega_2 \right\},$$

les ensembles $\partial\Omega^{\varepsilon,D}$, $\partial\Omega_1^D$ et $\partial\Omega_2^D$ correspondant aux parties du bord de Ω^ε , Ω_1 et Ω_2 respectivement sur lesquelles des conditions de Dirichlet sont imposées. On démontre alors le résultat suivant :

Théorème 10

- A) Convergence des valeurs propres : les solutions λ^ε de (P_ε) convergent vers les solutions λ de (P_0) lorsque $\varepsilon \rightarrow 0$.
- B) Convergence des espaces propres : on peut choisir des bases $\{u^{\varepsilon,k}\}_{k>0}$ de solutions de (P_ε) et $\{u_1^k, u_2^k\}_{k>0}$ de solutions de (P_0) telles que :
- a) $u^{\varepsilon,k}|_{\Omega_1} \rightarrow u_1^k$ dans $H^1(\Omega_1)$ lorsque $\varepsilon \rightarrow 0$.
 - b) $\frac{\partial}{\partial x_2}(\varepsilon^{1/2}u^{\varepsilon,k} \circ h^\varepsilon) \rightarrow \frac{\partial u_2^k}{\partial x_2}$ dans $L^2(\Omega_2)$ lorsque $\varepsilon \rightarrow 0$.
 - c) $\varepsilon^{1/2}u^{\varepsilon,k} \circ h^\varepsilon \rightarrow u_2^k$ dans $H^1(\Omega_2)$ lorsque $\varepsilon \rightarrow 0$.
 - d) En particulier, $\varepsilon^{1/2}u^{\varepsilon,k} \circ h^\varepsilon \rightarrow u_2^k$ dans $L^2(\Omega_2)$ lorsque $\varepsilon \rightarrow 0$.

A noter que des expérimentations numériques ont permis d'observer une convergence semblable pour des anses cylindrique, mais que la démonstration proposée n'a pas pu être adaptée à cette situation.

Optimisation de forme

Le but du chapitre 5 est la mise en œuvre d'un algorithme de minimisation des valeurs propres de Dirichlet pour les domaines du plan, qui soit éventuellement généralisable aux domaines de surfaces. L'approche retenue est une méthode de variation du bord du domaine basée sur la condition d'optimalité au premier ordre d'une fonction-coût J , en l'occurrence le produit du volume et de la valeur propre considérée. La première étape est la définition d'une dérivée sur l'espace des formes. Cette dérivée, associée à un domaine Ω et à un champ de déformation V , est définie par

$$d(J(\Omega))(\Omega, V) = \lim_{t \rightarrow 0} \frac{J(\Omega_t) - J(\Omega)}{t},$$

avec

$$\Omega_t = \{x + tV(x) : x \in \Omega\}.$$

En considérant $J(\Omega) = |\Omega| \cdot \lambda(\Omega)$, on trouve ainsi

$$d(J(\Omega))(\Omega, V) = \int_{\partial\Omega} \left(\lambda - |\Omega| \left(\frac{\partial u}{\partial n} \right)^2 \right) \cdot V \cdot n \, d\sigma,$$

avec λ la valeur propre considérée et u la fonction propre associée, $\|u\|_{L^2(\Omega)} = 1$.

On décrit ensuite l'algorithme d'optimisation de forme par variation du bord du domaine, avant de détailler deux points particuliers, à savoir le calcul numérique de la dérivée normale $\frac{\partial u}{\partial n}$ d'une fonction propre de Dirichlet u sur le bord d'un domaine et le traitement des changements topologiques éventuels du domaine en cours d'algorithme.

Pour déterminer numériquement la dérivée normale, on considère connue une solution (u, λ) du problème variationnel aux valeurs propres et on identifie la dérivée normale comme unique solution du problème variationnel

$$(P_n) \quad \begin{cases} \text{Trouver } \omega \in L^2(\partial\Omega) \text{ tel que, pour tout } v \in H^1(\Omega) : \\ a(\omega, v) = b(v) \end{cases},$$

avec

$$\begin{aligned} a(\omega, v) &= \int_{\partial\Omega} \omega v, \\ b(v) &= \int_{\Omega} \langle \nabla u, \nabla v \rangle - \lambda \int_{\Omega} uv. \end{aligned}$$

On résoud alors numériquement ce problème de manière très semblable au problème aux valeurs propres dont il est issu : constitution d'une base des fonctions admissibles à l'aide des fonctions formes, construction de la matrice symétrique définie positive et du membre de droite du système par assemblage de matrices locales (associées à chaque élément frontière de la triangulation) et résolution du système linéaire obtenu à l'aide d'une décomposition de Cholesky.

Les situations possibles d'auto-intersection de la courbe polygonale définissant le bord du domaine au fil de ses déformations durant les itérations de l'algorithme d'optimisation de forme sont finalement listées et un traitement de chacun de ces cas est proposé. Il s'agit essentiellement de redéfinir la courbe de manière adéquate lorsque le domaine se recoupe ou se sépare.

L'algorithme ainsi décrit a permis, non sans difficulté et de manière expérimentale, de déterminer les domaines du plan optimaux pour quelques premières valeurs propres de Dirichlet (voir plus loin). L'optimisation des domaines d'une surface donnée paraît également envisageable en tant que développement naturel de l'algorithme obtenu, même s'il faudra pour cela passer quelques obstacles, comme la variation du volume du domaine au cours des itérations successives.

1.2.3 Exemples et résultats numériques (chapitre 6)

Une série d'exemples numériques d'utilisation des algorithmes décrits dans les chapitres précédents et développés pour ce travail, contenant des résultats numériques et de brèves explications sur la modélisation des domaines choisis, constitue ce chapitre. Les valeurs propres numériques sont listées et comparées aux éventuelles valeurs théoriques. Les fonctions propres associées sont également illustrées, tant sur les surfaces que dans les cartes paramétriques.

Les premiers exemples traités permettent d'illustrer la modélisation de domaines classiques d'abord très simples, puis progressivement plus complexes. La première famille débute par le carré, maillé à l'aide d'un seul macro-élément et muni de la métrique euclidienne. Par quotient, on passe ensuite au cylindre (identification d'une paire de faces), puis au tore plat (identification de deux paires de faces). Les résolutions manuelles de ces domaines permettent de comparer les solutions théoriques et numériques. Pour finir cette série, l'ajout d'une métrique non euclidienne au tore permet de traiter le cas d'un tore de \mathbb{R}^3 . Dans le même ordre d'idée mais avec un maillage initial plus complexe, la deuxième série d'exemples débute par le disque, maillé ici avec 5 macro-éléments quadrilatères à faces courbes, et se termine avec la sphère de \mathbb{R}^3 , vue comme le quotient de deux disques munis de la métrique provenant de la projection stéréographique. Les valeurs propres théoriques de ces domaines sont également comparées aux valeurs numériques.

Le calcul numérique de demi-tores et hémisphères permettent également de vérifier l'incidence des symétries sur le spectre. On considère également deux paires de domaines isospectraux non isométriques, aux maillages plus ou moins complexes (7 et 21 macro-éléments respectivement). De par la nature de la preuve d'isospectralité de ces domaines (par transplantation des fonctions propres), on s'attend à ce que les algorithmes développés se prêtent bien à la comparaison des valeurs propres de ces domaines, ce qui est vérifié.

Quelques exemples de domaines à anses raisonnablement fines sont illustrés : de partie épaisse carrée, leurs anses sont tantôt planes, tantôt cylindriques et leurs raccords tantôt brutes, tantôt lisses. Particulièrement délicats à mailler, ils ont servi à conjecturer le théorème 10 à propos des domaines à anses fines planes et indiquent une généralisation de ce résultat aux anses cylindriques. La convergence est d'ailleurs particulièrement rapide, tant au niveau des valeurs que des fonctions propres, malgré la largeur modeste des anses.

Finalement, quelques uns des domaines obtenus par l'algorithme d'optimisation de forme sont présentés, accompagnés de leurs valeurs et fonctions propres. Ces dix premiers domaines, illustrés ci-dessous, sont discutés et confrontés (avec satisfaction) aux éventuels domaines optimaux théoriques (première et deuxième valeurs propres), résultats généraux théoriques et autres conjectures (troisième et quatrième valeurs propres), ainsi qu'aux résultats numériques d'Oudet ([35]) et du duo Antunes-Freitas ([5]). La septième valeur propre optimale, que mon algorithme garantit inférieure à 106.63, est à ce propos particulièrement intéressante, Antunes et Freitas proposant un domaine connexe et Oudet un domaine non-connexe. Bien que le domaine obtenu dans mon travail soit connexe, il semble sensiblement différent de celui avancé par Antunes et Freitas, qui semblent détenir un meilleur candidat.

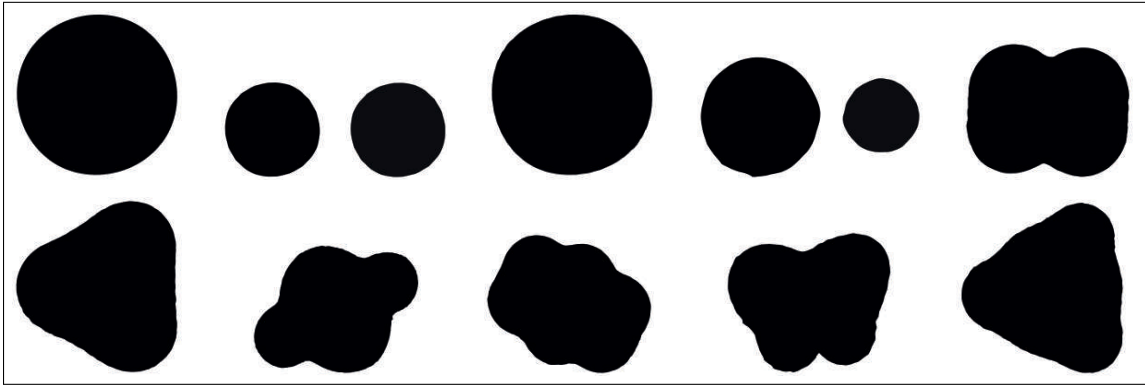


FIGURE 1.2 – Domaines optimaux déterminés numériquement.

1.2.4 Manuel d'utilisation (chapitre 7)

Partie majeure de ce travail, les programmes joints à ce document, développés sur la base des algorithmes décrits dans les chapitres précédents et les concrétisant, ne sont utiles et surtout utilisables que s'ils sont rendus suffisamment accessibles. C'est dans ce but que ce chapitre a été écrit. Il consiste en une description relativement détaillée du fonctionnement et de l'utilisation de ces programmes et est essentiellement destiné aux personnes intéressées par la programmation des algorithmes de modélisation et de résolution ou par leur utilisation, que ce soit pour traiter des exemples originaux ou dans un but de développement de nouveaux algorithmes. L'arborescence des fichiers joints y est détaillée, de même que les paramètres les plus importants des procédures principales. La structure et le rôle des fichiers de données et des routines de définition sont expliqués pour permettre à l'utilisateur de créer ses propres exemples. Les différentes options d'exécution des procédures sont également présentées. De nombreux détails complémentaires se trouvent naturellement à l'intérieur des programmes eux-mêmes et les nombreux exemples joints, entièrement modélisés, peuvent servir de modèles pour le traitement de nouveaux cas.

1.3 Notations et prérequis

Ci-dessous est précisé un certain nombre de notations et conventions adoptées pour ce travail, ainsi que quelques faits classiques d'analyse fonctionnelle et de géométrie ; pour plus de détails, le lecteur se référera par exemple à [9], [1], [19], ou [15] selon ses besoins. A noter que la section 7.9 contient quelques rappels élémentaires de géométrie différentielle au sujet des paramétrisations locales de surfaces et métriques associées (voir aussi [14]).

1.3.1 Domaines et surfaces

Pour toute la durée de ce travail, on considère une surface S régulière (disons \mathcal{C}^∞) et orientable, éventuellement à bord. Le domaine Ω de S désigne un ouvert borné (relativement

compact), de bord $\partial\Omega = \partial\Omega^D \amalg \partial\Omega^N$ (\amalg désignant la réunion disjointe) régulier (disons \mathcal{C}^1 par morceaux sans cusp), avec $\partial\Omega^D$ et $\partial\Omega^N$ possédant un nombre fini de composantes connexes, $\partial\Omega^D$ correspondant à des conditions de Dirichlet et $\partial\Omega^N$ à des conditions de Neumann.

1.3.2 Espaces de Sobolev

On note $L^2(\Omega)$ l'espace de Hilbert des fonctions de Ω de carré sommable, muni du produit scalaire

$$\langle u, v \rangle_{L^2(\Omega)} = \int_{\Omega} uv,$$

et $H^1(\Omega)$ l'espace de Hilbert (et dit de Sobolev) des fonctions de $L^2(\Omega)$ dont les dérivées partielles (au sens des distributions) sont dans $L^2(\Omega)$, muni du produit scalaire

$$\langle u, v \rangle_{H^1(\Omega)} = \langle u, v \rangle_{L^2(\Omega)} + \langle u, v \rangle_{1,\Omega},$$

avec

$$\langle u, v \rangle_{1,\Omega} = \int_{\Omega} \langle \nabla u, \nabla v \rangle$$

(∇ désignant le gradient).

Comme Ω est borné, de bord suffisamment régulier, le théorème de Rellich s'applique et l'injection canonique de $H^1(\Omega)$ dans $L^2(\Omega)$ est compacte, c'est-à-dire que l'image de tout borné de $H^1(\Omega)$ est relativement compact pour la norme de $L^2(\Omega)$.

On note encore $H_0^1(\Omega)$ le sous-espace de Hilbert de $H^1(\Omega)$ constitué des fonctions nulles au bord de Ω au sens de la trace : comme $\partial\Omega$ est suffisamment régulier, il existe une application trace d'ordre 0 continue

$$\gamma_0 : H^1(\Omega) \longrightarrow L^2(\partial\Omega), \quad u \longmapsto u|_{\partial\Omega}.$$

L'image de $H^1(\Omega)$ par γ_0 est un sous-espace strict de $L^2(\partial\Omega)$, noté $H^{1/2}(\partial\Omega)$.

Comme Ω est borné, l'inégalité de Poincaré s'applique et il existe une constante $C = C(\Omega)$ telle que

$$\int_{\Omega} u^2 \leq C \int_{\Omega} |\nabla u|^2,$$

pour tout $u \in H_0^1(\Omega)$. En corollaire, la semi-norme $|\cdot|_{1,\Omega}$ induite par $\langle \cdot, \cdot \rangle_{1,\Omega}$ est une norme sur $H_0^1(\Omega)$, équivalente à la norme induite par $H^1(\Omega)$.

On définit encore, pour tout espace de Hilbert H , la convergence faible d'une famille $(u_\varepsilon)_{\varepsilon>0}$ de H vers $u \in H$ de la manière suivante :

$$u_\varepsilon \rightharpoonup u \iff \langle u - u_\varepsilon, v \rangle \rightarrow 0, \quad \text{pour tout } v \in H.$$

Cette convergence est, comme son nom l'indique, plus faible que la convergence usuelle de H , dite forte. Elle possède cependant des propriétés intéressantes, notamment le fait que, de toute suite bornée de H , on peut extraire une sous-suite faiblement convergente. Cela implique, grâce au théorème de Rellich, que toute suite faiblement convergente de $H^1(\Omega)$ converge fortement dans $L^2(\Omega)$ vers la même limite.

Citons finalement le fait que toute suite faiblement convergente dont les normes convergent vers la norme de la limite faible est fortement convergente (toujours vers la même limite).

1.3.3 Théorie spectrale abstraite

Considérons H et L deux espaces de Hilbert séparables, $i : H \rightarrow L$ linéaire, injective, continue et compacte, ainsi que $a : H \times H \rightarrow \mathbb{R}$ bilinéaire, continue, symétrique et coercitive (i.e. telle qu'il existe une constante $\alpha > 0$ pour laquelle $a(v, v) \geq \alpha \|v\|_H$ pour tout $v \in H$). Soit alors le problème

$$(P_f) \quad \begin{cases} \text{Trouver } u \in H \text{ et } \lambda \in \mathbb{R} \text{ tels que, pour tout } v \in H : \\ a(u, v) = \lambda \langle u, v \rangle_L . \end{cases}$$

Alors il existe des solutions $\{(w_k, \lambda_k)\}_{k \geq 1}$ de (P_f) telles que les w_k , appelés vecteurs propres, forment une base hilbertienne de H et les λ_k , appelés valeurs propres, forment une suite positive (croissante) tendant vers l'infini.

Le quotient de Rayleigh associé à (P_f) est défini par

$$QR(v) = \frac{a(v, v)}{\|v\|_L^2},$$

pour tout $v \in H \setminus \{0\}$, et permet de caractériser les valeurs propres grâce à la propriété suivante, appelée théorème du Min-Max (ou principe de Poincaré) :

$$\lambda_k = \min_{V \in \mathcal{V}_k} \max_{v \in V \setminus \{0\}} QR(v),$$

où \mathcal{V}_k est l'ensemble des sous-espaces de dimension k de H .

1.3.4 Spectre du Laplacien

Considérons le Laplacien Δ sur Ω , défini par la formule de Green

$$-\int_{\Omega} \Delta u v = \int_{\Omega} \langle \nabla u, \nabla v \rangle - \int_{\partial\Omega} \frac{\partial u}{\partial n} v.$$

Alors le spectre du Laplacien sur Ω est un cas particulier du problème (P_f) , avec $H = V(\Omega)$, $L = L^2(\Omega)$, $i : H \hookrightarrow L$ l'injection canonique et $a(u, v) = \int_{\Omega} \langle \nabla u, \nabla v \rangle$. Les résultats

énoncés pour (P_f) restent donc vrais pour le Laplacien.

En particulier, le théorème du Min-Max implique le principe de monotonie suivant pour les valeurs propres de Dirichlet :

$$\Omega_1 \subset \Omega_2 \implies \lambda_k(\Omega_1) \geq \lambda_k(\Omega_2).$$

Il est également facile de voir l'effet d'une homothétie d'une surface sur le spectre d'un domaine (et en particulier l'effet d'une homothétie d'un domaine de \mathbb{R}^n sur son spectre) :

$$\lambda_k(l \cdot \Omega) = \frac{1}{l^2} \lambda_k(\Omega),$$

pour tout $l > 0$.

Citons encore le théorème de Courant sur les domaines nodaux (c'est-à-dire les ensembles connexes sur lesquels une fonction propre donnée ne change pas de signe) des domaines connexes : si $\{(w_k, \lambda_k)\}_{k \geq 1}$ sont des solutions de (P_f) dont les w_k forment une base hilbertienne de $V(\Omega)$ et les λ_k une suite croissante, alors le nombre de domaines nodaux de w_k est plus petit ou égal à k , pour tout $k \geq 1$. En corollaire, la première fonction propre a un signe constant et la première valeur propre est simple. De plus, la deuxième fonction propre a exactement deux domaines nodaux.

Chapitre 2

Analyse numérique du spectre du Laplacien

A partir du problème classique du spectre du Laplacien sur des domaines de surfaces, nous allons dériver sa formulation variationnelle. Adaptée à l'approximation numérique, elle constitue le contexte de ce travail. Dès lors, la formulation classique du problème aux valeurs propres ne sera plus évoquée ; pour plus de détails, notamment au sujet du lien entre les solutions classiques et variationnelles du problème aux valeurs propres, le lecteur se référera à [23].

L'approximation du spectre par la méthode de Galerkin sera ensuite présentée et nous nous intéresserons plus particulièrement à la méthode des éléments finis pour les domaines du plan ; le lecteur est invité à se référer à [16] ou [41] pour plus de détails à propos de ces méthodes, notamment dans le cas de domaines non polygonaux. A ce propos, il est à noter que l'ouvrage de Raviart et Thomas ([41]) a constitué l'ouvrage de référence pour la rédaction de ce chapitre.

Nous développerons finalement une méthode généralisée d'éléments finis permettant de traiter les domaines de surfaces via des triangulations dans des atlas et démontrerons sa convergence.

2.1 Formulation variationnelle du problème aux valeurs propres

Soient S une surface et Ω un domaine de S , de bord $\partial\Omega = \partial\Omega^D \cup \partial\Omega^N$, avec $\partial\Omega^D$ et $\partial\Omega^N$ possédant un nombre fini de composantes connexes. Le but est de résoudre numériquement

le problème

$$(P) \quad \left\{ \begin{array}{l} \text{Trouver } u : \Omega \rightarrow \mathbb{R} \text{ et } \lambda \in \mathbb{R} \text{ tels que :} \\ -\Delta u = \lambda u \quad \text{dans } \Omega \\ u = 0 \quad \text{sur } \partial\Omega^D \\ \frac{\partial u}{\partial n} = 0 \quad \text{sur } \partial\Omega^N, \end{array} \right.$$

où Δ est le Laplacien défini par $\Delta = \operatorname{div} \nabla$.

Pour cela, nous allons étudier le problème suivant, dit *problème variationnel associé à (P)* :

$$(P_f) \quad \left\{ \begin{array}{l} \text{Trouver } u \in V(\Omega) \text{ et } \lambda \in \mathbb{R} \text{ tels que, pour tout } v \in V(\Omega) : \\ a(u, v) = \lambda b(u, v), \end{array} \right.$$

avec

$$\begin{aligned} V(\Omega) &= \{v \in H^1(\Omega) : v = 0 \text{ sur } \partial\Omega^D\}, \\ a(u, v) &= \int_{\Omega} \langle \nabla u, \nabla v \rangle, \\ b(u, v) &= \int_{\Omega} uv. \end{aligned}$$

La raison est que toute solution de (P) est solution de (P_f). En effet, si u est une solution de (P) et si $v \in V(\Omega)$, alors on a

$$-\int_{\Omega} \Delta u v = \lambda \int_{\Omega} uv = \lambda b(u, v).$$

Mais, la formule de Green donne

$$-\int_{\Omega} \Delta u v = \int_{\Omega} \langle \nabla u, \nabla v \rangle - \int_{\partial\Omega} \frac{\partial u}{\partial n} v = \int_{\Omega} \langle \nabla u, \nabla v \rangle = a(u, v),$$

puisque $v = 0$ sur $\partial\Omega^D$ et $\frac{\partial u}{\partial n} = 0$ sur $\partial\Omega^N$.

De plus, si $\partial\Omega$ est suffisamment régulier (\mathcal{C}^1 par morceaux sans cusp), alors les solutions de (P_f) appartiennent à $H^2(\Omega)$. Ainsi, dans ce cas, le raisonnement inverse montre que les solutions de (P) sont les solutions de (P_f) (voir [23] pour plus de détails).

Les solutions u , respectivement λ , de (P_f) sont appelées *fonctions* (ou *vecteurs*) *propres*, respectivement *valeurs propres*, de Ω .

Pour la suite, remarquons que, dans le cas où $\partial\Omega^D$ est de mesure nulle, il suffit de remplacer $V(\Omega)$ par $\{v \in H^1(\Omega) : v \perp u_0\}$ pour exclure la solution triviale $u_0 \equiv 1$, $\lambda = 0$.

2.2 Méthode de Galerkin

Considérons H et L deux espaces de Hilbert séparables, $i : H \rightarrow L$ linéaire, injective, continue et compacte ainsi que $a : H \times H \rightarrow \mathbb{R}$ bilinéaire, continue, symétrique et coercitive. Soit alors le problème

$$(P_f) \quad \begin{cases} \text{Trouver } u \in H \text{ et } \lambda \in \mathbb{R} \text{ tels que, pour tout } v \in H : \\ a(u, v) = \lambda \langle u, v \rangle_L . \end{cases}$$

Remarquons qu'il s'agit d'une généralisation du problème (P_f) de la section précédente, puisqu'il suffit de prendre $H = V(\Omega)$, $L = L^2(\Omega)$, $i : H \hookrightarrow L$ l'injection canonique et $a(u, v) = \int_{\Omega} \langle \nabla u, \nabla v \rangle$ pour retrouver le problème en question.

Les solutions u , respectivement λ , de (P_f) sont appelées *vecteurs propres*, respectivement *valeurs propres*, de (P_f) . Grâce à l'analyse fonctionnelle, on montre facilement le résultat suivant (voir [41]) :

Théorème 1 *Il existe une base hilbertienne $\{w_i\}_{i \geq 1}$ de H formée de vecteurs propres de (P_f) , de valeurs propres associées $0 \leq \lambda_1 \leq \lambda_2 \leq \dots \rightarrow \infty$.*

Notons $H_h \subset H$ un sous espace vectoriel de dimension $N = N_h < \infty$, avec $h > 0$ un indice (que l'on fera tendre vers 0 et qui contrôle en quelque sorte la taille de H_h par rapport à H). On considère alors le problème suivant, dit *problème approché associé* à (P_f) :

$$(P_{f,h}) \quad \begin{cases} \text{Trouver } u_h \in H_h \text{ et } \lambda_h \in \mathbb{R} \text{ tels que, pour tout } v_h \in H_h : \\ a(u_h, v_h) = \lambda_h \langle u_h, v_h \rangle_L . \end{cases}$$

On trouve alors, de la même manière que le théorème 1, le résultat suivant :

Théorème 2 *Il existe une base orthonormée $\{w_{i,h}\}_{i=1}^N$ de H_h formée de vecteurs propres de $(P_{f,h})$, de valeurs propres associées $0 \leq \lambda_{1,h} \leq \lambda_{2,h} \leq \dots \leq \lambda_{N,h}$.*

On peut toutefois faire une preuve constructive de ce théorème, ce qui a l'avantage de fournir une méthode explicite de résolution du problème $(P_{f,h})$ qui sera employée lors de la résolution numérique.

Preuve constructive

Soit $\{\varphi_i\}_{i=1}^N$ une base de H_h .

On cherche $u_h = \sum_{j=1}^N \xi_j \varphi_j$ et $\lambda_h \in \mathbb{R}$ tels que $a(u_h, \varphi_i) = \lambda_h \langle u_h, \varphi_i \rangle_L$, pour tout i .

Mais $a(u_h, \varphi_i) = \sum_{j=1}^N \xi_j a(\varphi_j, \varphi_i)$ et $\langle u_h, \varphi_i \rangle_L = \sum_{j=1}^N \xi_j \langle \varphi_j, \varphi_i \rangle_L$.

Ainsi, en notant $r_{ij} = a(\varphi_j, \varphi_i)$, $m_{ij} = \langle \varphi_j, \varphi_i \rangle_L$, $\xi = (\xi_i)_{i=1}^N$, $R = (r_{ij})_{i,j=1}^N$ et

$M = (m_{ij})_{i,j=1}^N$, on trouve que $(P_{f,h})$ est équivalent à chercher $\xi \in \mathbb{R}^N$ et $\lambda_h \in \mathbb{R}$ vérifiant l'équation matricielle

$$R\xi = \lambda_h M\xi. \quad (2.1)$$

Remarquons encore que R et M sont des matrices symétriques définies positives. Notons alors $M = N^t N$ la décomposition de Cholesky de M , avec N une matrice triangulaire supérieure. Ainsi, chercher ξ et λ_h vérifiant l'équation matricielle (2.1) équivaut à chercher ξ et λ_h vérifiant, en notant $(N^{-1})^t = N^{-t}$,

$$N^{-t} R\xi = \lambda_h N\xi$$

ou encore, en notant $A = N^{-t} R N^{-1}$ et $\eta = N\xi$, au problème matriciel

$$(P_m) \quad \begin{cases} \text{Trouver } \eta \in \mathbb{R}^N \text{ et } \lambda_h \in \mathbb{R} \text{ tels que :} \\ A\eta = \lambda_h \eta. \end{cases}$$

On remarque alors que A est une matrice symétrique définie positive. Par un résultat bien connu d'algèbre linéaire, on sait qu'il existe alors des solutions $\{\eta_i\}_{i=1}^N$ et $\{\lambda_{i,h}\}_{i=1}^N$ du problème (P_m) telles que $A\eta_i = \lambda_{i,h}\eta_i$, $0 \leq \lambda_{1,h} \leq \lambda_{2,h} \leq \dots \leq \lambda_{N,h}$ et $\langle \eta_i, \eta_j \rangle_L = \delta_{ij}$, pour tous i, j .

Ainsi, les $\{w_{i,h}\}_{i=1}^N$ et $\lambda_{i,h}$ associés sont solutions de $(P_{f,h})$, par construction.

De plus, on a

$$\begin{aligned} \langle w_{i,h}, w_{j,h} \rangle_L &= \left\langle \sum_{k=1}^N \xi_{ik} \varphi_k, \sum_{l=1}^N \xi_{jl} \varphi_l \right\rangle_L = \sum_{k,l=1}^N \xi_{ik} \xi_{jl} \langle \varphi_k, \varphi_l \rangle_L \\ &= \langle M\xi_i, \xi_j \rangle_L = \langle N^t \xi_i, N^t \xi_j \rangle_L = \langle \eta_i, \eta_j \rangle_L = \delta_{ij}. \end{aligned}$$

□

Il s'agit maintenant de voir que, si H_h converge vers H (dans un sens à préciser) lorsque h tend vers 0, alors il en est de même des solutions de (P_f) et $(P_{f,h})$.

Notons

$$H_m = \langle w_1, \dots, w_m \rangle,$$

où $\{w_i\}_{i \geq 1}$ est une base hilbertienne de L formée des vecteurs propres de (P_f) , avec $w_i \in H$ pour tout i et $0 \leq \lambda_1 \leq \lambda_2 \leq \dots \rightarrow \infty$. La bonne notion de convergence de H_h vers H est la suivante :

$$\lim_{h \rightarrow 0} \inf_{v_h \in H_h} \|v - v_h\|_H = 0, \text{ pour tout } v \in H_m. \quad (2.2)$$

En effet, on a le résultat suivant :

Théorème 3 *Sous la condition (2.2), il existe h_0 et $C_0 > 0$ tels que, pour tout $h \leq h_0$,*

$$0 \leq \lambda_{m,h} - \lambda_m \leq C_0 \cdot \sup_{v \in H_m, \|v\|_L=1} \inf_{v_h \in H_h} \|v - v_h\|_H^2 \xrightarrow{h \rightarrow 0} 0.$$

Nous allons faire une esquisse de preuve ; pour une preuve complète, voir [41].

Considérons Π_h l'opérateur de projection orthogonale de H sur H_h associé au produit scalaire $\langle \cdot, \cdot \rangle_a = a(\cdot, \cdot)$, i.e. l'opérateur $\Pi_h : H \rightarrow H_h$ défini par

$$a(\Pi_h u - u, v) = 0, \text{ pour tous } v \in H_h \text{ et } u \in H.$$

Considérons encore

$$\sigma_{m,h} = \inf_{v \in H_m \setminus \{0\}} \frac{\|\Pi_h v\|_L}{\|v\|_L}.$$

On va utiliser le résultat suivant (sans preuve) :

Lemme 1 *Pour tout $m \leq N$, il existe $C = C(m) > 0$ tel que*

$$\sigma_{m,h}^2 \geq 1 - \sup_{v \in H_m, \|v\|_L=1} \|v - \Pi_h v\|_H^2.$$

Esquisse de preuve du théorème 3

1re étape : Par le théorème du Min-Max, on obtient immédiatement que, pour tout m ,

$$\lambda_{m,h} \geq \lambda_m. \quad (2.3)$$

2e étape : Par continuité et coercivité de a , on montre facilement que, pour tout $v \in H$,

$$\|v - \Pi_h v\|_H \leq \tilde{C} \inf_{v_h \in H_h} \|v - v_h\|_H. \quad (2.4)$$

Ainsi, par la condition (2.2), on trouve que, pour tout $v \in H_m$,

$$\lim_{h \rightarrow 0} \|v - \Pi_h v\|_H = 0. \quad (2.5)$$

3e étape : Dès que $\sigma_{m,h} > 0$ (ce qui est le cas pour h suffisamment petit), on a

$$\lambda_{m,h} \leq \frac{1}{\sigma_{m,h}^2} \lambda_m. \quad (2.6)$$

En effet, si $\sigma_{m,h} > 0$, alors $\Pi_h H_m$ est un sous-espace vectoriel de H de dimension m . Ainsi, par le théorème du Min-Max,

$$\lambda_{m,h} \leq \max_{v \in H_m \setminus \{0\}} QR(v) = \max_{v \in H_m \setminus \{0\}} \frac{a(\Pi_h v, \Pi_h v)}{\|\Pi_h v\|_L^2}.$$

Or, si $v \in H_m \setminus \{0\}$, $a(\Pi_h v, \Pi_h v) \leq a(v, v)$, donc

$$\lambda_{m,h} \leq \max_{v \in H_m \setminus \{0\}} \frac{a(v, v)}{\|\Pi_h v\|_L^2} = \max_{v \in H_m \setminus \{0\}} \frac{\|v\|_L^2}{\|\Pi_h v\|_L^2} \frac{a(v, v)}{\|v\|_L^2} \leq \frac{1}{\sigma_{m,h}^2} \lambda_m.$$

4e étape : Comme $\dim H_m = m < \infty$, alors par (2.5),

$$\sup_{v \in H_m, \|v\|_L=1} \|v - \Pi_h v\|_H^2 \xrightarrow{h \rightarrow 0} 0.$$

Il existe donc $h_0 > 0$ tel que, pour tout $h \leq h_0$,

$$C \sup_{v \in H_m, \|v\|_L=1} \|v - \Pi_h v\|_H^2 \leq 1/2,$$

avec C comme dans le lemme 1. Ainsi, on a pour tout $h \leq h_0$,

$$\begin{aligned} \lambda_{m,h} &\stackrel{(2.6)}{\leq} \frac{1}{\sigma_{m,h}^2} \lambda_m \stackrel{(\text{lemme 1})}{=} \frac{1}{1 - C \sup_{v \in H_m, \|v\|_L=1} \|v - \Pi_h v\|_H^2} \lambda_m \\ &\leq \left(1 + 2C \sup_{v \in H_m, \|v\|_L=1} \|v - \Pi_h v\|_H^2 \right) \lambda_m, \end{aligned}$$

d'où, en notant $C_0 = 2C\tilde{C}\lambda_m$,

$$\begin{aligned} 0 &\stackrel{(2.3)}{\leq} \lambda_{m,h} - \lambda_m \leq 2C \sup_{v \in H_m, \|v\|_L=1} \|v - \Pi_h v\|_H^2 \lambda_m \\ &\stackrel{(2.4)}{\leq} C_0 \cdot \sup_{v \in H_m, \|v\|_L=1} \inf_{v_h \in H_h} \|v - v_h\|_H^2 \xrightarrow{h \rightarrow 0} 0, \end{aligned}$$

par hypothèse (2.2). □

2.3 Eléments finis

2.3.1 Eléments finis de Lagrange

Considérons K une partie de \mathbb{R}^n compacte, connexe et d'intérieur non vide, $\Sigma = \{x_j\}_{j=1}^N$ un ensemble fini de points distincts de K , appelés *nœuds* (ou *sommets*), et P un espace vectoriel de fonctions de K dans \mathbb{R} , de dimension finie.

On dit que Σ est *P-unisolvant* lorsque, pour tout $(\alpha_1, \dots, \alpha_N) \in \mathbb{R}^N$ donné, il existe un unique $p \in P$ vérifiant, pour tout j , $p(x_j) = \alpha_j$. Remarquons qu'une condition nécessaire pour que Σ soit *P-unisolvant* est

$$\dim P = |\Sigma| = N.$$

Lorsque Σ est *P-unisolvant*, (K, P, Σ) est appelé *élément fini de Lagrange*.

Pour un élément fini de Lagrange (K, P, Σ) donné, on appelle *fonctions formes* les $p_i \in P$

($1 \leq i \leq N$) tels que $p_i(x_j) = \delta_{ij}$. Remarquons que les fonctions formes existent et sont uniques. De plus, elles forment une base de P .

Pour un élément de Lagrange (K, P, Σ) et une base de fonctions formes $(p_i)_{i=1}^N$ donnés, on définit l'opérateur de P -interpolation de Lagrange de la manière suivante :

$$\begin{aligned} \Pi : \mathbb{R}^K &\longrightarrow P \\ v &\longmapsto \Pi v = \sum_{i=1}^N v(x_i) p_i. \end{aligned}$$

L'élément Πv est appelé le P -interpolé de Lagrange de v sur Σ . On a en effet, pour tout j ,

$$\Pi v(x_j) = \sum_{i=1}^N v(x_i) p_i(x_j) = v(x_j).$$

Nous allons à présent voir une méthode qui permet, à partir d'un élément fini de Lagrange, de générer une famille d'éléments finis de Lagrange.

Soient $(\hat{K}, \hat{P}, \hat{\Sigma})$ un élément fini de Lagrange, K une partie de \mathbb{R}^n , compacte, connexe et d'intérieur non vide et $F : \hat{K} \rightarrow K$ une application bijective. Notons $F(\hat{x}) = x$. On définit $P = \{p \in \mathbb{R}^K : p \circ F \in \hat{P}\}$ et $\Sigma = F(\hat{\Sigma})$. On note enfin $\hat{p} = p \circ F$.

Proposition 1 *Sous les conditions ci-dessus, (K, P, Σ) est un élément fini de Lagrange. De plus, si $\hat{\Pi}$ est l'opérateur de \hat{P} -interpolation sur $\hat{\Sigma}$ et Π celui de P -interpolation sur Σ , alors on a, pour tout $v \in \mathbb{R}^K$,*

$$(\Pi v) \circ F = \Pi(v \circ F),$$

c'est-à-dire

$$\widehat{\Pi v} = \hat{\Pi} \hat{v}.$$

Deux tels éléments finis de Lagrange sont dits *équivalents*; si de plus F est affine, ils sont dits *affine-équivalents*.

Preuve (tirée de [41])

Il s'agit d'abord de voir que Σ est P -unisolvant. Soit donc $(\alpha_1, \dots, \alpha_N) \in \mathbb{R}^N$; on veut trouver $p \in P$ vérifiant $p(x_i) = \alpha_i$, pour tout i . Prenons $\hat{p} \in \hat{P}$ tel que $\hat{p}(\hat{x}_i) = \alpha_i$ pour tout i . Alors l'élément $p \in P$ défini par $p = \hat{p} \circ F^{-1}$ fait l'affaire!

De plus, si $p, q \in P$ vérifient $p(x_i) = q(x_i)$ pour tout i , alors les éléments \hat{p} et \hat{q} associés vérifient

$$\hat{p}(\hat{x}_i) = p(x_i) = q(x_i) = \hat{q}(\hat{x}_i),$$

pour tout i . Ainsi, par \hat{P} -unisolvance de $\hat{\Sigma}$, on trouve $\hat{p} = \hat{q}$, i.e. $p = q$.

Soit maintenant $\hat{x} \in \hat{K}$. On a

$$\begin{aligned} (\Pi v) \circ F(\hat{x}) &= \left(\sum_{i=1}^N v(x_i) p_i \right) (F(\hat{x})) = \sum_{i=1}^N v(x_i) p_i(F(\hat{x})) \\ &= \sum_{i=1}^N v(x_i) \hat{p}_i(\hat{x}) = \sum_{i=1}^N v(F(\hat{x}_i)) \hat{p}_i(\hat{x}) = \Pi(v \circ F)(\hat{x}). \end{aligned}$$

□

2.3.2 Éléments finis simpliciaux

Nous allons à présent construire une famille particulière d'éléments finis de Lagrange : les éléments finis simpliciaux.

Soit K un n -simplexe de \mathbb{R}^n : il s'agit de l'enveloppe convexe de $(n+1)$ points donnés de \mathbb{R}^n , notés $a_j = (a_{1j}, \dots, a_{nj})^t$, non situés dans un même hyperplan de \mathbb{R}^n , i.e. tels que la matrice

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1,n+1} \\ \vdots & & \vdots \\ a_{n,1} & \cdots & a_{n,n+1} \\ 1 & \cdots & 1 \end{pmatrix}$$

est inversible.

Remarque : pour $n = 2$, K est un triangle ; pour $n = 3$, il s'agit d'un tétraèdre.

Soit maintenant $x \in \mathbb{R}^n$. On définit, pour tout $j \in \{1, \dots, n+1\}$, $\varphi_j = \varphi_j(x)$ comme la solution du système suivant :

$$\begin{cases} \sum_{j=1}^{n+1} a_{ij} \varphi_j = x_i, \text{ pour tout } i \in \{1, \dots, n\} \\ \sum_{j=1}^{n+1} \varphi_j = 1, \end{cases}$$

c'est-à-dire, en notant $\varphi = (\varphi_j)_{j=1}^{n+1}$ et $\tilde{A} = (a_{ij})_{1 \leq i \leq n, 1 \leq j \leq n+1}$,

$$\begin{cases} \tilde{A} \varphi = x \\ \sum_{j=1}^{n+1} \varphi_j = 1. \end{cases}$$

Les $\varphi_j(x)$ sont appelées *coordonnées barycentriques* de x par rapport aux a_j . Les fonctions $\varphi_j : x \mapsto \varphi_j(x)$ sont les fonctions affines de \mathbb{R}^n dans \mathbb{R} .

On a ainsi

$$K = \{x \in \mathbb{R}^n : 0 \leq \varphi_j \leq 1, \text{ pour tout } j \in \{1, \dots, n+1\}\}.$$

Notons encore, pour $k \in \mathbb{N}$ donné, P_k l'espace vectoriel des polynômes de \mathbb{R}^n dans \mathbb{R} de degrés inférieurs ou égaux à k . On définit finalement le *treillis principal d'ordre k du n -simplexe K* , noté Σ_k , par

$$\Sigma_k = \left\{ x \in \mathbb{R}^n : \varphi_j(x) \in \left\{ 0, \dots, \frac{k-1}{k}, 1 \right\} \text{ pour tout } j \in \{1, \dots, n+1\} \right\}$$

si $k > 0$ et

$$\Sigma_0 = \left\{ x \in \mathbb{R}^n : \varphi_j(x) = \frac{1}{n+1} \text{ pour tout } j \in \{1, \dots, n+1\} \right\}.$$

On a alors le résultat suivant :

Théorème 4 *Pour tout $k \in \mathbb{N}$, (K, P_k, Σ_k) ainsi défini est un élément fini de Lagrange, que l'on appelle n -simplexe de type k . De plus, les n -simplexes de type k sont deux à deux affine-équivalents.*

Nous n'utiliserons que les n -simplexes de type 1, pour lesquels la preuve est triviale; remarquons d'ailleurs que, dans ce cas, les coordonnées barycentriques correspondent aux fonctions formes. Pour les n -simplexes de types $k \neq 1$, voir [41].

Ce résultat montre qu'il suffit d'étudier les n -simplexes dits de référence (ou standard) : ceux définis par les sommets $\hat{a}_j = (\delta_{ij})_{i=1}^n$ pour $j \leq n$ et $\hat{a}_{n+1} = (0, \dots, 0)$.

2.3.3 Méthode des éléments finis pour les domaines euclidiens

Le but est d'utiliser les éléments finis pour construire des sous-espaces de dimensions finies H_h de $H = V(\Omega)$ vérifiant la condition de convergence (2.2), que l'on rappelle ici :

$$\lim_{h \rightarrow 0} \inf_{v_h \in H_h} \|v - v_h\|_H = 0, \text{ pour tout } v \in H_m,$$

avec H_m l'espace engendré par les m premières solutions de (P_f) .

Supposons, pour éviter les difficultés techniques, que le domaine Ω de \mathbb{R}^n est polyédrique (voir [16] ou [41] pour le cas plus général).

Prenons tout d'abord une *triangulation* τ_h de $\bar{\Omega}$, avec $h = \max_{K \in \tau_h} h_K$ et h_K le diamètre de K , c'est-à-dire $\bar{\Omega} = \bigcup_{K \in \tau_h} K$, avec

- a) K polyèdres de \mathbb{R}^n , d'intérieurs non vides.
 b) Si $K_1 \neq K_2$ sont deux éléments de τ_h , alors $\overset{\circ}{K}_1 \cap \overset{\circ}{K}_2 = \phi$.
 c) Toute face de tout $K_1 \in \tau_h$ est soit une face d'un $K_2 \in \tau_h$ ($K_1 \neq K_2$), soit une partie de $\partial\Omega$.

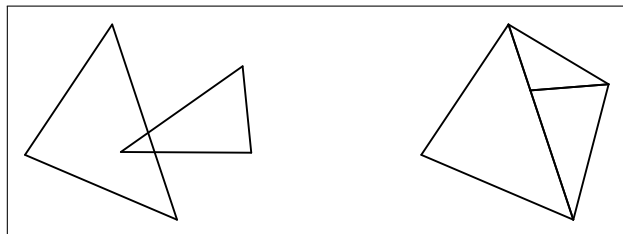
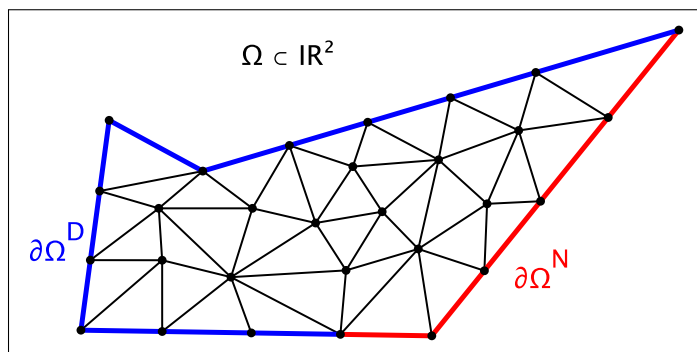


FIGURE 2.1 – Cas interdits.

Nous supposons de plus que cette triangulation respecte la partition $\partial\Omega = \partial\Omega^D \amalg \partial\Omega^N$ du bord de Ω , c'est-à-dire que la troisième condition est remplacée par

- c') Toute face de $K_1 \in \tau_h$ est soit une face d'un $K_2 \in \tau_h$ ($K_1 \neq K_2$), soit une partie de $\partial\Omega^D$, soit une partie de $\partial\Omega^N$.

FIGURE 2.2 – Triangulation τ_h de $\bar{\Omega}$.

Pour tout $K \in \tau_h$, considérons encore un élément fini de Lagrange (K, P_K, Σ_K) tel que $P_K \subset H^1(K)$. On définit alors

$$H_h = \{v \in \mathcal{C}^0(\bar{\Omega}) : v|_K \in P_K \text{ et } v|_{\partial\Omega^D} = 0\} \subset H.$$

La suite consiste à mettre de bonnes conditions de compatibilité et de régularité sur les éléments finis considérés (les simplexes) afin que la condition de convergence (2.2) soit vérifiée.

Introduisons l'opérateur $\Pi_h : \mathcal{C}^0(\bar{\Omega}) \rightarrow L^2(\bar{\Omega})$ défini par $\Pi_h v(x) = \Pi_K v(x)$, pour tous $v \in \mathcal{C}^0(\bar{\Omega})$, $K \in \tau_h$ et $x \in \overset{\circ}{K}$, où Π_K est l'opérateur de P_K -interpolation sur Σ_K .

Pour que $H_h = \Pi_h(\{v \in \mathcal{C}^0(\bar{\Omega}) : v|_{\partial\Omega^D} = 0\})$, nous avons besoin de deux types de conditions :

Compatibilité : soient $K_1, K_2 \in \tau_h$ adjacents, de face commune $\Gamma = K_1 \cap K_2$. Alors

- a) $P_{K_1}|_{\Gamma} = P_{K_2}|_{\Gamma}$ (compatibilité des fonctions).
- b) $\Sigma_{K_1} \cap \Gamma = \Sigma_{K_2} \cap \Gamma$ (compatibilité des points).

Régularité : pour tout $K \in \tau_h$, l'élément fini (K, P_K, Σ_K) est de classe \mathcal{C}^0 , i.e.

- a) $P_K \subset \mathcal{C}^0(K)$.
- b) Pour toute face Γ de K , $\Sigma' = \Sigma \cap \Gamma$ est P' -unisolvant, où $P' = \{p|_{\Gamma} : p \in P\}$.

Remarquons que tout élément fini affine-équivalent à un élément fini de classe \mathcal{C}^0 l'est encore.

Proposition 2 Soit τ_h une triangulation de $\bar{\Omega}$ construite à l'aide de simplexes. Pour $k \geq 1$ donné, on associe à tout $K \in \tau_h$ le simplexe de type k construit sur K . Alors les conditions ci-dessus sont vérifiées.

Finalement, pour assurer la condition de convergence (2.2) de H_h vers H , nous exigerons que la famille de triangulations (τ_h) de $\bar{\Omega}$ soit *régulière*, i.e. qu'elle satisfasse les conditions suivantes :

- a) Tous les éléments finis (K, P_K, Σ_K) de toutes les triangulations τ_h sont affine-équivalents à un élément fini de référence $(\hat{K}, \hat{P}, \hat{\Sigma})$ de classe \mathcal{C}^0 .
- b) Pour tout couple (Γ_1, Γ_2) de faces de \hat{K} et pour toute application \hat{F} affine inversible de $\mathbb{R}^n \rightarrow \mathbb{R}^n$ telle que $\Gamma_2 = \hat{F}(\Gamma_1)$, on a

$$\hat{\Sigma} \cap \Gamma_2 = \hat{F}(\hat{\Sigma} \cap \Gamma_1) \quad \text{et} \quad \{\hat{p}|_{\Gamma_2} : \hat{p} \in \hat{P}\} = \{\hat{p} \circ \hat{F}|_{\Gamma_1} : \hat{p} \in \hat{P}\};$$

- c) Les diamètres h_K des éléments vérifient $\lim_{h \rightarrow 0} \left(\max_{K \in \tau_h} h_K \right) = 0$.

- d) Il existe une constante $\sigma \geq 1$ telle que, pour tout h et pour tout $K \in \tau_h$, on a

$$h_K / \rho_K \leq \sigma,$$

avec ρ_K la rondeur de K , c'est-à-dire le maximum des diamètres des boules contenues dans K .

Remarquons que les deux premières conditions, relatives aux types d'éléments finis utilisés, assurent les conditions de compatibilités vues précédemment ; les deux conditions suivantes portent quand à elles sur la géométrie des $K \in \tau_h$. Cette notion de régularité permet d'énoncer le résultat de convergence suivant, démontré dans [41].

Théorème 5 *Soient Ω un ouvert polyédrique de \mathbb{R}^n , $n \leq 3$, et (τ_h) une famille régulière de triangulations de $\bar{\Omega}$ associées à un élément fini de référence $(\hat{K}, \hat{P}, \hat{\Sigma})$ de classe \mathcal{C}^0 et respectant la partition $\partial\Omega = \partial\Omega^D \amalg \partial\Omega^N$. Supposons encore qu'il existe $k \geq 1$ tel que*

$$P_k \subset \hat{P} \subset H^1(\hat{K}).$$

Alors, si $u \in H^{k+1}(\Omega)$, on a

$$\|u - \Pi_h u\|_{H^1(\Omega)} \leq C_1 h^k \left(\sum_{|\alpha|=k+1} \|\partial^\alpha u\|_{L^2(\Omega)} \right)^{1/2} \leq C_1 h^k \|u\|_{H^{k+1}(\Omega)},$$

avec C_1 une constante indépendante de h .

Il s'en suit alors le résultat final suivant :

Théorème 6 *Soit (τ_h) une famille régulière de triangulations de $\bar{\Omega}$ associées à un n -simplexe de type $k \geq 1$, $n \leq 3$, et respectant la partition $\partial\Omega = \partial\Omega^D \amalg \partial\Omega^N$. Alors, si $u \in H^{l+1}(\Omega)$ avec $1 \leq l \leq k$, il existe une constante C_1 telle que*

$$\|u - \Pi_h u\|_{H^1(\Omega)} \leq C_1 h^l \|u\|_{H^{l+1}(\Omega)}.$$

De plus, si $H_m \subset H^{l+1}(\Omega)$ avec $1 \leq l \leq k$ et H_m l'espace engendré par les m premières solutions de (P_f) , il existe des constantes h_0 et $C_2 > 0$ tels que, pour tout $h \leq h_0$,

$$0 \leq \lambda_{m,h} - \lambda_m \leq C_2 h^l \sup_{v \in H_m, \|v\|_{L^2(\Omega)}=1} \|v\|_{H^{l+1}(\Omega)}.$$

Il est en effet facile de vérifier qu'une telle famille de triangulations vérifie les hypothèses du théorème 5, d'où la première inégalité. On a ainsi construit des sous-espaces de dimensions finies H_h de $H = V(\Omega)$ vérifiant la condition de convergence (2.2) ; la deuxième inégalité découle donc immédiatement du théorème 3.

Remarquons que la condition $H_m \subset H^{l+1}(\Omega)$ avec $l \geq 1$ dépend de la géométrie de Ω . Par exemple, si $\partial\Omega$ est \mathcal{C}^1 par morceaux sans cusp, alors $H_m \subset H^2(\Omega)$ (voir [23]) et $l = 1$ fait l'affaire.

2.3.4 Méthode des éléments finis pour les domaines de coques

Nous allons maintenant reprendre la méthode utilisée pour les domaines euclidiens et l'adapter aux domaines de coques (c'est-à-dire aux surfaces pouvant être paramétrées via une seule carte), en travaillant dans une carte. Plus précisément, notons (U, f) une carte d'une coque S telle que $\bar{\Omega} \subset f(U)$. Il s'agit de travailler sur $D = f^{-1}(\Omega)$ comme précédemment, en tenant compte de la métrique G de la surface S , c'est-à-dire de la première forme fondamentale de S associée à (U, f) .

Un rapide calcul en coordonnées donne

$$\int_{\Omega} \langle \nabla u, \nabla v \rangle = \sum_{i,j=1,2} \int_D \sqrt{\det G} g^{ij} \frac{\partial \tilde{u}}{\partial x_i} \frac{\partial \tilde{v}}{\partial x_j} dx,$$

$$\int_{\Omega} uv = \int_D \sqrt{\det G} \tilde{u} \tilde{v} dx,$$

avec g^{ij} le $(ij)^e$ coefficient de l'inverse de G , $\tilde{u} = u \circ f$ et $\tilde{v} = v \circ f$.

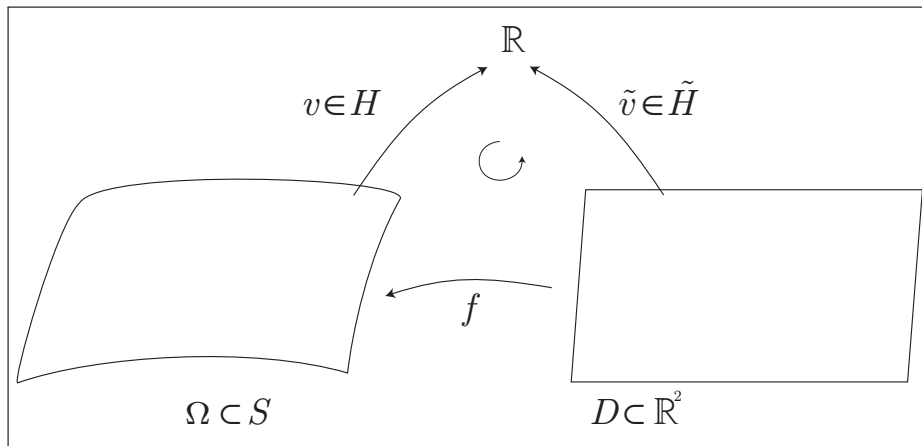


FIGURE 2.3 – Méthode des éléments finis pour les domaines de coques.

Ainsi, le problème (P_f) est équivalent au problème

$$(\tilde{P}_f) \quad \begin{cases} \text{Trouver } \tilde{u} \in \tilde{H} \text{ et } \lambda \in \mathbb{R} \text{ tels que, pour tout } \tilde{v} \in \tilde{H} : \\ \tilde{a}(\tilde{u}, \tilde{v}) = \lambda \tilde{b}(\tilde{u}, \tilde{v}) , \end{cases}$$

avec

$$\tilde{H} = \{ \tilde{v} = v \circ f : v \in H \} ,$$

$$\tilde{a}(\tilde{u}, \tilde{v}) = \int_{\Omega} \langle \nabla u, \nabla v \rangle = \sum_{i,j=1,2} \int_D \sqrt{\det G} g^{ij} \frac{\partial \tilde{u}}{\partial x_i} \frac{\partial \tilde{v}}{\partial x_j} dx ,$$

$$\tilde{b}(\tilde{u}, \tilde{v}) = \int_{\Omega} uv = \int_D \sqrt{\det G} \tilde{u} \tilde{v} dx .$$

Supposons comme précédemment, pour éviter les difficultés techniques, que le domaine D est polygonal. On considère alors une triangulation τ_h de \bar{D} respectant la partition $\partial D = \partial D^D \amalg \partial D^N$ associée à la partition de $\partial\Omega = \partial\Omega^D \amalg \partial\Omega^N$ et satisfaisant les mêmes critères que précédemment (compatibilité et régularité) mais pour D . On définit alors

$$\tilde{H}_h = \{v \in \mathcal{C}^0(\bar{D}) : v|_K \in P_K \text{ et } v|_{\partial D^D} = 0\} \subset \tilde{H}.$$

Alors, par les théorèmes 3 et 6, les solutions du problème

$$(\tilde{P}_{f,h}) \quad \begin{cases} \text{Trouver } \tilde{u}_h \in \tilde{H}_h \text{ et } \lambda_h \in \mathbb{R} \text{ tels que, pour tout } \tilde{v}_h \in \tilde{H}_h : \\ \tilde{a}(\tilde{u}_h, \tilde{v}_h) = \lambda_h \tilde{b}(\tilde{u}_h, \tilde{v}_h) \end{cases}$$

convergent vers celles de (\tilde{P}_f) de la même manière que pour les domaines euclidiens.

Remarquons pour terminer que, en définissant

$$H_h = \{v \in \mathbb{R}^\Omega : v \circ f \in \tilde{H}_h\} \subset H,$$

alors on trouve également que H_h converge vers H au sens de (2.2), donc les solutions de $(P_{f,h})$ convergent vers celles de (P_f) , toujours de la même manière. En particulier, on peut énoncer le résultat suivant, pendant du théorème 6 pour les domaines de coques :

Théorème 7 *Soient (U, f) une carte d'une coque S telle que $\bar{\Omega} \subset f(U)$ et $D = f^{-1}(\Omega)$. Soit (τ_h) une famille régulière de triangulations de \bar{D} associées à un n -simplexe de type $k \geq 1$, $n \leq 3$, et respectant la partition $\partial D = \partial D^D \amalg \partial D^N$ (associée à $\partial\Omega = \partial\Omega^D \amalg \partial\Omega^N$). Alors, si $u \in H^{l+1}(\Omega)$ avec $1 \leq l \leq k$, il existe une constante C_1 telle que*

$$\|u - \Pi_h u\|_{H^1(\Omega)} \leq C_1 h^l \|u\|_{H^{l+1}(\Omega)}.$$

De plus, si $H_m \subset H^{l+1}(\Omega)$ avec $1 \leq l \leq k$ et H_m l'espace engendré par les m premières solutions de (P_f) , il existe des constantes h_0 et $C_2 > 0$ tels que, pour tout $h \leq h_0$,

$$0 \leq \lambda_{m,h} - \lambda_m \leq C_2 h^l \sup_{v \in H_m, \|v\|_{L^2(\Omega)}=1} \|v\|_{H^{l+1}(\Omega)}.$$

Cette méthode est donc une généralisation de la méthode des éléments finis pour les domaines du plan. En effet, il suffit de prendre $U = \mathbb{R}^2$ et f la fonction identité. Dans ce cas, $D = \Omega$ et G est la métrique euclidienne.

De plus, comme pour les domaines du plan, la condition $H_m \subset H^{l+1}(\Omega)$ avec $l \geq 1$ est satisfaite si par exemple $\partial\Omega$ (et donc ∂D) est \mathcal{C}^1 par morceaux sans cusp, en prenant $l = 1$ (voir [23]).

2.3.5 Méthode des éléments finis pour les domaines de surfaces

Il s'agit une nouvelle fois de prendre la méthode utilisée précédemment, pour les domaines euclidiens puis pour les coques, mais en considérant cette fois-ci le domaine Ω comme une réunion finie de coques (non vides) Ω_α d'intérieurs disjoints, c'est-à-dire telles que

$$\bar{\Omega} = \bigcup_{\alpha=1}^A \bar{\Omega}_\alpha, \quad \Omega_\alpha \cap \Omega_\beta = \emptyset \text{ pour tous } \alpha \neq \beta \text{ et } \bar{\Omega}_\alpha \subset f_\alpha(U_\alpha) \text{ pour tout } \alpha, \text{ où } \{(U_\alpha, f_\alpha)\}_{\alpha=1}^A$$

est un atlas de S . Notons encore $D_\alpha = f_\alpha^{-1}(\Omega_\alpha)$, pour tout α . On considère alors Ω comme un ensemble de domaines $\{\Omega_\alpha\}$ (ou $\{D_\alpha\}$) que l'on identifie le long de leurs bords.

Définissons la relation d'équivalence \mathcal{R} suivante sur la réunion disjointe $\amalg D_\alpha$ des D_α : $P \in D_\alpha$ est en relation avec $Q \in D_\beta$, noté $P\mathcal{R}Q$, lorsque $f_\alpha(P) = f_\beta(Q)$. Autrement dit, deux points de l'atlas sont équivalents lorsqu'ils définissent le même point sur la surface.

Notons encore, pour tout α ,

$$\begin{aligned} H_\alpha &= \{v|_{\Omega_\alpha} : v \in H\}, \\ \tilde{H}_\alpha &= \{v_\alpha \circ f_\alpha : v_\alpha \in H_\alpha\}. \end{aligned}$$

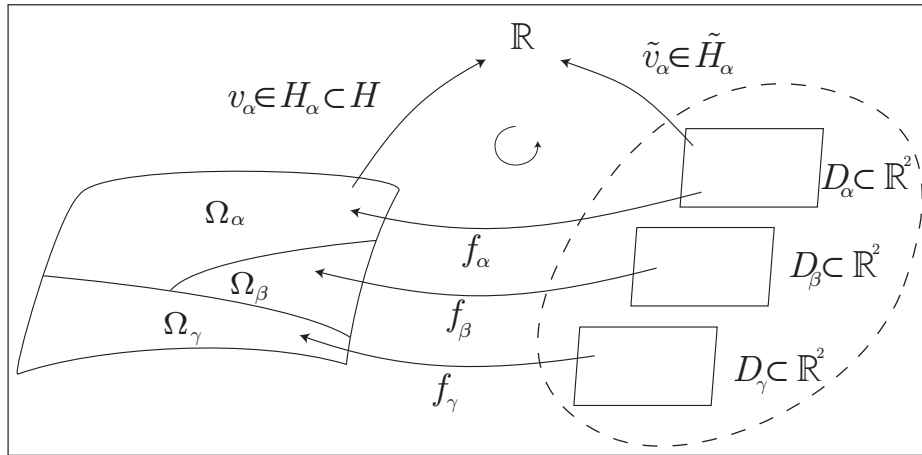


FIGURE 2.4 – Méthode des éléments finis pour les domaines de surfaces.

Alors (P_f) est équivalent au problème

$$(\tilde{P}_f) \quad \begin{cases} \text{Trouver } \tilde{u} \in \tilde{H} \text{ et } \lambda \in \mathbb{R} \text{ tels que, pour tout } \tilde{v} \in \tilde{H} \text{ et pour tout } \alpha : \\ \tilde{a}_\alpha(\tilde{u}|_{D_\alpha}, \tilde{v}|_{D_\alpha}) = \lambda \tilde{b}_\alpha(\tilde{u}|_{D_\alpha}, \tilde{v}|_{D_\alpha}), \end{cases}$$

avec

$$\begin{aligned}\tilde{H} &= \left\{ \tilde{v} \in \mathbb{R}^{\text{IID}\alpha} : \tilde{v}|_{D_\alpha} \in \tilde{H}_\alpha \text{ et } \tilde{v}(P) = \tilde{v}(Q) \text{ si } PRQ \right\}, \\ \tilde{a}_\alpha(\tilde{u}_\alpha, \tilde{v}_\alpha) &= \sum_{i,j=1,2} \int_{D_\alpha} \sqrt{\det G_\alpha} g_\alpha^{ij} \frac{\partial \tilde{u}_\alpha}{\partial x_i} \frac{\partial \tilde{v}_\alpha}{\partial x_j} dx, \\ \tilde{b}_\alpha(\tilde{u}_\alpha, \tilde{v}_\alpha) &= \int_{D_\alpha} \sqrt{\det G_\alpha} \tilde{u}_\alpha \tilde{v}_\alpha dx.\end{aligned}$$

On considère alors des triangulations $\tau_{h,\alpha}$ des domaines $\overline{D_\alpha} = \overline{f_\alpha^{-1}(\Omega_\alpha)}$ respectant les partitions $\partial D_\alpha = \partial D_\alpha^D \amalg \partial D_\alpha^N \amalg \partial D_\alpha^0$ associée à la partition de $\partial\Omega = \partial\Omega^D \amalg \partial\Omega^N$ (∂D_α^0 dénotant la partie de ∂D_α ne correspondant à aucune partie de $\partial\Omega$ et sur laquelle le recollement aux différentes cartes a lieu) et satisfaisant les critères usuels de compatibilité et de régularité pour les D_α . On exige de plus une *inter-compatibilité* en α des différentes triangulations : pour h fixé, si P est un nœud de $\overline{D_\alpha}$, alors $f_\beta^{-1} \circ f_\alpha(P)$ est un nœud de $\overline{D_\beta}$, pour tout β tel que $f_\alpha(P) \in f_\beta(\overline{D_\beta})$. Autrement dit, chaque point de la classe d'équivalence d'un nœud est lui-même un nœud (pour h fixé).

On définit alors

$$\begin{aligned}\tilde{H}_{\alpha,h} &= \{v \in \mathcal{C}^0(\overline{D_\alpha}) : v|_K \in P_K \text{ et } v|_{\partial D_\alpha^D} = 0\} \subset \tilde{H}_\alpha, \\ \tilde{H}_h &= \left\{ v \in \mathbb{R}^{\text{IID}\alpha} : v|_{\overline{D_\alpha}} \in \tilde{H}_{\alpha,h} \right\} \subset \tilde{H}, \\ \tilde{a}(\tilde{u}, \tilde{v}) &= \sum_{\alpha=1}^A \tilde{a}_\alpha(\tilde{u}|_{D_\alpha}, \tilde{v}|_{D_\alpha}), \\ \tilde{b}(\tilde{u}, \tilde{v}) &= \sum_{\alpha=1}^A \tilde{b}_\alpha(\tilde{u}|_{D_\alpha}, \tilde{v}|_{D_\alpha}).\end{aligned}$$

Alors, à nouveau par les théorèmes 3 et 6, les solutions du problème

$$(\tilde{P}_{f,h}) \quad \begin{cases} \text{Trouver } \tilde{u}_h \in \tilde{H}_h \text{ et } \lambda_h \in \mathbb{R} \text{ tels que, pour tout } \tilde{v}_h \in \tilde{H}_h : \\ \tilde{a}(\tilde{u}_h, \tilde{v}_h) = \lambda_h \tilde{b}(\tilde{u}_h, \tilde{v}_h) \end{cases}$$

convergent vers celles de (\tilde{P}_f) de la même manière que précédemment.

Remarquons pour terminer que, en définissant

$$H_h = \left\{ v \in \mathbb{R}^\Omega : v \circ f_\alpha \in \tilde{H}_{\alpha,h} \text{ pour tout } \alpha \right\} \subset H,$$

alors on trouve également que H_h converge vers H au sens de (2.2), donc les solutions de $(P_{f,h})$ convergent vers celles de (P_f) , toujours de la même manière. En particulier, on peut énoncer le résultat suivant, pendant du théorème 6 pour les domaines de surfaces :

Théorème 8 Soit $\{(U_\alpha, f_\alpha)\}_{\alpha=1}^A$ un atlas d'une surface S tel qu'il existe des coques Ω_α d'intérieurs disjoints vérifiant $\overline{\Omega_\alpha} \subset f_\alpha(U_\alpha)$ et $\bar{\Omega} = \bigcup_{\alpha=1}^A \overline{\Omega_\alpha}$. Notons alors, pour tout α , $D_\alpha = f_\alpha^{-1}(\Omega_\alpha)$. Soient $(\tau_{h,\alpha})$ des familles régulières et inter-compatibles en α de triangulations des $\overline{D_\alpha}$ associées à un n -simplexe de type $k \geq 1$, $n \leq 3$, et respectant les partitions $\partial D_\alpha = \partial D_\alpha^D \amalg \partial D_\alpha^N \amalg \partial D_\alpha^0$ (associées à $\partial\Omega = \partial\Omega^D \amalg \partial\Omega^N$). Alors, si $u \in H^{l+1}(\Omega)$ avec $1 \leq l \leq k$, il existe une constante C_1 telle que

$$\|u - \Pi_h u\|_{H^1(\Omega)} \leq C_1 h^l \|u\|_{H^{l+1}(\Omega)} .$$

De plus, si $H_m \subset H^{l+1}(\Omega)$ avec $1 \leq l \leq k$ et H_m l'espace engendré par les m premières solutions de (P_f) , il existe des constantes h_0 et $C_2 > 0$ tels que, pour tout $h \leq h_0$,

$$0 \leq \lambda_{m,h} - \lambda_m \leq C_2 h^l \sup_{v \in H_m, \|v\|_{L^2(\Omega)}=1} \|v\|_{H^{l+1}(\Omega)} .$$

Cette méthode est donc une généralisation de la méthode des éléments finis pour les domaines de coques et donc pour ceux du plan. En effet, il suffit de prendre $A = 1$, c'est-à-dire un atlas formé d'une seule carte, pour se ramener au cas d'un domaine de coque, puis $U = \mathbb{R}^2$ et f la fonction identité pour le cas d'un domaine du plan.

Et à nouveau, comme pour les domaines du plan, la condition $H_m \subset H^{l+1}(\Omega)$ avec $l \geq 1$ est satisfaite si par exemple les ∂D_α sont \mathcal{C}^1 par morceaux sans cusp, en prenant $l = 1$ (voir [23]).

Chapitre 3

Résolution numérique

Dans le chapitre précédent, nous avons étudié un algorithme permettant d'estimer le spectre du Laplacien. Il s'agit à présent de préciser la mise en œuvre numérique de la méthode.

Dans un premier temps seront abordées les façons de définir numériquement des domaines ainsi que les procédures de construction de triangulations de domaines, appelés *maillages* (et souvent confondus, par abus de langage, avec les triangulations en résultant), via un *mailleur*, soit un algorithme numérique plus ou moins automatique de maillage. C'est dans ce but qu'un mailleur classique, dit *par macro-éléments*, sera considéré. Le principe de base de cette méthode est de subdiviser le domaine à mailler en sous-domaines, les *macro-éléments*, dont on peut facilement créer une triangulation. Ce mailleur a l'avantage d'être rapide, simple et efficace pour des domaines peu complexes. Il peut par contre générer de grandes largeurs de bande (c'est-à-dire un grand nombre de diagonales non-nulles) des futures matrices associées au problème considéré, notamment dans le cas où on identifie des faces (domaines de surfaces). Or, les largeurs de bande des matrices tiennent un rôle important dans la résolution numérique, tant au niveau de la mémoire nécessaire que dans le temps de calcul effectif; il est donc nécessaire de réduire les largeurs de bande au maximum. De plus, avec ce mailleur, la grandeur des mailles est difficilement ajustable localement. Enfin, la découpe manuelle du domaine en macro-éléments peut rendre difficile certaines automatisations et n'est pas adaptée à certains domaines complexes.

Pour pallier à ces inconvénients, des algorithmes de traitement de triangulation, en l'occurrence de renumérotation des nœuds et de raffinement, s'imposent, de même qu'une méthode plus flexible de maillage. Nous considérerons dans ce but un mailleur basé sur la méthode de Voronoï-Delaunay, qui ne demande à l'utilisateur que de fournir le bord du domaine à mailler (par exemple via une paramétrisation ou une liste de nœuds).

Dans un deuxième temps sera abordée la résolution numérique du problème aux valeurs propre lui-même, à savoir la mise en œuvre du calcul matriciel (construction des matrices et intégration numérique) ainsi que la résolution numérique de problèmes matriciels aux valeurs propres via l'algorithme de Lanczos.

3.1 Maillage par macro-éléments

Le mailleur classique utilisé dans ce travail, dit par macro-éléments, demande au préalable à l'utilisateur de décomposer manuellement le domaine à mailler en sous-domaines, les macro-éléments, de types triangle ou quadrilatère, avec éventuellement une face courbe dans ce dernier cas. Chaque macro-élément est ensuite maillé grâce à une grille, adaptée selon le type de macro-élément. Ces différentes triangulations sont ensuite assemblées pour constituer une triangulation du domaine complet.

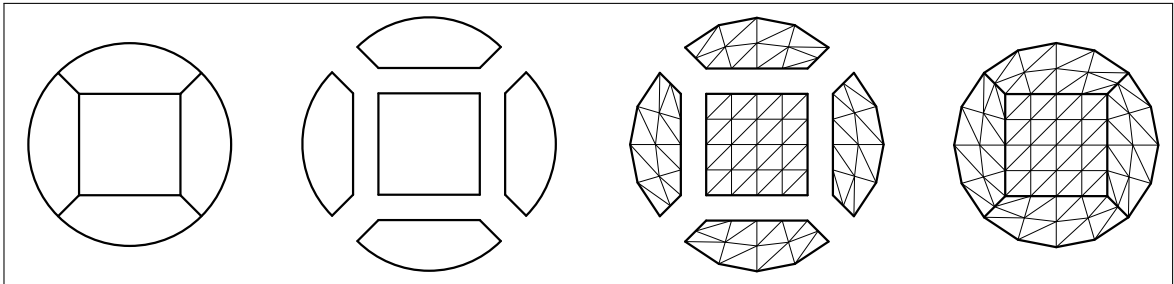


FIGURE 3.1 – Exemple de maillage d'un disque par décomposition en 5 macro-éléments.

Concrètement, il s'agit pour l'utilisateur de donner, pour chaque macro-élément, le type de macro-élément, les coordonnées des sommets, une paramétrisation de l'éventuelle face courbe, le nombre de segments de la grille et les éventuelles conditions au bord en vue de la résolution du problème. Dans le cas de domaine non euclidien, le numéro de carte du macro-élément et ses éventuelles fonctions d'identification des faces aux autres macro-éléments sont nécessaires afin de construire les classes d'équivalence des nœuds.

En retour, le mailleur calculera une liste de nœuds contenant essentiellement leurs coordonnées et leur éventuelle condition au bord. Dans le cas de domaine non euclidien, elle contiendra également leur numéro de carte ainsi que celui de leur classe d'équivalence. Le mailleur calculera également une liste des éléments contenant essentiellement les numéros de leurs sommets et les éventuelles conditions au bords de leurs faces.

3.1.1 Macro-éléments de type quadrilatère

Le maillage d'un quadrilatère consiste essentiellement en une décomposition structurée en plus petits quadrilatères, eux-même découpés en triangles. On donne le quadrilatère par ses sommets P_1 , P_2 , P_3 et P_4 et on indique le nombre n_x de subdivisions des segments P_1P_2 et P_3P_4 , ainsi que le nombre n_y de subdivisions des segments P_2P_3 et P_4P_1 . Les nœuds $P_{s,t}$ du macro-éléments sont alors calculés successivement, en faisant varier entre 0 et 1 les paramètres s en $n_x + 1$ valeurs et t en $n_y + 1$ valeurs, de la manière suivante :

$$P_{s,t} = A_t + s(B_t - A_t),$$

avec

$$A_t = (1-t)P_1 + tP_4 \text{ et } B_t = (1-t)P_2 + tP_3.$$

On trouve ainsi

$$P_{s,t} = (1-s)(1-t)P_1 + s(1-t)P_2 + stP_3 + (1-s)tP_4 = \sum_{i=1}^4 f_i(s,t)P_i,$$

où les f_i sont les *fonctions formes* aux nœuds P_i , c'est-à-dire les fonctions

$$f_1(s,t) = (1-s)(1-t),$$

$$f_2(s,t) = s(1-t),$$

$$f_3(s,t) = st,$$

$$f_4(s,t) = (1-s)t.$$

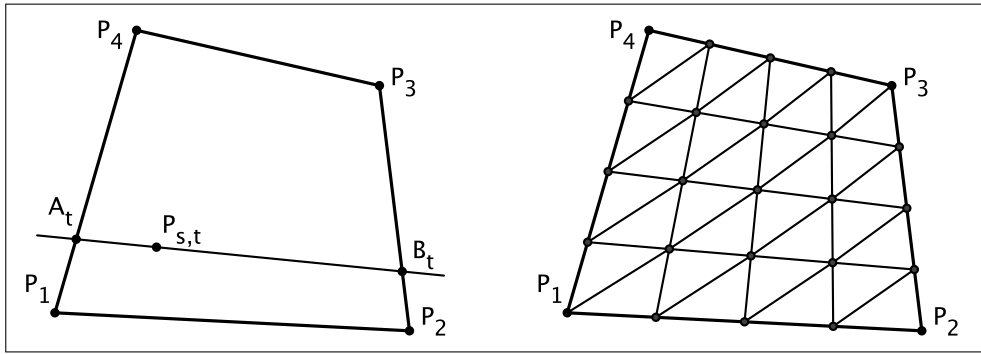


FIGURE 3.2 – Maillage d'un quadrilatère.

Simultanément aux nœuds du macro-élément, on construit ses éléments de la manière suivante : pour s et t fixés, on définit les éléments $K_{1,s,t}$ et $K_{2,s,t}$ par leurs sommets $\{P_{s,t}, P_{s+1,t}, P_{s+1,t+1}\}$ et $\{P_{s,t}, P_{s+1,t+1}, P_{s,t+1}\}$.

3.1.2 Macro-éléments de type quadrilatères à une face courbe

Considérons à présent un quadrilatère ayant une face courbe et trois faces rectilignes. Nous pouvons supposer que la face courbe est la deuxième du quadrilatère, c'est-à-dire le segment P_2P_3 , et qu'elle est donnée par une courbe paramétrée γ , idéalement par longueur d'arc, de paramètre variant entre a et b . En particulier, on doit avoir $\gamma(a) = P_2$ et $\gamma(b) = P_3$. Pour reprendre les notations précédentes, il s'agit de remplacer l'expression de B_t par $\gamma(ta + (1-t)b)$.

On trouve ainsi

$$\begin{aligned} P_{s,t} &= (1-s)(1-t)P_1 + (1-s)tP_4 + s\gamma(ta + (1-t)b) \\ &= f_1(s,t)P_1 + f_4(s,t)P_4 + s\gamma(ta + (1-t)b). \end{aligned}$$

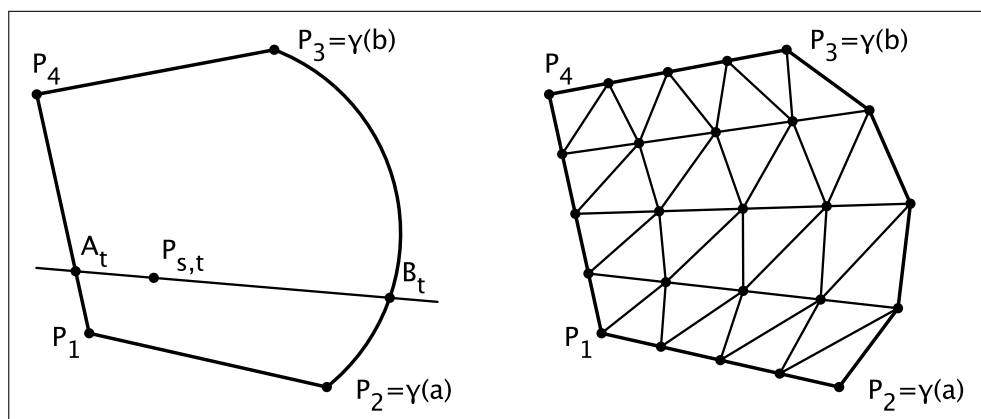


FIGURE 3.3 – Maillage d'un quadrilatère ayant une face courbe.

Les éléments sont construits simultanément aux nœuds, de manière identique au cas des quadrilatères sans face courbe.

3.1.3 Macro-éléments de type triangle

On donne un triangle par ses sommets P_1 , P_2 , P_3 et on indique le nombre n_x de subdivisions du segment P_1P_2 , ainsi que le nombre n_y de subdivisions des segments P_2P_3 et P_3P_1 , avec la condition que n_x soit **inférieur ou égal** à n_y .

L'idée est de reprendre le maillage de quadrilatères, en faisant correspondre au triangle le parallélogramme associé de sommets P_1 , P_2 , P_1' et P_3 (où P_1' est l'image de P_1 par la symétrie dont le centre est le point milieu de P_2P_3), puis de déplacer les nœuds proches de l'arête P_2P_3 afin qu'ils se retrouvent dessus (voir figure 3.4).

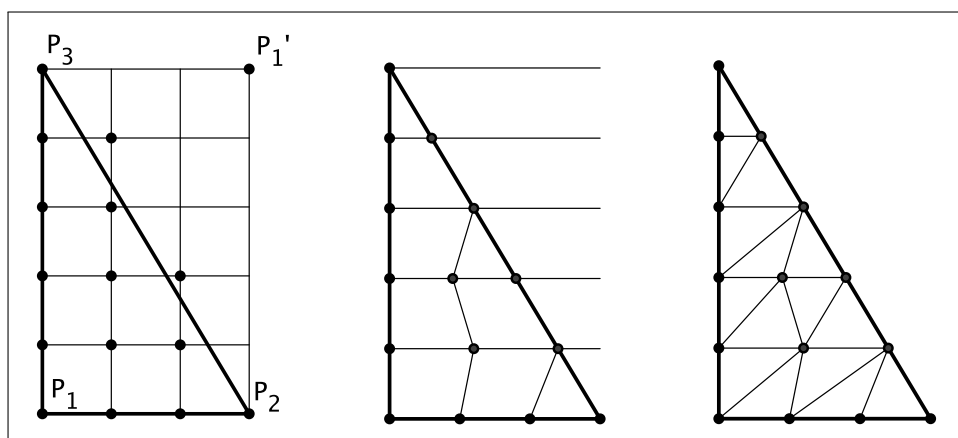


FIGURE 3.4 – Maillage d'un triangle par correspondance avec le parallélogramme associé.

En détails, on fixe pour chaque valeur du paramètre t pris entre 0 et 1 (en $n_y + 1$ valeurs) un nombre $n_x^l = n_x^l(t)$ d'intervalles à la ligne définie par t . Autrement dit, pour t donné, le paramètre s prendra $n_x^l(t) + 1$ valeurs. Ce nombre, variant entre 0 (lorsque $t = 1$) et n_x (lorsque $t = 0$), est donné par

$$n_x^l = \begin{cases} \max \{1, [(1-t)n_x]\} & \text{si } t \neq 1 \\ 0 & \text{si } t = 1 \end{cases}$$

où $[\cdot]$ désigne la partie entière.

Les nœuds $P_{s,t}$ du macro-éléments sont alors calculés successivement, en faisant varier entre 0 et 1 les paramètres t en $n_y + 1$ valeurs et s en $n_x^l + 1$ valeurs, de la manière suivante :

$$P_{s,t} = A_t + s(B_t - A_t),$$

avec

$$A_t = (1-t)P_1 + tP_3 \text{ et } B_t = (1-t)P_2 + tP_3.$$

On trouve ainsi

$$P_{s,t} = (1-t)(1-s)P_1 + (1-t)sP_2 + tP_3 = f_1(s,t)P_1 + f_2(s,t)P_2 + tP_3,$$

où les f_i sont les fonctions formes aux sommets P_i définis plus haut.

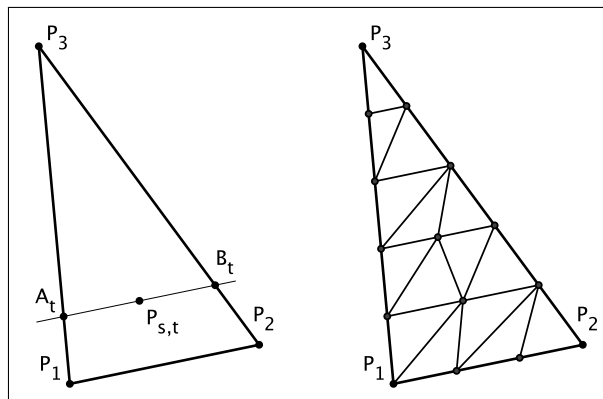


FIGURE 3.5 – Maillage d'un triangle.

A nouveau, les éléments sont construits simultanément aux nœuds du macro-élément, de manière semblable aux quadrilatères.

3.1.4 Généralisation à des macro-éléments ayant plusieurs faces courbes

Même si cela n'a pas été utilisé dans ce travail, il peut être agréable d'utiliser des macro-éléments ayant plusieurs faces courbes.

Pour les quadrilatères, on suppose que les faces courbes sont données par des courbes paramétrées γ_i de paramètre variant, pour simplifier, entre 0 et 1. En particulier, on doit avoir $\gamma_i(0) = P_i$ pour i entre 1 et 4, $\gamma_i(1) = P_{i+1}$ pour i entre 1 et 3 ainsi que $\gamma_4(1) = P_1$. On trouve alors, en suivant une logique comparable au cas précédents,

$$P_{s,t} = (1-t)\gamma_1(s) + s\gamma_2(t) + t\gamma_3(1-s) + (1-s)\gamma_4(1-t) \\ - \left((1-s)(1-t)P_1 + s(1-t)P_2 + stP_3 + (1-s)tP_4 \right).$$

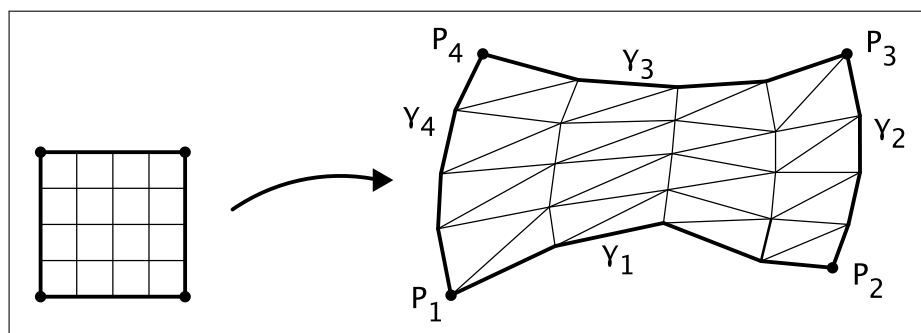


FIGURE 3.6 – Maillage d'un quadrilatère à faces courbes.

Il est également possible de trouver une formule équivalente pour les triangles; voir par exemple [22].

3.2 Traitements de triangulation

Les procédures suivantes de traitement de triangulation, à savoir la renumérotation et le raffinement, permettent d'adapter une triangulation donnée, indépendamment du mailler dont il est issu. La renumérotation des noeuds d'une triangulation est effectuée de manière à diminuer la différence de numéros de noeuds voisins, dans le but de réduire la largeur de bande des futures matrices du problème considéré. Le raffinement permet quant à lui d'obtenir une triangulation plus fine qu'initialement, soit localement, soit de manière globale. Le but d'une telle procédure est évidemment d'améliorer les solutions du problème approché associé à la triangulation et au problème considéré, par exemple au voisinage d'une partie sensible du domaine.

3.2.1 Renumerotation

Afin de réduire la largeur de bande des futures matrices associées au problème considéré, on renumérote les classes de noeuds de sorte à réduire la différence entre les numéros des classes des noeuds appartenant à un même triangle. La raison est que, si (a_{ij}) dénote par exemple la

matrice de masse de notre problème, a_{ij} est non nul seulement si $\text{supp}(\varphi_i) \cap \text{supp}(\varphi_j) \neq \emptyset$, c'est-à-dire seulement si les nœuds des classes C_i et C_j appartiennent à un même triangle.

La méthode de renumérotation employée dans ce travail, simple mais non optimale, consiste à fixer un premier nœud (par exemple au bord du domaine), puis à renuméroter les nœuds équivalents ainsi que ses voisins à la suite ; une fois tous les voisins renumérotés, on recommence avec les voisins du deuxième nœud et ainsi de suite.

On remarque que, bien que cette méthode de renumérotation soit relativement facile à mettre en place et permette de faire des calculs à une vitesse satisfaisante (alors qu'ils seraient souvent laborieux avec un mailleur classique seul), la bande des matrices obtenues après renumérotation s'élargit légèrement à mesure qu'on avance sur la diagonale des matrices. Une manière simple (mais non utilisée dans ce travail) de réduire cet effet consiste à mailler ou renuméroter séparément des sous domaines puis à les assembler. Des algorithmes plus évolués permettent également d'obtenir de meilleurs résultats (voir [22]).

3.2.2 Raffinement

Une qualité indispensable d'un bon algorithme de raffinement est de conserver la qualité de la triangulation, c'est-à-dire la rondeur de ses éléments. L'algorithme suivant permet un raffinement d'un domaine ou d'un sous-domaine par itérations successives. A chaque itération, les mailles sont raffinées d'un facteur 2 à l'intérieur de la partie considérée ainsi que sur les éléments voisins (c'est-à-dire sur tous les éléments **contenant** un nœud dans la partie à raffiner), par découpage des éléments en 4. Les voisins de ces nouveaux éléments sont alors découpés en 2 (voir 4 si besoin) afin d'obtenir une triangulation correcte. Sur les éléments découpés en 2, on observe une perte de rondeur à chaque itération ; il ne s'agit toutefois que de quelques éléments (ceux du bord du domaine de raffinement) et, surtout, il s'agit à chaque itération d'éléments différents : la marge donnée par la sélection élargie des éléments à couper en 4 permet de garantir une certaine qualité de la triangulation, indépendamment du nombre d'itérations de raffinement.

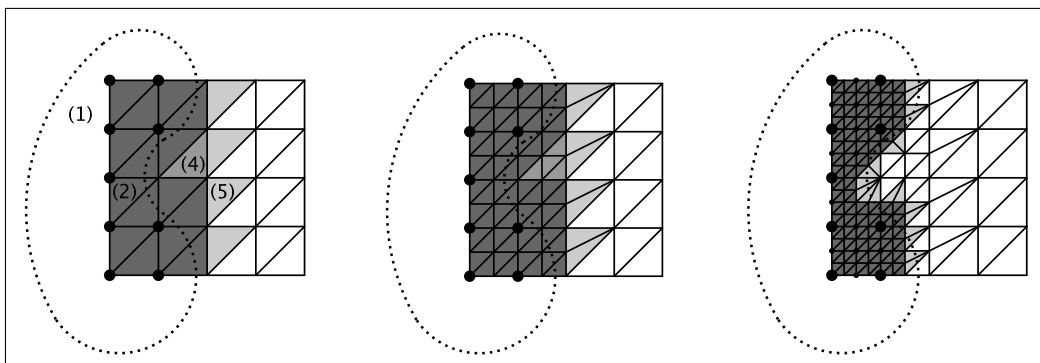


FIGURE 3.7 – Raffinements successifs d'un domaine.

Algorithme de raffinement d'une partie d'un domaine (ou du domaine complet) :

1. On donne la triangulation du domaine ainsi que la partie que l'on veut raffiner comme une liste de (classes de) nœuds.
2. On détermine les éléments à couper en 4 : il s'agit de ceux ayant **au moins** un sommet dans la partie à raffiner (ou équivalent à un tel).
3. On détermine les nœuds appartenant aux éléments à couper en 4 (ou équivalents à des tels), *l'adhérence* de la partie à raffiner.
4. Si un élément a tous ses sommets dans l'adhérence, on le rajoute dans les éléments à couper en 4.
5. On détermine les éléments à couper en 2 : il s'agit de ceux ayant exactement 2 sommets dans l'adhérence. Le nœud *extérieur* à l'adhérence sera utilisé pour la partition de l'élément.
6. On procède finalement à la découpe des éléments, comme illustrée par la figure 3.8. Considérons pour cela un élément de sommets P_1 , P_2 et P_3 .
 - Pour une découpe en 4 éléments, on construit les nœuds P'_1 , P'_2 et P'_3 , définis comme les points milieu des côtés de l'élément, avec P'_i sur le côté opposé à P_i , puis on remplace l'ancien élément par les triangles $P_1P'_3P'_2$, $P_2P'_1P'_3$, $P_3P'_2P'_1$ et $P'_1P'_2P'_3$.
 - Pour une découpe en 2 éléments, supposons que P_3 soit le nœud extérieur à l'adhérence des éléments à couper en 4. On construit alors le point milieu P'_3 du segment P_1P_2 puis on remplace l'ancien élément par les triangles $P_1P'_3P_3$ et $P_2P_3P'_3$.

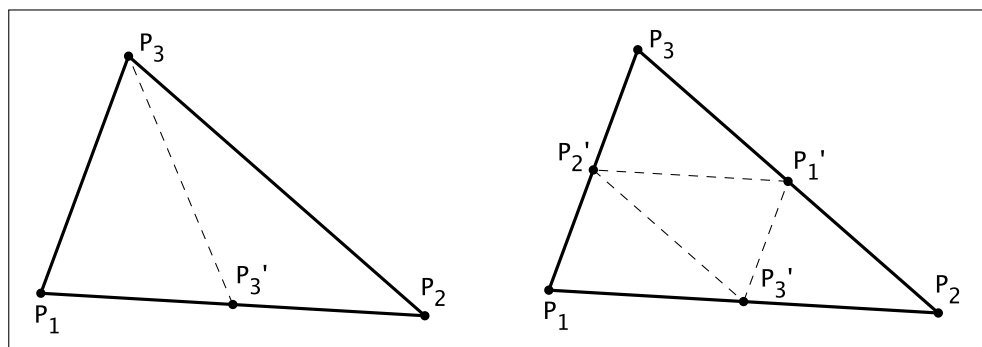


FIGURE 3.8 – Découpe d'éléments en 2 ou 4.

Remarque : dans le cas où une face de l'élément est courbe, on peut utiliser la paramétrisation de la courbe et les valeurs du paramètre aux sommets pour ajuster les nouveaux points sur la courbe.

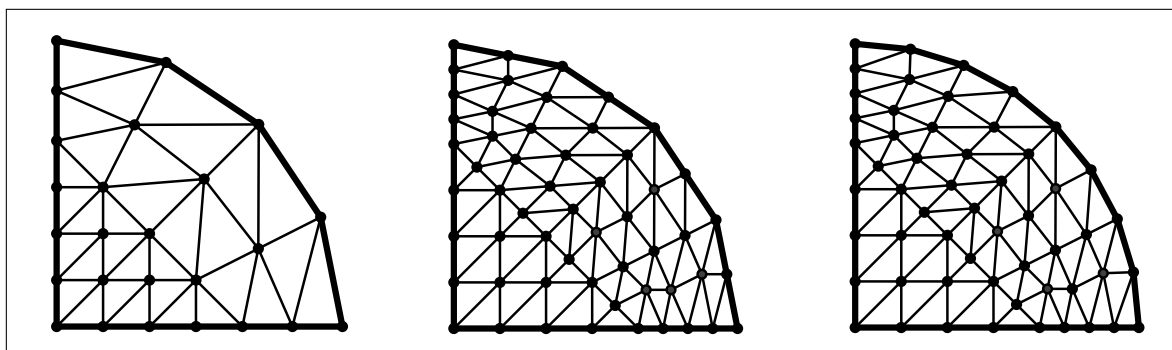


FIGURE 3.9 – Triangulation d’un quart de cercle, puis raffinement local sans, puis avec prise en compte de la paramétrisation du bord.

3.3 Maillage de Delaunay

Considérons à présent un maillage plus sophistiqué que celui par macro-éléments, basé sur la méthode de Voronoï-Delaunay (pour une présentation détaillée, voir [22]).

Ce maillage présente essentiellement deux avantages. Premièrement, il permet une grande automatisation puisqu’il suffit à l’utilisateur de fournir les nœuds du bord du domaine. Deuxièmement, la triangulation obtenue est optimale par rapport aux nœuds qui la composent, en le sens qu’il s’agit d’une triangulation de Delaunay : le cercle circonscrit à chaque élément ne contient aucun nœud en dehors des sommets de l’élément. Autrement dit, chaque paire de triangles adjacents par une arête est telle que le plus petit des angles formés par deux côtés consécutifs est maximal (par rapport aux paires de triangles adjacents par une arête possibles formant le même quadrilatère ; voir figure 3.10).

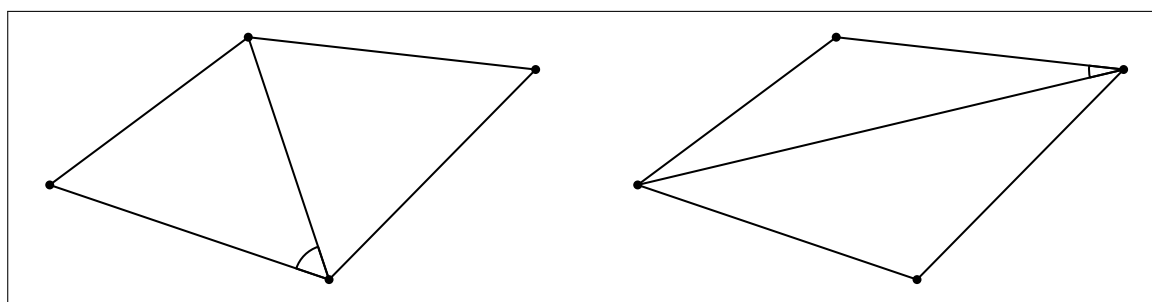


FIGURE 3.10 – Deux triangulations d’un quadrilatère. Celle de gauche est de Delaunay.

Algorithme :

1. On donne le domaine à mailler par son bord en tant que liste de segments orientés, c'est-à-dire une liste de paires de nœuds. Idéalement, ces segments sont isométriques.
2. On détermine les nœuds internes au domaine. Pour cela, on crée une (petite) grille recouvrant le domaine, dont on garde uniquement les points qui se trouvent à l'intérieur du domaine, à plus d'une certaine distance (proportionnelle aux longueurs des segments du bord) des nœuds du bord du domaine (afin d'éviter de créer des éléments trop plats). Pour déterminer si un point est à l'intérieur au domaine, on calcule son indice par rapport à la courbe orientée définissant le bord du domaine (voir section 3.3.2).
3. On détermine ensuite une *boîte*, c'est-à-dire quatre nœuds supplémentaires formant un rectangle, lequel englobe les nœuds du bord du domaine. De préférence, on choisit une grande boîte (proportionnellement au domaine).
4. On maille la boîte à l'aide de deux triangles dont les sommets sont ceux de la boîte.
5. On insère un à un les points de cette grille et ceux du bord du domaine dans la triangulation, selon la stratégie suivante, directement inspirée de la méthode de Voronoï-Delaunay : on détermine puis détruit tous les éléments de la triangulation tels que le point à insérer est interne aux cercles circonscrits (voir section 3.3.1), puis on remaille la partie détruite en joignant le nouveau point aux arêtes externes de la réunion des éléments supprimés.

On obtient alors un *maillage-boîte* du domaine, avec nœuds internes. Il s'agit alors de vérifier que le bord du domaine soit respecté par la triangulation.

6. On vérifie que les segments du bord apparaissent en tant qu'arêtes des éléments de la triangulation. Si un segment n'apparaît pas, on insère son point milieu dans la triangulation (avec la même stratégie que précédemment) et on recommence la vérification (en remplaçant, dans la liste des segments du bord du domaine, le segment non respecté par ses deux moitiés).

Le *maillage-boîte* obtenu respecte alors le bord du domaine. En particulier, chaque élément de la triangulation est soit intérieur, soit extérieur au domaine. Il s'agit alors de supprimer les éléments et nœuds superflus.

7. On supprime les éléments extérieurs au domaine. Pour déterminer si un élément est extérieur au domaine, on calcule l'indice d'un point intérieur au triangle (par exemple

son centre de gravité) par rapport à la courbe orientée définissant le bord du domaine (voir section 3.3.2).

8. On supprime les sommets de la boîte.

On obtient ainsi une triangulation respectant le bord du domaine et dont les éléments sont tous intérieurs au domaine. Il s'agit donc d'une triangulation de Delaunay du domaine, avec nœuds internes.

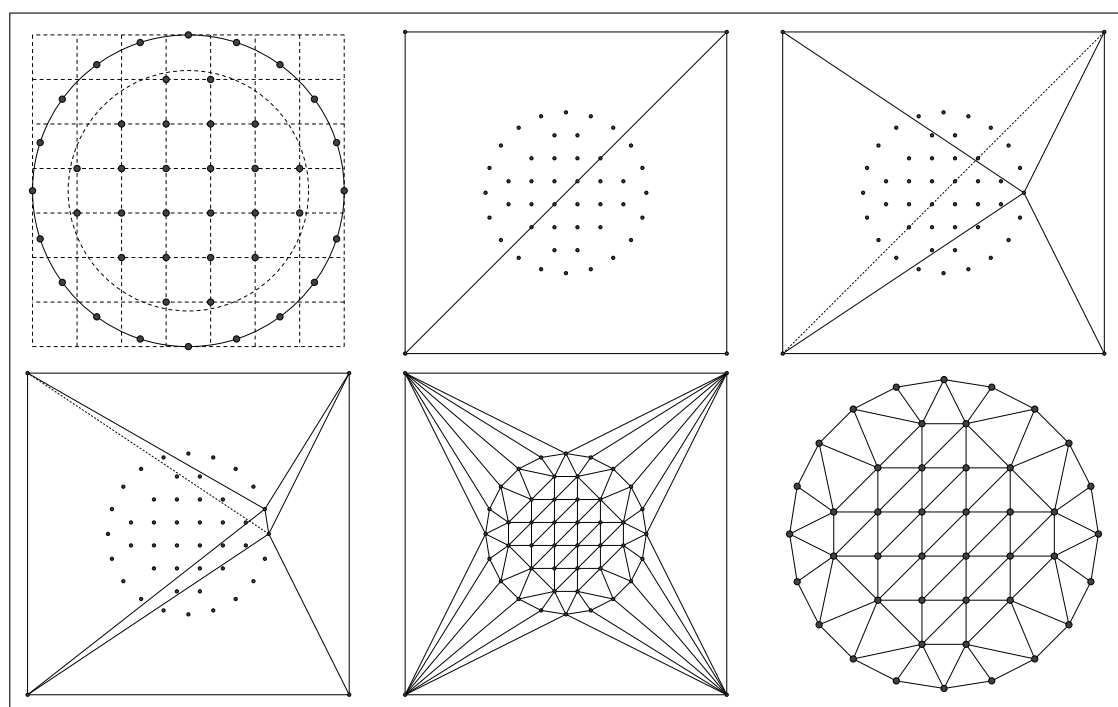


FIGURE 3.11 – Quelques étapes d'un maillage de Delaunay du disque.

Remarquons qu'on aurait aussi pu insérer uniquement les nœuds du bord du domaine, supprimer la boîte et enfin seulement ajouter les nœuds internes au domaine. En effet, l'algorithme d'insertion de points dans une triangulation fonctionne sous l'unique condition que les points soient internes à la réunion des éléments de la triangulation (ce qui explique le rôle de la boîte). De même, la triangulation obtenue est de Delaunay si la triangulation initiale l'est (ce qui est le cas de toute triangulation d'une boîte rectangulaire dont les seuls nœuds sont ses sommets).

Afin d'illustrer le problème du respect du bord du domaine par la triangulation, considérons le domaine polygonal représenté ci-dessous (figure de gauche). Après un maillage grossier, on observe que la triangulation ne respecte pas le bord du domaine (deuxième figure). Ce problème est résolu par l'insertion d'un point supplémentaire (troisième figure) et il est

alors possible de déterminer et de supprimer les éléments extérieurs au domaine (quatrième figure).

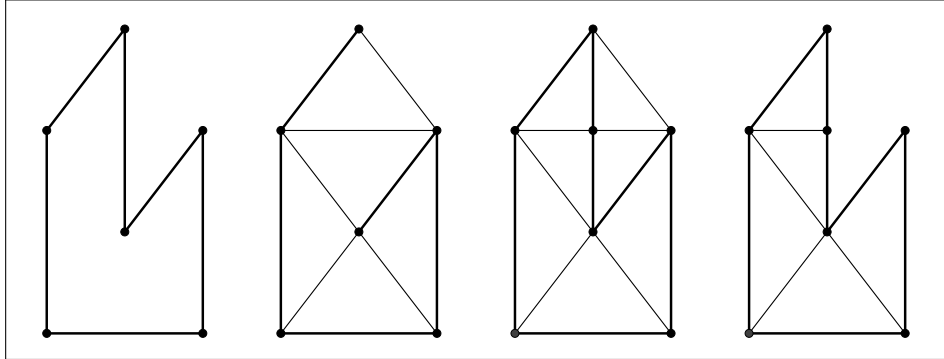


FIGURE 3.12 – Domaine initial, triangulation de Delaunay sans, puis avec correction du bord et triangulation finale.

Parmi les étapes de l'algorithme ci-dessus, deux points techniques méritent qu'on s'attarde à leur mise en œuvre : la détermination de l'appartenance d'un point au cercle circonscrit d'un triangle et le calcul de l'indice d'un point par rapport à une courbe (polygonale).

3.3.1 Appartenance d'un point au cercle circonscrit d'un triangle

Dans \mathbb{R}^2 , soit K un triangle donné par ses sommets P_1, P_2, P_3 et soit A un point. Il s'agit de décider si le point A appartient au cercle circonscrit de K . Pour cela, on détermine le centre C du cercle circonscrit et on compare sa distance au point A avec le rayon du cercle, lui-même défini comme la distance entre C et P_1 (ou P_2 ou P_3).

Le centre C est, par définition, le point équidistant aux points P_1, P_2 et P_3 . Il est donc obtenu comme point d'intersection de 2 droites médiatrices des arêtes de K . Une équation paramétrique d'une telle droite est donnée par

$$d_{ij}(\lambda_{ij}) = \frac{P_i + P_j}{2} + \lambda_{ij}(P_i - P_j)^\perp,$$

avec $i, j \in \{1, 2, 3\}$ différents.

En intersectant d_{12} et d_{13} , on obtient l'équation linéaire

$$\lambda_{13}(P_1 - P_3)^\perp + \lambda_{12}(P_1 - P_2)^\perp = \frac{P_2 - P_3}{2},$$

qu'il s'agit de résoudre et de remplacer dans une des équations paramétriques afin de trouver C . En coordonnées, en notant $P_i = (x_i, y_i)$ et, pour simplifier les écritures,

$$a = (P_2 - P_1)^\perp = \begin{pmatrix} y_2 - y_1 \\ x_1 - x_2 \end{pmatrix}, b = (P_1 - P_3)^\perp = \begin{pmatrix} y_1 - y_3 \\ x_3 - x_1 \end{pmatrix}, c = \frac{P_2 - P_3}{2} = \begin{pmatrix} \frac{x_2 - x_3}{2} \\ \frac{y_2 - y_3}{2} \end{pmatrix},$$

on obtient l'équation matricielle

$$\begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix} \begin{pmatrix} \lambda_{12} \\ \lambda_{13} \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}.$$

Ainsi, en supposant le triangle K non dégénéré, on trouve

$$\begin{pmatrix} \lambda_{12} \\ \lambda_{13} \end{pmatrix} = \frac{1}{a_1 b_2 - a_2 b_1} \begin{pmatrix} b_2 & -b_1 \\ -a_2 & a_1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \frac{1}{a_1 b_2 - a_2 b_1} \begin{pmatrix} b_2 c_1 - b_1 c_2 \\ a_1 c_2 - a_2 c_1 \end{pmatrix},$$

d'où finalement pour le centre C du cercle circonscrit

$$C = \frac{1}{2} \begin{pmatrix} x_1 + x_2 \\ y_1 + y_2 \end{pmatrix} + \frac{b_1 c_2 - b_2 c_1}{a_2 b_1 - a_1 b_2} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}.$$

3.3.2 Indice d'un point par rapport à une courbe

Dans \mathbb{R}^2 , on utilise l'indice d'un point par rapport à une courbe orientée fermée pour déterminer s'il appartient au domaine défini par la courbe (lorsque le point n'appartient pas à cette dernière). Rappelons que l'indice est le nombre de tours effectués par la courbe autour du point. Ainsi, un point appartient à un domaine si et seulement si son indice par rapport à la courbe orientée définissant son bord vaut 1 (dans le cas contraire, il vaut 0, éventuellement -1 si le domaine n'est pas compact). Remarquons que cela est valable même si le bord du domaine contient plusieurs composantes connexes (données par autant de courbes orientées fermées).

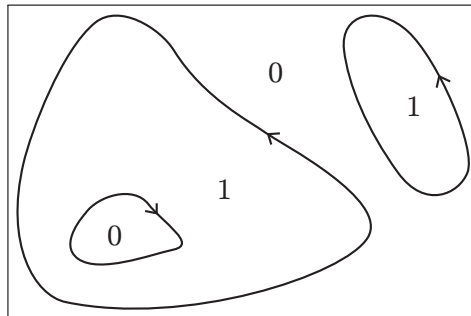


FIGURE 3.13 – Valeurs de l'indice du bord d'un domaine.

Voici deux approches du calcul de l'indice dans le cas d'un domaine polygonal Ω (la deuxième ayant été retenue pour la programmation effective du mailleur de Delaunay), qu'on suppose donné par le polygone connexe de sommets consécutifs P_1, \dots, P_n . Notons $P_{n+1} = P_1$ et

$$Ind_{\Omega}(P) = \sum_{j=1}^n \delta_j$$

l'indice de Ω par rapport à un point P avec, pour chaque j , δ_j la contribution du segment $P_j P_{j+1}$ à l'indice.

Approche par l'analyse complexe

Le premier calcul est basé sur le célèbre résultat d'analyse complexe ci-dessous (voir [2]).

Théorème 9 Soit $\gamma : [a, b] \rightarrow \mathbb{C}$ une courbe fermée, \mathcal{C}^1 par morceaux et soit $z_0 \in \mathbb{C} \setminus \text{Im}(\gamma)$. On définit l'indice $\text{Ind}_\gamma(z_0)$ de z_0 par rapport à γ par

$$\text{Ind}_\gamma(z_0) = \frac{1}{2\pi i} \oint_\gamma \frac{dz}{z - z_0},$$

$$\text{avec } \oint_\gamma f(z)dz = \int_a^b f(\gamma(t)) \cdot \gamma'(t)dt.$$

Alors la fonction

$$\begin{aligned} \text{Ind}_\gamma : \mathbb{C} \setminus \text{Im}(\gamma) &\longrightarrow \mathbb{C} \\ z_0 &\longmapsto \text{Ind}_\gamma(z_0) \end{aligned}$$

est à valeurs dans \mathbb{Z} , constante sur chaque composante connexe de $\mathbb{C} \setminus \text{Im}(\gamma)$, nulle sur la composante non bornée. De plus, $\text{Ind}_\gamma(z_0)$ représente le nombre de tours effectués par la courbe γ autour du point z_0 .

Afin d'utiliser ce résultat, considérons la paramétrisation du bord du domaine

$$\begin{aligned} \gamma : [0, n] &\longrightarrow \mathbb{C} \\ t &\longmapsto \gamma_{[t]+1}(t - [t]), \end{aligned}$$

avec $[.]$ la partie entière et

$$\gamma_j(s) = sP_{k+1} + (1-s)P_k = s(P_{k+1} - P_k) + P_k$$

la paramétrisation du segment $P_k P_{k+1}$ ($s \in [0, 1[$) où $k \equiv j \pmod{n}$.

Calculons la contribution δ_j du segment $P_j P_{j+1}$ à l'indice $\text{Ind}_\gamma(P)$

$$\int_0^1 \frac{\gamma'_j(s)}{\gamma_j(s) - P} ds = \int_0^1 \frac{P_{j+1} - P_j}{s(P_{j+1} - P_j) + P_j - P} ds = \int_0^1 \frac{ds}{s + \frac{P_j - P}{P_{j+1} - P_j}}.$$

Notons pour simplifier $a = \text{Re} \left(\frac{P_j - P}{P_{j+1} - P_j} \right)$ et $b = \text{Im} \left(\frac{P_j - P}{P_{j+1} - P_j} \right)$. Ainsi on trouve

$$\begin{aligned} \int_0^1 \frac{\gamma'_j(s)}{\gamma_j(s) - P} ds &= \int_0^1 \frac{ds}{(s+a) + ib} \\ &= \int_0^1 \frac{s+a}{(s+a)^2 + b^2} ds - ib \int_0^1 \frac{ds}{(s+a)^2 + b^2} \\ &= \frac{1}{2} \int_{a^2+b^2}^{(1+a)^2+b^2} \frac{du}{u} - ib \int_a^{1+a} \frac{dv}{v^2 + b^2}. \end{aligned}$$

Par le théorème, comme $Ind_\gamma(P) \in \mathbb{Z} \subset \mathbb{R}$, seul le deuxième terme est non nul. Si $b = 0$ (i.e. si $P_j - P$ est parallèle à $P_{j+1} - P_j$), alors $Ind_\gamma(P) = 0$; sinon on trouve

$$Im \left(\int_0^1 \frac{\gamma'_j(s)}{\gamma_j(s) - P} ds \right) = \arctan \left(\frac{a}{b} \right) - \arctan \left(\frac{1+a}{b} \right).$$

On obtient ainsi que la contribution δ_j du segment $P_j P_{j+1}$ à l'indice $Ind_\gamma(P)$ vaut

$$\delta_j = \begin{cases} 0 & \text{si } b = 0 \\ \frac{1}{2\pi} \left(\arctan \left(\frac{a}{b} \right) - \arctan \left(\frac{1+a}{b} \right) \right) & \text{sinon .} \end{cases}$$

Calculons finalement a et b en fonction de $P_j = (x_j, y_j)$, $P_{j+1} = (x_{j+1}, y_{j+1})$ et $P = (x, y)$, i.e. $P_j = x_j + iy_j$, $P_{j+1} = x_{j+1} + iy_{j+1}$ et $P = x + iy$ en notations complexes.

$$\begin{aligned} \frac{P_j - P}{P_{j+1} - P_j} &= \frac{(x_j - x) + i(y_j - y)}{(x_{j+1} - x_j) + i(y_{j+1} - y_j)} \\ &= \frac{((x_j - x) + i(y_j - y)) ((x_{j+1} - x_j) - i(y_{j+1} - y_j))}{(x_{j+1} - x_j)^2 - (y_{j+1} - y_j)^2} \\ &= \frac{(x_j - x)(x_{j+1} - x_j) + (y_j - y)(y_{j+1} - y_j)}{(x_{j+1} - x_j)^2 - (y_{j+1} - y_j)^2} \\ &\quad + i \frac{(x_{j+1} - x_j)(y_j - y) - (x_j - x)(y_{j+1} - y_j)}{(x_{j+1} - x_j)^2 - (y_{j+1} - y_j)^2}, \end{aligned}$$

c'est-à-dire

$$\begin{aligned} a &= \frac{(x_j - x)(x_{j+1} - x_j) + (y_j - y)(y_{j+1} - y_j)}{(x_{j+1} - x_j)^2 - (y_{j+1} - y_j)^2}, \\ b &= \frac{(x_{j+1} - x_j)(y_j - y) - (x_j - x)(y_{j+1} - y_j)}{(x_{j+1} - x_j)^2 - (y_{j+1} - y_j)^2}. \end{aligned}$$

Approche manuelle

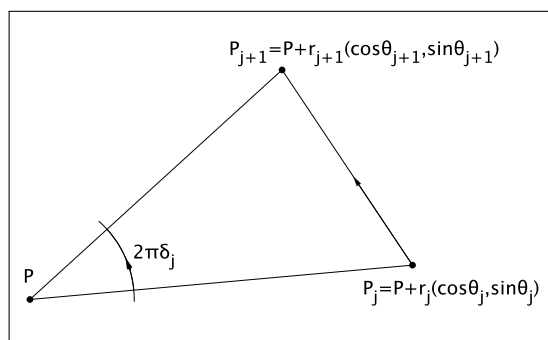
Soit, pour tout j , θ_j la mesure dans l'intervalle $]-\pi/4, 7\pi/4]$ de l'angle de P_j par rapport à P . Autrement dit, θ_j est l'unique angle de $]-\pi/4, 7\pi/4]$ tel qu'il existe $r_j > 0$ avec $P_j = P + r_j(\cos \theta_j, \sin \theta_j)$.

La contribution δ_j du segment $P_j P_{j+1}$ à l'indice $Ind_\Omega(P)$ vaut alors

$$\delta_j = \frac{1}{2\pi} (\alpha_{j+1} - \alpha_j),$$

où $\alpha_1 = \theta_1$ et, une fois $\alpha_j \equiv \theta_j \pmod{\pi}$ fixé, on pose $\alpha_{j+1} \equiv \theta_{j+1} \pmod{\pi}$ avec

$$\begin{cases} \alpha_{j+1} \in [\alpha_j, \alpha_j + \pi[& \text{si l'orientation du triangle } PP_j P_{j+1} \text{ est positive ou nulle} \\ \alpha_{j+1} \in]\alpha_j - \pi, \alpha_j[& \text{sinon .} \end{cases}$$

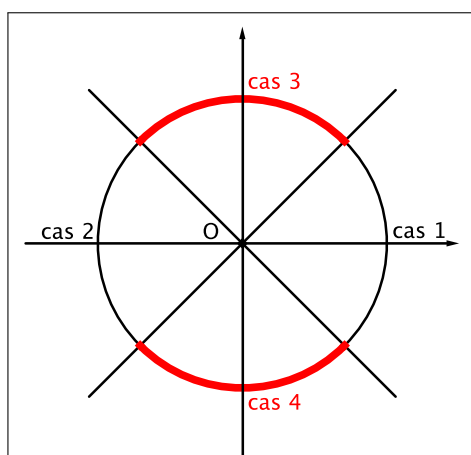
FIGURE 3.14 – Contribution δ_j du segment $P_j P_{j+1}$ à l'indice.

Remarques :

- On calcule θ_j en évaluant (par exemple) la fonction $\theta : \mathbb{S}^1 \rightarrow \mathbb{R}$ donnée ci-dessous au point $\frac{P_j - P}{\|P_j - P\|}$.

$$\theta(x, y) = \begin{cases} \arcsin(y) & \text{si } |y| < |x| \text{ et } x > 0 \\ \pi - \arcsin(y) & \text{si } |y| < |x| \text{ et } x < 0 \\ \arccos(x) & \text{si } |y| > |x| \text{ et } y > 0 \\ 2\pi - \arccos(x) & \text{si } |y| > |x| \text{ et } y < 0. \end{cases}$$

- L'orientation du triangle $PP_j P_{j+1}$ est donnée par le signe de $\cos \theta_j \sin \theta_{j+1} - \sin \theta_j \cos \theta_{j+1}$.

FIGURE 3.15 – Définition de la fonction θ .

3.4 Maillage mixte

Le mailleur de Delaunay présenté précédemment a la particularité que la triangulation obtenue du cœur du domaine (c'est-à-dire la partie constituée des nœuds internes) ressemble à une triangulation obtenue à l'aide d'un mailleur par macro-éléments. Partant de cette observation et dans le but d'optimiser le temps de calcul, j'ai élaboré un mailleur mixte, c'est-à-dire une procédure maillant le cœur du domaine de la même manière qu'avec des macro-éléments (à l'aide d'une grille) et le reste avec Delaunay.

Algorithme :

1. Comme pour le mailleur de Delaunay, on donne le domaine par son bord.
2. On crée comme avant une (petite) grille recouvrant le domaine, dont on garde non seulement les points intérieurs au domaine (se trouvant suffisamment loin des nœuds du bord), mais également les éléments formés par ces points (voir le mailleur par macro-éléments). Le sous-domaine ainsi maillé est appelé *cœur* du domaine.
3. On maille ensuite par Delaunay la partie définie par le bord du domaine et le bord de la partie cœur parcouru en sens inverse (appelé *bord interne*), sans insérer de point interne. Le sous-domaine ainsi maillé est appelé *frontière* du domaine.
4. Si le maillage de la frontière a nécessité l'insertion de nœuds supplémentaires sur le bord interne (afin de le respecter), il s'agit également de les insérer dans la triangulation du cœur du domaine.
5. On assemble ces deux triangulations pour obtenir celle du domaine complet.

La complexité de cet algorithme, les problèmes potentiels de non respect du bord interne par le maillage de la frontière et, face à cela, le faible gain de temps par rapport au mailleur de Delaunay pur (pour les exemples traités du moins) font que je n'ai en général pas utilisé cet algorithme, qui, de ce fait, n'est pas inclu dans les programmes joints à ce document.

3.5 Calcul matriciel

Soient $\{D_\alpha\}$ un ensemble de domaines polygonaux de \mathbb{R}^2 , \mathcal{R} une relation d'équivalence sur la réunion disjointe $\amalg D_\alpha$ des D_α et $\{\tau_{h,\alpha}\}$ un ensemble de triangulations des D_α associées à un 2-simplexe (K, P_K, Σ_K) de type 1 (P_K est l'ensemble des polynômes de \mathbb{R}^2 dans \mathbb{R} de degrés inférieurs ou égaux à 1). On suppose que la relation \mathcal{R} et les triangulations $\tau_{h,\alpha}$ respectent les partitions $\partial D_\alpha = \partial D_\alpha^D \amalg \partial D_\alpha^N \amalg \partial D_\alpha^0$. Finalement, on suppose que les triangulations $\tau_{h,\alpha}$ respectent les conditions de compatibilité, régularité et inter-compatibilité.

Il s'agit de résoudre le problème

$$(\tilde{P}_{f,h}) \quad \begin{cases} \text{Trouver } \tilde{u}_h \in \tilde{H}_h \text{ et } \lambda_h \in \mathbb{R} \text{ tels que, pour tout } \tilde{v}_h \in \tilde{H}_h : \\ \tilde{a}(\tilde{u}_h, \tilde{v}_h) = \lambda_h \tilde{b}(\tilde{u}_h, \tilde{v}_h) , \end{cases}$$

avec

$$\begin{aligned} \tilde{H}_{\alpha,h} &= \{v \in \mathcal{C}^0(\overline{D_i}) : v|_K \in P_K \text{ et } v|_{\partial D_\alpha^D} = 0\} , \\ \tilde{H}_h &= \left\{ v \in \mathbb{R}^{\Pi D_\alpha} : v|_{\overline{D_\alpha}} \in \tilde{H}_{\alpha,h} \right\} . \end{aligned}$$

Notons N_0 le nombre de classes d'équivalence de nœuds des triangulations, N le nombre de celles dont les nœuds ne se trouvent pas sur $\Pi \partial D_\alpha^D$ et $\{C_i\}_{i=1}^{N_0} = \{[P_i]\}_{i=1}^{N_0}$ les classes d'équivalence de nœuds des triangulations, avec $P_i \in \Pi \partial D_\alpha^D$ si et seulement si $i > N$.

Considérons alors la base de \tilde{H}_h constituée des *fonctions formes* de $\{\tau_{h,\alpha}\}$, c'est-à-dire des fonctions $\{\varphi_i\}_{i=1}^N$ définies sur ΠD_α telles que $\varphi_i(P_j) = \chi_{C_i}(P_j)$. Restreintes aux éléments des triangulations, il s'agit des coordonnées barycentriques (ou fonctions formes). Du point de vue numérique, l'avantage de cette base est que le support de chacun de ses éléments est petit : celui de φ_i est constitué uniquement des triangles contenant les nœuds de la classe C_i .

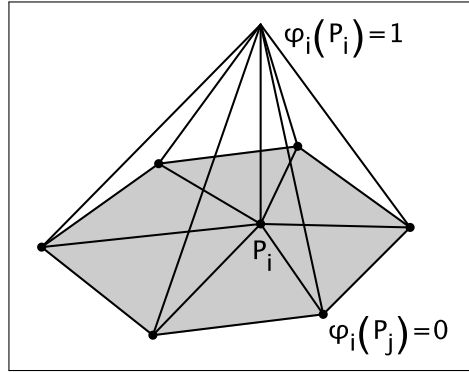


FIGURE 3.16 – Fonctions formes $\varphi_i(P_j) = \delta_{ij}$.

Ainsi, chercher la solution \tilde{u}_h de $(\tilde{P}_{f,h})$ revient à chercher les coefficients de \tilde{u}_h dans la base des fonctions formes. Autrement dit, si $\tilde{u}_h = \sum_{j=1}^N \xi_j \varphi_j$, on cherche $\{\xi_i\}_{i=1}^N$ tels que

$$\sum_{j=1}^N \xi_j \tilde{a}(\varphi_j, \varphi_i) = \lambda_h \sum_{j=1}^N \xi_j \tilde{b}(\varphi_j, \varphi_i) ,$$

pour tout $i \in \{1, \dots, N\}$.

Ainsi, en notant $r_{ij} = \tilde{a}(\varphi_j, \varphi_i)$, $m_{ij} = \tilde{b}(\varphi_j, \varphi_i)$, $\xi = (\xi_i)_{i=1}^N$, $R = (r_{ij})_{i,j=1}^N$ et

$M = (m_{ij})_{i,j=1}^N$, on trouve que $(\tilde{P}_{f,h})$ est équivalent à chercher $\xi \in \mathbb{R}^N$ et $\lambda_h \in \mathbb{R}$ vérifiant l'équation matricielle

$$R\xi = \lambda_h M\xi. \quad (3.1)$$

On appelle R *matrice de rigidité* et M *matrice de masse*. Remarquons que ces matrices sont symétriques définies positives. Notons alors $M = N^t N$ la décomposition de Cholesky de M , avec N triangulaire supérieure. Ainsi, chercher ξ et λ_h vérifiant l'équation matricielle (3.1) équivaut à chercher ξ et λ_h vérifiant

$$N^{-t} R\xi = \lambda_h N\xi$$

ou encore, en notant $A = N^{-t} R N^{-1}$ et $\eta = N\xi$, à chercher η et λ_h vérifiant

$$A\eta = \lambda_h \eta.$$

Notons toutefois que, lors de la résolution numérique du problème, nous ne calculerons pas explicitement la matrice A . En effet, l'algorithme de résolution du problème matriciel aux valeurs propres utilisé, à savoir l'algorithme de Lanczos, a l'avantage de ne pas modifier la matrice A du problème et ne demande à l'utilisateur qu'une procédure de calcul du produit matriciel $v \mapsto w = Av$. Il s'agira donc de proposer une procédure de résolution de l'équation en w

$$w = N^{-t} R N^{-1} v,$$

c'est-à-dire de déterminer w_1 , w_2 et enfin w par résolutions successives des problèmes

$$\begin{aligned} Nw_1 &= v, \\ w_2 &= R w_1, \\ N^t w &= w_2. \end{aligned}$$

3.5.1 Construction des matrices de rigidité et de masse

Le calcul des matrices de rigidité et de masse du problème peut se faire triangle après triangle, via des matrices locales de taille 3×3 , assemblées ensuite dans les matrices globales. En effet,

$$\begin{aligned} r_{ij} &= \tilde{a}(\varphi_j, \varphi_i) = \sum_{\alpha} \int_{D_{\alpha}} \tilde{a}_{\alpha}(\varphi_j|_{D_{\alpha}}, \varphi_i|_{D_{\alpha}}) \\ &= \sum_{\alpha} \sum_{k,l=1}^2 \sum_{K \in \tau_{h,\alpha}} \int_{D_{\alpha}} \sqrt{\det G_{\alpha}} g_{\alpha}^{kl} \frac{\partial \varphi_j}{\partial x_k} \frac{\partial \varphi_i}{\partial x_l} \\ &= \sum_{\alpha} \sum_{K \in \tau_{h,\alpha}} r_{ij}^K, \end{aligned}$$

avec

$$r_{ij}^K = \sum_{k,l=1}^2 \int_K \sqrt{\det G_{\alpha}} g_{\alpha}^{kl} \frac{\partial \varphi_j}{\partial x_k} \frac{\partial \varphi_i}{\partial x_l}. \quad (3.2)$$

De même,

$$\begin{aligned} m_{ij} &= \tilde{b}(\varphi_j, \varphi_i) = \sum_{\alpha} \int_{D_{\alpha}} \tilde{b}_{\alpha}(\varphi_j|_{D_{\alpha}}, \varphi_i|_{D_{\alpha}}) \\ &= \sum_{\alpha} \sum_{K \in \tau_{h,\alpha}} \int_{D_{\alpha}} \sqrt{\det G_{\alpha}} \varphi_j \varphi_i \\ &= \sum_{\alpha} \sum_{K \in \tau_{h,\alpha}} m_{ij}^K, \end{aligned}$$

avec

$$m_{ij}^K = \int_K \sqrt{\det G_{\alpha}} \varphi_j \varphi_i. \quad (3.3)$$

Et comme le support de φ_i contient uniquement les triangles contenant les nœuds de la classe C_i , r_{ij}^K et m_{ij}^K sont non nuls seulement si les classes C_i et C_j ont un nœud appartenant à K . Il s'agit donc de calculer, pour chaque triangle K disons de sommets appartenant aux classes C_{i_1} , C_{i_2} et C_{i_3} , les matrices locales $R^K = (r_{i_j i_k}^K)_{j,k=1}^3$ et $M^K = (m_{i_j i_k}^K)_{j,k=1}^3$.

3.5.2 Construction des matrices locales

Soit K un triangle donné, de sommets P_{i_1} , P_{i_2} et P_{i_3} . Considérons \hat{K} le 2-simplexe de référence, c'est-à-dire le triangle de sommets $\hat{P}_1(0,0)$, $\hat{P}_2(1,0)$ et $\hat{P}_3(0,1)$, et l'application affine F_K envoyant \hat{K} sur K , avec $F_K(\hat{P}_j) = P_{i_j}$, pour $j \in \{1, 2, 3\}$.

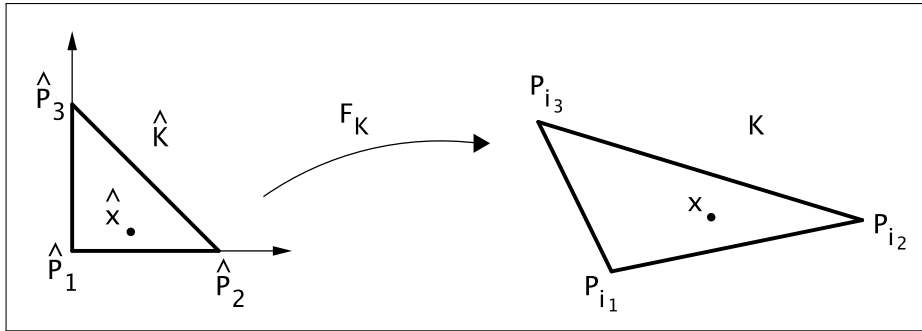


FIGURE 3.17 – 2-simplexe de référence et application associée.

En notant $\hat{\varphi}_j$ ($j \in \{1, 2, 3\}$) les coordonnées barycentriques de \hat{K} , on trouve donc

$$F_K(\hat{x}) = \sum_{j=1}^3 P_{i_j} \hat{\varphi}_j(\hat{x}),$$

avec

$$\begin{aligned}\hat{\varphi}_1(\hat{x}_1, \hat{x}_2) &= 1 - \hat{x}_1 - \hat{x}_2, \\ \hat{\varphi}_2(\hat{x}_1, \hat{x}_2) &= \hat{x}_1, \\ \hat{\varphi}_3(\hat{x}_1, \hat{x}_2) &= \hat{x}_2.\end{aligned}$$

Ainsi, lorsque l'on veut calculer les matrices locales R^K et M^K associées à K , dont les composantes $r_{i_j i_k}^K$ et $m_{i_j i_k}^K$ ($j, k \in \{1, 2, 3\}$) sont du type $\int_K \psi(x) dx$, on utilise la formule

$$\int_K \psi(x) dx = \int_{\hat{K}} \psi(F_K(\hat{x})) \cdot J_K(\hat{x}) d\hat{x}, \quad (3.4)$$

avec J_K le déterminant de la jacobienne de F_K . Remarquons que J_K est constant (puisque F_K est affine). En détails, comme $DF_K = \sum_{j=1}^3 P_{i_j} D\hat{\varphi}_j(\hat{x})$, on trouve

$$J_K = (-P_{i_{11}} + P_{i_{21}})(-P_{i_{12}} + P_{i_{32}}) - (-P_{i_{12}} + P_{i_{22}})(-P_{i_{11}} + P_{i_{31}}).$$

On est ainsi ramené à des intégrales sur \hat{K} et il s'agit d'utiliser des techniques d'intégration numérique afin de calculer les matrices locales, avant de les assembler dans les matrices globales.

3.5.3 Intégration numérique

Les deux formules suivantes d'intégration numérique d'une fonction η sur un triangle K sont utilisées dans ce travail ; elles ont toutes deux pour ordre de convergence $O(h)$ (avec h la taille du triangle), ce qui est cohérent avec la méthode des éléments finis employée. Pour plus de détails concernant ces formules, voir [50] ou, dans le contexte des éléments finis, [16].

a) Si P_i ($i \in \{1, 2, 3\}$) désignent les sommets du triangle K , alors

$$\int_K \eta(x) dx \cong \frac{1}{3} \text{Aire}(K) \sum_{i=1}^3 \eta(P_i). \quad (3.5)$$

Cette formule est exacte lorsque η est un polynôme de degré inférieur ou égal à 1.

b) Si Q_i ($i \in \{1, 2, 3\}$) désignent les points milieux des arêtes du triangle K , alors

$$\int_K \eta(x) dx \cong \frac{1}{3} \text{Aire}(K) \sum_{i=1}^3 \eta(Q_i). \quad (3.6)$$

Cette formule est exacte lorsque η est un polynôme de degré inférieur ou égal à 2.

Dans notre cas, K est le 2-simplexe de référence \hat{K} , qui est d'aire $1/2$.

Ainsi, appliquer la deuxième formule d'intégration (3.6) aux composantes des matrices locales $r_{i_j i_k}^K$ (3.2) et $m_{i_j i_k}^K$ (3.3) donne, avec (3.4),

$$r_{i_j i_k}^K \cong \frac{1}{6} \sum_{l=1}^3 \sum_{m,n=1}^2 J_K \sqrt{\det G_\alpha(F_K(\hat{Q}_l))} g_\alpha^{kl}(F_K(\hat{Q}_l)) \frac{\partial \hat{\varphi}_{i_k}}{\partial x_m}(\hat{Q}_l) \frac{\partial \hat{\varphi}_{i_j}}{\partial x_n}(\hat{Q}_l),$$

$$m_{i_j i_k}^K \cong \frac{1}{6} \sum_{l=1}^3 J_K \cdot \sqrt{\det G_\alpha(F_K(\hat{Q}_l))} \hat{\varphi}_{i_k}(\hat{Q}_l) \hat{\varphi}_{i_j}(\hat{Q}_l).$$

Lorsque la métrique du domaine associé est euclidienne, cette formule d'intégration appliquée aux matrices du système a l'avantage d'être exacte; en particulier, les valeurs propres du système restent supérieures à celles du domaine polygonale associé.

De même, appliquer la première formule d'intégration (3.5) aux composantes de la matrice locale de masse (3.3) donne, avec (3.4),

$$m_{i_j i_k}^K \cong \frac{1}{6} \sum_{l=1}^3 J_K \sqrt{\det G_\alpha(F_K(\hat{P}_l))} \hat{\varphi}_{i_k}(\hat{P}_l) \hat{\varphi}_{i_j}(\hat{P}_l) = \frac{1}{6} \delta_{jk} J_K \sqrt{\det G_\alpha(F_K(\hat{P}_j))}.$$

L'application de cette formule d'intégration à la matrice de masse a donc l'avantage de rendre cette dernière diagonale, ce qui simplifie et donc accélère beaucoup les calculs lors de la résolution du problème matriciel aux valeurs propres généralisé (3.1).

Cette procédure, connue sous le nom de *mass-lumping*, ne permet toutefois plus de garantir que les valeurs propres associées soient supérieures à celles du domaine polygonal initial. Armentano et Durán ([6]) ont cependant démontré que les valeurs propres obtenues avec *mass-lumping* sont inférieures à celles obtenues par une intégration exacte. De plus, lorsque les fonctions propres associées sont singulières, les valeurs propres obtenues avec *mass-lumping* sont supérieures aux valeurs exactes (pour une triangulation suffisamment fine), ce qui fait du *mass-lumping* un procédé en tout point supérieur à l'intégration exacte. Lorsque les fonctions propres associées sont régulières par contre, Armentano et Durán n'ont pu que conjecturer l'inégalité inverse, à savoir que les valeurs propres obtenues avec *mass-lumping* seraient inférieures aux valeurs exactes (pour une triangulation suffisamment fine), ce qui permettrait de coincer les valeurs exactes entre celles calculées avec et sans *mass-lumping*.

Notons encore que l'utilisation du *mass-lumping* s'apparente au procédé consistant, une fois la matrice de masse calculée de manière exacte, à sommer les éléments d'une même (demi-) ligne sur la diagonale dans le but de rendre la matrice diagonale. En effet, si on prend le cas d'un problème de Neumann sur un domaine du plan (i.e. avec métrique euclidienne) et

qu'on applique la deuxième formule d'intégration (3.6), on trouve

$$\sum_{j=1}^3 \int_K \varphi_{i_1} \varphi_{i_j} \cong \frac{1}{6} \sum_{k=1}^3 \left((\hat{\varphi}_{i_1}(\hat{Q}_k))^2 + \hat{\varphi}_{i_1}(\hat{Q}_k) \hat{\varphi}_{i_2}(\hat{Q}_k) + \hat{\varphi}_{i_1}(\hat{Q}_k) \hat{\varphi}_{i_3}(\hat{Q}_k) \right) = \frac{1}{6} J_K$$

puisque les termes de la somme prennent 4 fois la valeur 1/4 et 5 fois la valeur 0. En assemblant les matrices locales dans la matrice de masse, c'est-à-dire en sommant sur les triangles K , on obtient donc le même résultat qu'en utilisant la première formule d'intégration (3.5).

3.6 Résolution du problème aux valeurs propres matriciel

On se propose de résoudre le problème

$$(P_m) \quad \begin{cases} \text{Trouver } \eta \in \mathbb{R}^N \text{ et } \lambda \in \mathbb{R} \text{ tels que :} \\ A\eta = \lambda\eta, \end{cases}$$

avec A une matrice symétrique définie positive. Pour ce faire, on utilise l'algorithme itératif de Lanczos, qui permet le calcul d'un (petit) nombre de valeurs propres d'une partie fixée du spectre, dans le cas qui nous intéresse les plus petites valeurs propres, ainsi que des vecteurs propres correspondants. L'algorithme, basé sur les méthodes de la puissance et de projection, a l'avantage de ne pas modifier la matrice A du problème et ne demande à l'utilisateur qu'une procédure de calcul de produit matriciel $v \mapsto Av$. Pour plus de détails concernant les méthodes et algorithmes présentés dans ce qui suit, le lecteur est invité à se référer, par exemple, à [46], [38], [13] ou [32].

3.6.1 Méthode de la puissance

Cette méthode propose d'approcher la plus grande valeur propre λ de A ainsi qu'un vecteur propre η associé.

Algorithme :

1. Fixer $q \in \mathbb{R}^N$ un vecteur initial et poser $q_1 = q / \|q\|$.
2. Jusqu'à la convergence souhaitée, faire $q_{k+1} = \frac{Aq_k}{\|Aq_k\|}$.

Proposition 3 *Soit A une matrice diagonalisable possédant une seule valeur propre λ_1 de valeur absolue maximale. Alors, si le vecteur initial q n'est pas orthogonal à l'espace propre associé, la suite $\left\{ \frac{\|Aq_k\|}{\|q_k\|} \right\}_{k \geq 1}$ converge vers $|\lambda_1|$ et la suite $\{q_k\}_{k \geq 1}$ converge vers un vecteur propre associé.*

Preuve

Soient $\{v_i\}_{i=1}^n$ une famille orthonormée de vecteurs propres de A , $\{\lambda_i\}_{i=1}^n$ les valeurs propres associés avec $|\lambda_1| > |\lambda_2|, \dots, |\lambda_n|$ et $\{\alpha_i\}_{i=1}^n$ tel que $q = \sum_{i=1}^n \alpha_i u_i$.

On a alors

$$\begin{aligned} q_1 &= \frac{1}{\|q\|} \cdot \sum_{i=1}^n \alpha_i u_i, \\ Aq_1 &= \frac{1}{\|q\|} \cdot \sum_{i=1}^n \alpha_i \lambda_i u_i, \\ q_2 &= \frac{1}{\|q\|} \cdot \frac{1}{\|Aq_1\|} \cdot \sum_{i=1}^n \alpha_i \lambda_i^2 u_i, \\ Aq_2 &= \frac{1}{\|q\|} \cdot \frac{1}{\|Aq_1\|} \cdot \sum_{i=1}^n \alpha_i \lambda_i^3 u_i. \end{aligned}$$

Donc, par induction,

$$q_k = \frac{1}{\|q\|} \cdot \frac{1}{\|Aq_1\|} \cdot \dots \cdot \frac{1}{\|Aq_{k-1}\|} \cdot \sum_{i=1}^n \alpha_i \lambda_i^{k-1} u_i.$$

En notant $C_k = \frac{1}{\|q\|} \cdot \frac{1}{\|Aq_1\|} \cdot \dots \cdot \frac{1}{\|Aq_{k-1}\|}$, on a donc

$$q_k = C_k \lambda_1^{k-1} \left(\alpha_1 u_1 + \sum_{i=2}^n \alpha_i \left(\frac{\lambda_i}{\lambda_1} \right)^{k-1} u_i \right).$$

Or $\left(\frac{\lambda_i}{\lambda_1} \right)^{k-1} \xrightarrow{k \rightarrow \infty} 0$, donc $q_k - (C_k \lambda_1^{k-1} \alpha_1) u_1 \xrightarrow{k \rightarrow \infty} 0$.

Finalement, comme q_k et u_1 sont normés, alors $q_k \xrightarrow{k \rightarrow \infty} u_1$ et à fortiori $\frac{\|Aq_k\|}{\|q_k\|} \xrightarrow{k \rightarrow \infty} |\lambda_1|$. □

Citons encore les variantes suivantes de l'algorithme :

- Le principe de la **méthode de la puissance inverse** consiste à itérer A^{-1} au lieu de A pour obtenir la plus petite valeur propre. Pour une question de coût, on ne calcule toutefois pas explicitement l'inverse de A ; on utilisera plutôt une décomposition LR de la matrice (décomposition de Cholesky dans le cas symétrique), avant de résoudre les systèmes linéaires triangulaires associés.

- Un *shift* permet d'améliorer la vitesse de convergence et de chercher la valeur propre la plus éloignée, respectivement la plus proche, de σ , en itérant $(A - \sigma I)$ plutôt que A , respectivement $(A - \sigma I)^{-1}$ plutôt que A^{-1} . Le shift permet également d'éviter le problème de non unicité de la valeur propre de valeur absolue maximale. Cette variante est appelée **méthode de la puissance (inverse) avec shift**.

3.6.2 Méthodes de projection

Le principe des méthodes de projection est d'approcher un vecteur propre η par un vecteur $\tilde{\eta}$ d'un espace \mathcal{K} , en imposant $A\tilde{\eta} - \tilde{\lambda}\tilde{\eta} \perp \mathcal{K}$ pour un certain $\tilde{\lambda}$. Autrement dit, on résoud $A_m y = \tilde{\lambda}y$, avec $A_m = Q^t A Q$, où Q est une matrice dont les colonnes forment une base orthonormée de \mathcal{K} (de dimension m), et $\tilde{\eta} = Qy$. La matrice A_m représente donc la projection de $A|_{\mathcal{K}}$ sur \mathcal{K} dans la base donnée par Q .

Algorithme de Rayleigh-Ritz :

1. Calculer une base orthonormée $\{q_1, \dots, q_m\}$ de \mathcal{K} . On définit $Q = \begin{pmatrix} | & & | \\ q_1 & \dots & q_m \\ | & & | \end{pmatrix}$.
2. Calculer $A_m = Q^t A Q$.
3. Calculer les valeurs propres de A_m et sélectionner celles désirées, disons $\tilde{\lambda}_1, \dots, \tilde{\lambda}_k$.
4. Calculer les vecteurs propres y_1, \dots, y_k de A_m associés.
5. Calculer les *vecteurs de Ritz*, c'est-à-dire les vecteurs propres approchés $\tilde{\eta}_1, \dots, \tilde{\eta}_k$ de A associés aux y_i via la relation $\tilde{\eta}_i = Qy_i$.

Remarque : si \mathcal{K} est invariant par A , les valeurs et vecteurs propres sont exacts.

3.6.3 Méthode des sous-espaces de Krylov

Il s'agit d'un cas particulier des méthodes de projection où l'on choisit pour \mathcal{K} le *sous-espace de Krylov* défini par

$$\mathcal{K}_m = \langle q, Aq, A^2q, \dots, A^{m-1}q \rangle = \{p(A)q : p \text{ pol. de degré } \leq m-1\},$$

où q est un vecteur initial.

L'algorithme correspondant consiste en l'algorithme de Rayleigh-Ritz, dans lequel on utilise en première étape *l'algorithme d'orthonormalisation de Gram-Schmidt* pour construire une base orthonormée de $\{q, Aq, A^2q, \dots, A^{m-1}q\}$.

Algorithme d'Arnoldi :

1. Fixer $q \in \mathbb{R}^N$ un vecteur initial et poser $q_1 = q / \|q\|$.
2. Pour $j = 1$ à m , faire :

$$h_{ij} = \langle Aq_i, q_j \rangle, \text{ pour tout } i \in \{1, \dots, j\},$$

$$r = Aq_j - \sum_{i=1}^j h_{ij}q_i,$$

$$h_{j+1,j} = \|r\|,$$

$$q_{j+1} = r/h_{j+1,j}.$$

En notant $Q_m = \begin{pmatrix} | & & | \\ q_1 & \dots & q_m \\ | & & | \end{pmatrix}$ et $A_m = (h_{ij})$, on obtient

$$AQ_m = Q_m A_m + h_{m+1,m} q_{m+1} e_m^t,$$

donc

$$Q_m^t A Q_m = A_m.$$

Ainsi, les valeurs propres approchées $\tilde{\lambda}$, obtenues par projection sur \mathcal{K}_m , sont les valeurs propres de A_m et les vecteurs de Ritz associés sont définis par $\tilde{\eta} = Q_m y$, avec y vecteur propre de A_m associé à $\tilde{\lambda}$.

3.6.4 Algorithme de Lanczos

Il s'agit du cas particulier de l'algorithme d'Arnoldi, lorsque A est symétrique : dans ce cas, les calculs sont simplifiés puisque A_m est tridiagonale symétrique.

L'algorithme d'Arnoldi devient alors, en notant $\alpha_j = h_{jj}$ et $\beta_j = h_{j-1,j}$:

1. Fixer $q \in \mathbb{R}^N$ un vecteur initial et poser $q_1 = q / \|q\|$.
2. Poser $\beta_1 = 0$ et $q_0 = 0$.
3. Pour $j = 1$ à m , faire :

$$r = Aq_j - \beta_j q_{j-1},$$

$$\alpha_j = \langle r, q_j \rangle,$$

$$\beta_{j+1} = \|r\|,$$

$$q_{j+1} = r/\beta_{j+1}.$$

En arithmétique exacte, les vecteurs q_i forment une base orthogonale et les vecteurs de Ritz $\tilde{\eta}$ ainsi que les valeurs propres approchées $\tilde{\lambda}$ sont de bonnes approximations des vecteurs et valeurs propres de A . Dans la pratique toutefois, suite aux erreurs d'arrondis des calculs numériques, l'orthogonalité est rapidement perdue et il n'est pas rare que des vecteurs q_i se retrouvent linéairement dépendants. Aussi, tel quel, l'algorithme de Lanczos semble instable et il s'agit donc d'employer des procédures d'orthogonalisation, soit à chaque itération (avec un coût important), soit de manière sélective. Toutefois, Paige a montré dans sa thèse ([37]) que ce phénomène n'empêchait pas l'algorithme de Lanczos de trouver des valeurs et vecteurs propres corrects, les erreurs ne faisant en fait que ralentir la convergence !

Des variations et améliorations de l'algorithme de Lanczos permettent de résoudre ce problème, ainsi que ceux de dépendance au vecteur initial q . Les plus connues sont les algorithmes *block Lanczos* (voir [46]), qui utilisent un bloc de plusieurs vecteurs initiaux à la place d'un seul, et les algorithmes qui redémarrent après un certain nombre d'itérations, comme la méthode *implicitly restarted Lanczos* (voir [13] ou [38]). Ces deux algorithmes sont d'ailleurs utilisés simultanément dans la librairie *ARPACK* (voir [32] ou [34]), elle-même employée dans ce travail.

Chapitre 4

Problèmes d'anses fines

Parmi les domaines bidimensionnels a priori peu adaptés à un calcul numérique du spectre du Laplacien, on trouve les domaines à anses fines, c'est-à-dire ceux dont une partie est beaucoup plus fine que le reste du domaine, au point que l'on serait tenté de considérer l'anse comme un ensemble unidimensionnel. Pour illustrer cela, il suffit de penser à une rivière qui se déverse dans un océan : sa largeur étant beaucoup plus petite que les dimensions de l'océan, la rivière semble unidimensionnelle à l'échelle du plan d'eau.

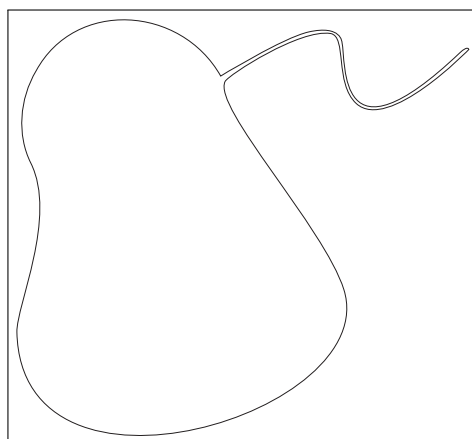


FIGURE 4.1 – Domaine à anse fine.

La difficulté principale liée au traitement numérique d'un tel domaine est la qualité de sa triangulation, puisque le fait de devoir utiliser des mailles très fines pour trianguler l'anse (et donc son voisinage) génère un très grand nombre de nœuds. Et même un processus de maillage séparé pour les deux parties du domaine, couplé à des procédures de raffinement et de recollement de triangulations, ne permet d'apporter une solution relativement satisfaisante que sur la partie épaisse du domaine (voir figure 4.2). On peut donc se demander s'il est possible de considérer l'anse comme un domaine (essentiellement) unidimensionnel et/ou d'effectuer la résolution numérique complète (maillage et calcul du spectre) de ces deux parties du domaine séparément.

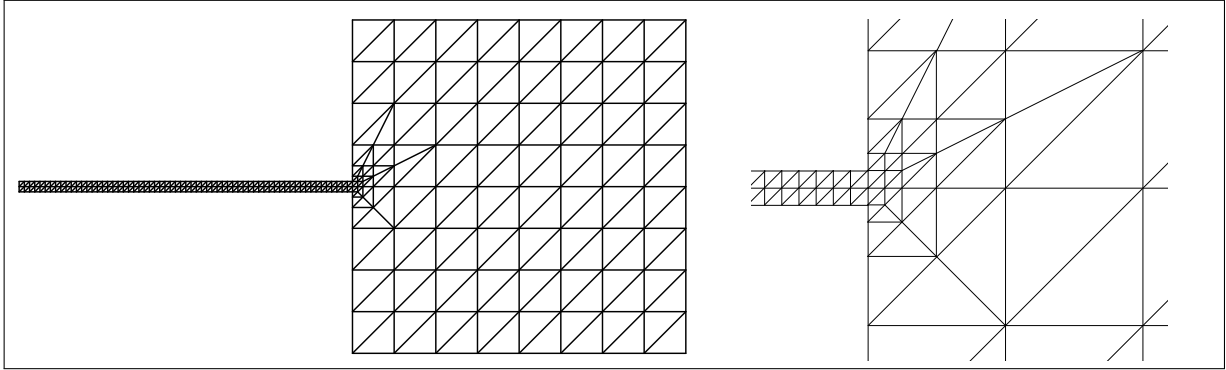


FIGURE 4.2 – Maillage d'un domaine à anse fine.

Malgré ces difficultés, des expériences numériques ont été effectuées à l'aide des algorithmes présentés dans ce travail, pour des anses raisonnablement petites par rapport au reste du domaine (voir la section 6.4), tant pour des domaines de \mathbb{R}^2 que pour des domaines de surfaces, à anses cylindriques par exemple. Les résultats numériques obtenus sont très satisfaisants et rejoignent les résultats théoriques d'Anné pour les sous-variétés ([4]), liant les spectres des domaines à anses fines, des domaines privés de leurs anses et de domaines essentiellement unidimensionnels correspondant aux anses.

Le résultat théorique présenté dans ce chapitre concerne essentiellement les domaines du plan et confirme la convergence observée lors des expériences numériques préliminaires. L'approche proposée emploie la formulation variationnelle du problème aux valeurs propres. La preuve n'a toutefois pas pu être adaptée au cas des anses cylindriques, bien qu'une convergence similaire ait été observée expérimentalement.

4.1 Énoncé du théorème

Soient Ω_1 un domaine de \mathbb{R}^2 , x_0 un point de $\partial\Omega_1$ et Ω_2 un domaine de \mathbb{R}^2 vérifiant, si Π_1 dénote la projection canonique de \mathbb{R}^2 sur Ox_1 et δ et L sont deux réels strictement positifs, $\Pi_1(\Omega_2) = [-\delta, L]$. Soit, pour tout $\varepsilon > 0$,

$$\begin{aligned} \psi^\varepsilon : \quad \Omega_2 &\longrightarrow \mathbb{R}^2 \\ (x_1, x_2) &\longmapsto (x_1, \varepsilon x_2) \end{aligned}$$

l'écrasement de Ω_2 sur Ox_1 . Soit $\gamma : [-\delta, L] \longrightarrow \mathbb{R}^2$ une courbe injective, régulière (\mathcal{C}^2 par morceaux), paramétrée par longueur d'arc, de courbure bornée, telle que $\gamma(0) = x_0$, $\gamma([-\delta, L]) \cap \Omega_1 = \gamma([-\delta, 0])$, et $\gamma'(0) \nparallel \partial\Omega_1$. Soit

$$\begin{aligned} h : \quad [-\delta, L] \times \mathbb{R} &\longrightarrow \mathbb{R}^2 \\ (x_1, x_2) &\longmapsto \gamma(x_1) + x_2 n(x_1), \end{aligned}$$

avec $n = (-\gamma'_2, \gamma'_1)$, et soit, pour tout $\varepsilon > 0$,

$$h^\varepsilon = h \circ \psi^\varepsilon : \quad \begin{array}{l} \Omega_2 \longrightarrow \mathbb{R}^2 \\ (x_1, x_2) \longmapsto \gamma(x_1) + \varepsilon x_2 n(x_1) \end{array}$$

l'écrasement sur $Im(\gamma)$. Nous verrons que, pour ε petit, h^ε est un difféomorphisme sur son image Ω_2^ε .

Notons encore $\Gamma^\varepsilon = \partial\Omega_1 \cap \Omega_2^\varepsilon$, $\Gamma = (\{0\} \times \mathbb{R}) \cap \Omega_2$ et $\Omega^\varepsilon = \Omega_1 \cup \Omega_2^\varepsilon$.

On considère les décompositions $\partial\Omega_1 = \partial\Omega_1^D \amalg \partial\Omega_1^N$, $\partial\Omega_2 = \partial\Omega_2^D \amalg \partial\Omega_2^N$, ainsi que $\partial\Omega^{\varepsilon,D} = \partial\Omega^\varepsilon \cap (\partial\Omega_1^D \cup h^\varepsilon(\partial\Omega_2^D))$ et $\partial\Omega^{\varepsilon,N} = \partial\Omega^\varepsilon \cap (\partial\Omega_1^N \cup h^\varepsilon(\partial\Omega_2^N))$.

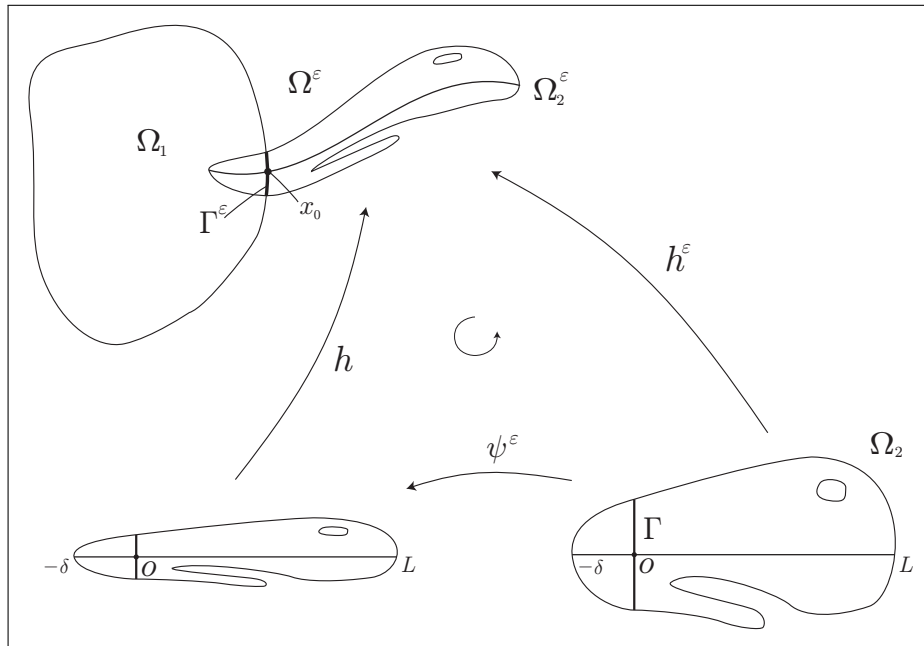


FIGURE 4.3 – Définition de l'anse.

Considérons encore les espaces suivants :

$$V(\Omega^\varepsilon) = \{v \in H^1(\Omega^\varepsilon) : v = 0 \text{ sur } \partial\Omega^{\varepsilon,D}\},$$

$$V(\Omega_1) = \{v \in H^1(\Omega_1) : v = 0 \text{ sur } \partial\Omega_1^D\},$$

$$V_0(\Omega_2) = \left\{ v \in H^1(\Omega_2) : v = 0 \text{ sur } \partial\Omega_2^D \text{ et dans } (\mathbb{R}^- \times \mathbb{R}) \cap \Omega_2, \frac{\partial v}{\partial x_2} = 0 \text{ dans } \Omega_2 \right\}.$$

On considère finalement les problèmes suivants (pour $\varepsilon > 0$ petit) :

$$(P_\varepsilon) \quad \begin{cases} \text{Trouver } u^\varepsilon \in V(\Omega^\varepsilon) \text{ et } \lambda^\varepsilon \in \mathbb{R} \text{ tels que, pour tout } v^\varepsilon \in V(\Omega^\varepsilon) : \\ \int_{\Omega^\varepsilon} \langle \nabla u^\varepsilon, \nabla v^\varepsilon \rangle = \lambda^\varepsilon \int_{\Omega^\varepsilon} u^\varepsilon v^\varepsilon, \end{cases}$$

$$(P_0) \quad \begin{cases} \text{Trouver } (u_1, u_2) \in V(\Omega_1) \times V_0(\Omega_2) \text{ et } \lambda \in \mathbb{R} \text{ tels que,} \\ \text{pour tout } (v_1, v_2) \in V(\Omega_1) \times V_0(\Omega_2) : \\ \int_{\Omega_1} \langle \nabla u_1, \nabla v_1 \rangle = \lambda \int_{\Omega_1} u_1 v_1 \quad \text{et} \quad \int_{\Omega_2} \frac{\partial u_2}{\partial x_1} \frac{\partial v_2}{\partial x_1} = \lambda \int_{\Omega_2} u_2 v_2 . \end{cases}$$

On va voir le résultat suivant :

Théorème 10

A) *Convergence des valeurs propres* : les solutions λ^ε de (P_ε) convergent vers les solutions λ de (P_0) lorsque $\varepsilon \rightarrow 0$.

B) *Convergence des espaces propres* : on peut choisir des bases $\{u^{\varepsilon,k}\}_{k>0}$ de solutions de (P_ε) et $\{u_1^k, u_2^k\}_{k>0}$ de solutions de (P_0) telles que :

a) $u^{\varepsilon,k}|_{\Omega_1} \rightarrow u_1^k$ dans $H^1(\Omega_1)$ lorsque $\varepsilon \rightarrow 0$.

b) $\frac{\partial}{\partial x_2}(\varepsilon^{1/2} u^{\varepsilon,k} \circ h^\varepsilon) \rightarrow \frac{\partial u_2^k}{\partial x_2}$ dans $L^2(\Omega_2)$ lorsque $\varepsilon \rightarrow 0$.

c) $\varepsilon^{1/2} u^{\varepsilon,k} \circ h^\varepsilon \rightarrow u_2^k$ dans $H^1(\Omega_2)$ lorsque $\varepsilon \rightarrow 0$.

d) En particulier, $\varepsilon^{1/2} u^{\varepsilon,k} \circ h^\varepsilon \rightarrow u_2^k$ dans $L^2(\Omega_2)$ lorsque $\varepsilon \rightarrow 0$.

Au delà du théorème lui-même, un point intéressant est d'identifier les solutions de (P_0) , en particulier celles correspondant à l'anse. En effet, bien que les fonctions limites soient constantes orthogonalement à l'anse, le problème limite peut être relativement complexe selon la géométrie (et la topologie) de l'anse, comme le montrent les quelques exemples ci-dessous.

Dans le cas où $\Omega_2 = [-\delta, L] \times [-l, l]$ avec $l > 0$ et où les conditions sont de Neumann sur Ω_2 (i.e. $\partial\Omega_2^N = \partial\Omega_2$), l'identification est triviale : il s'agit des solutions du problème

$$\begin{cases} \text{Trouver } (u_1, u_2) \in V(\Omega_1) \times V \text{ et } \lambda \in \mathbb{R} \text{ tels que,} \\ \text{pour tout } (v_1, v_2) \in V(\Omega_1) \times V : \\ \int_{\Omega_1} \langle \nabla u_1, \nabla v_1 \rangle = \lambda \int_{\Omega_1} u_1 v_1 \quad \text{et} \quad \int_0^L u_2' v_2' = \lambda \int_0^L u_2 v_2 , \end{cases}$$

où $V = \{v \in H^1([0, L]) : v(0) = 0\}$.

A noter que des conditions de Dirichlet sur le bord de l'anse ont pour effet de saucissonner complètement le problème sur l'anse (voir de l'évacuer totalement si des conditions de Dirichlet apparaissent sur toute la longueur). Considérons par exemple la même anse qu'avant, soit $\Omega_2 = [-\delta, L] \times [-l, l]$ avec $l > 0$, mais munie de conditions de Dirichlet sur les parties $\partial\Omega_2^D$ de $\partial\Omega_2$ définies par la figure ci-dessous.

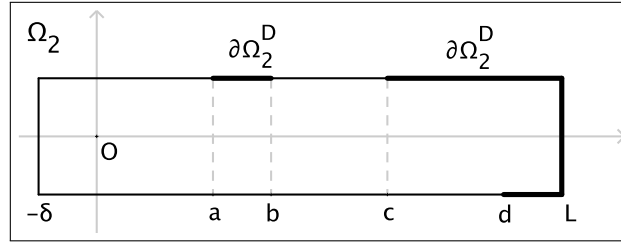


FIGURE 4.4 – Anse munie de conditions mixtes.

On peut alors identifier les solutions limites à celles du problème

$$\left\{ \begin{array}{l} \text{Trouver } (u_1, u_{21}, u_{22}) \in V(\Omega_1) \times H_0^1([0, a]) \times H_0^1([b, c]) \text{ et } \lambda \in \mathbb{R} \text{ tels que,} \\ \text{pour tout } (v_1, v_{21}, v_{22}) \in V(\Omega_1) \times H_0^1([0, a]) \times H_0^1([b, c]) : \\ \int_{\Omega_1} \langle \nabla u_1, \nabla v_1 \rangle = \lambda \int_{\Omega_1} u_1 v_1, \int_0^a u'_{21} v'_{21} = \lambda \int_0^a u_{21} v_{21} \text{ et } \int_b^c u'_{22} v'_{22} = \lambda \int_b^c u_{22} v_{22}, \end{array} \right.$$

la fonction limite sur l'anse étant nulle sur $([a, b] \cup [c, L]) \times [-l, l]$.

Dans le cas où, pour tout $x_1 \in [0, L]$, la coupe $\Omega_2 \cap (\{x_1\} \times \mathbb{R})$ est connexe mais où la largeur de l'anse varie le long de l'axe $x_2 = 0$, on voit que l'identification peut se faire de la même manière mais en modifiant la mesure sur $[0, L]$, les parties plus épaisses de l'anse ayant plus de poids que les parties plus fines.

Un exemple beaucoup moins trivial apparaît lorsque les coupes $\Omega_2 \cap (\{x_1\} \times \mathbb{R})$ ne sont plus toutes connexes. Pour illustrer cela, considérons une anse non simplement connexe, telle que représentée ci-dessous, munie de conditions au bord de Neumann.

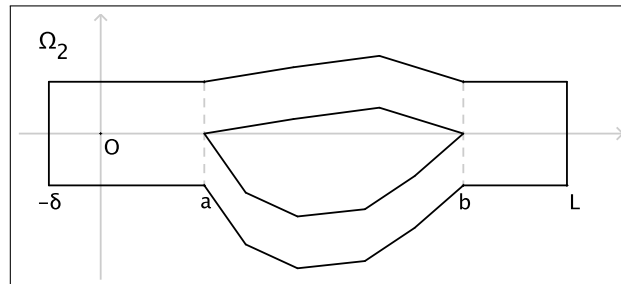


FIGURE 4.5 – Anse non simplement connexe.

Les solutions limites sont alors identifiables à celles du problème

$$\left\{ \begin{array}{l} \text{Trouver } (u_1, u_{21}, u_{221}, u_{222}, u_{23}) \in V(\Omega_1) \times V \times H^1([a, b]) \times H^1([a, b]) \times H^1([b, L]) \\ \text{et } \lambda \in \mathbb{R} \text{ tels que,} \\ \text{pour tout } (v_1, v_{21}, v_{221}, v_{222}, v_{23}) \in V(\Omega_1) \times V \times H^1([a, b]) \times H^1([a, b]) \times H^1([b, L]) \\ \text{et pour tout } i \in \{1, 2\} : \\ \int_{\Omega_1} \langle \nabla u_1, \nabla v_1 \rangle = \lambda \int_{\Omega_1} u_1 v_1 , \\ \int_0^a u'_{21} v'_{21} = \lambda \int_0^a u_{21} v_{21} , \int_a^b u'_{22i} v'_{22i} = \lambda \int_a^b u_{21i} v_{22i} \text{ et } \int_b^L u'_{23} v'_{23} = \lambda \int_b^L u_{23} v_{23} , \\ \text{avec } u_{21}(a) = u_{22i}(a) \text{ et } u_{22i}(b) = u_{23}(b) , \end{array} \right.$$

où $V = \{v \in H^1([0, a]) : v(0) = 0\}$.

Comme on l'entrevoit, les problèmes limites peuvent rapidement atteindre une certaine complexité; on laisse d'ailleurs au lecteur motivé l'identification du problème limite correspondant à l'anse de la figure 4.3 (par exemple sous condition de Neumann), dont les coupes, de largeur non constante, se séparent en plusieurs endroits! Les problèmes d'anses fines permettent donc de construire une multitude d'exemples de spectres.

Il serait également envisageable de développer un algorithme de résolution numérique de problèmes à anses fines, bien que cela n'ait pas été effectué dans ce travail. On remarque d'ailleurs que les problèmes d'identification pourraient facilement être résolu par des procédés proches de ce qui a été fait dans le cas de surfaces paramétrées par plusieurs cartes; de même, la prise en compte de différentes mesures ne pose pas davantage de difficultés que les métriques des surfaces.

4.2 Preuve du théorème sous hypothèses fortes

Nous allons dans un premier temps démontrer le théorème dans un cas plus simple, afin de dissocier les difficultés techniques et les difficultés fondamentales. Aussi, nous considérerons tout d'abord le cas où la courbe γ définit un segment de droite (disons $\gamma(t) = (t, 0)$) et est perpendiculaire à $\partial\Omega_1$, celui-ci étant rectiligne localement de part et d'autre de x_0 (i.e. $\Gamma^\varepsilon = h^\varepsilon(\Gamma) \subset Ox_2$). Dans ce cas, l'application h est l'injection canonique de Ω_2 dans \mathbb{R}^2 et $\psi^\varepsilon = h^\varepsilon$ envoie (x_1, x_2) sur $(x_1, \varepsilon x_2)$.

Dans un deuxième temps, nous passerons en revue les incidences sur la preuve qu'implique l'affaiblissement de ces hypothèses (section 4.3) et remarquerons qu'il est possible d'obtenir un résultat similaire lorsque Ω^ε est muni d'une métrique lisse indépendante de ε (section 4.4).

Voici de manière synthétique les étapes de la preuve du théorème sous hypothèses fortes :

1. Convergence d'une sous-suite de valeurs propres.
2. Convergence faible d'une sous-suite de fonctions propres.
3. Identification du problème limite sur Ω_1 .
4. Convergence forte des fonctions propres sur Ω_1 .
5. Identification du problème limite sur Ω_2 .
6. Convergence forte des fonctions propres sur Ω_2 .
7. Identification des solutions limites.
8. Convergence de la suite entière de valeurs et fonctions propres.

Adoptons encore les quelques notations supplémentaires suivantes, utiles pour la preuve :

$$\begin{aligned} V^\varepsilon(\Omega_1) &= \{v \in H^1(\Omega_1) : v = 0 \text{ sur } \partial\Omega_1^D \cap \partial\Omega^\varepsilon\}, \\ V(\Omega_2^\varepsilon) &= \{v \in H^1(\Omega_2^\varepsilon) : v = 0 \text{ sur } \partial\Omega_2^{\varepsilon,D}\}, \\ V(\Omega_2) &= \{v \in H^1(\Omega_2) : v = 0 \text{ sur } \partial\Omega_2^D\}, \end{aligned}$$

$$\partial\Omega_2^{\varepsilon,D} = \psi^\varepsilon(\partial\Omega_2^D), \quad \partial\Omega_2^{\varepsilon,N} = \psi^\varepsilon(\partial\Omega_2^N), \quad \partial\Omega_1^{\varepsilon,D} = \partial\Omega^\varepsilon \cap \partial\Omega_1^D \quad \text{et} \quad \partial\Omega_1^{\varepsilon,N} = \partial\Omega^\varepsilon \cap \partial\Omega_1^N.$$

4.2.1 Convergence d'une sous-suite de valeurs propres

Lemme 2 *Pour tout $k \geq 1$, la famille de valeurs propres $\{\lambda^{\varepsilon,k}\}_{\varepsilon>0}$ est bornée.*

Preuve

Soient f_1, \dots, f_k k premières fonctions propres de Dirichlet d'une boule fermée B contenue dans Ω_1 et V_k l'espace vectoriel engendré par ces fonctions. Pour toute fonction $f \in V_k$, notons \bar{f}^ε son prolongement par zéro sur Ω^ε . Alors on a par le théorème du Min-Max

$$\lambda_k(\Omega^\varepsilon) \leq \sup_{f \in V_k} QR(\bar{f}^\varepsilon) = \sup_{f \in V_k} QR(f) = \lambda_k(B).$$

Donc la famille $\{\lambda_k(\Omega^\varepsilon)\}_{\varepsilon>0} = \{\lambda^{\varepsilon,k}\}_{\varepsilon>0}$ est bornée. □

On peut donc extraire de chaque famille $\{\lambda^{\varepsilon,k}\}_{\varepsilon>0}$ une sous-famille convergente. Dans la suite, nous ne considérerons que cette sous-famille, sans notation supplémentaire; nous verrons à la fin du développement qu'en fait toute la famille converge.

4.2.2 Convergence faible d'une sous-suite de fonctions propres

Notons, pour tout $\varepsilon > 0$, $\{u^{\varepsilon,k}\}$ une famille totale de fonctions propres de Ω^ε vérifiant la relation d'orthonormalisation

$$\langle u^{\varepsilon,k}, u^{\varepsilon,l} \rangle_{L^2(\Omega^\varepsilon)} = \varepsilon^{2p} \delta_{kl},$$

où p sera déterminé ultérieurement. On a donc

$$\langle u^{\varepsilon,k}, u^{\varepsilon,l} \rangle_{1,\Omega^\varepsilon} = \varepsilon^{2p} \lambda^{\varepsilon,k} \delta_{kl}.$$

Notons encore $(u_1^{\varepsilon,k}, u_2^{\varepsilon,k}) = \varphi^\varepsilon(u^{\varepsilon,k})$, où

$$\begin{aligned} \varphi^\varepsilon : V(\Omega^\varepsilon) &\longrightarrow V^\varepsilon(\Omega_1) \times V(\Omega_2) \\ u &\longmapsto (u_1, u_2) \end{aligned}$$

est définie par $u_1 = u|_{\Omega_1}$ et $u_2 = \varepsilon^q u \circ h^\varepsilon$ et où q sera déterminé ultérieurement.

Lemme 3 *Pour tout $k \geq 1$, la famille $\{(u_1^{\varepsilon,k}, u_2^{\varepsilon,k})\}_{\varepsilon > 0}$ est bornée dans $H^1(\Omega_1) \times H^1(\Omega_2)$ si $p \geq 0$ et $p + q \geq 1/2$. Dans ce cas, pour tout $\delta > 0$ et pour tout $k \geq 1$, la famille $\{(u_1^{\varepsilon,k}, u_2^{\varepsilon,k})\}_{\varepsilon > \delta}$ appartient à $V^\delta(\Omega_1) \times V(\Omega_2)$, i.e. $u_1^{\varepsilon,k}$ est nulle sur $\partial\Omega_1^{\delta,D}$ et $u_2^{\varepsilon,k}$ sur $\partial\Omega_2^D$.*

Preuve

On a

$$\left\| (u_1^{\varepsilon,k}, u_2^{\varepsilon,k}) \right\|_{H^1(\Omega_1) \times H^1(\Omega_2)}^2 = \left\| u_1^{\varepsilon,k} \right\|_{H^1(\Omega_1)}^2 + \left\| u_2^{\varepsilon,k} \right\|_{H^1(\Omega_2)}^2,$$

avec

$$\begin{aligned} \left\| u_1^{\varepsilon,k} \right\|_{H^1(\Omega_1)}^2 &= \int_{\Omega_1} |\nabla u_1^{\varepsilon,k}|^2 + \int_{\Omega_1} (u_1^{\varepsilon,k})^2 \leq \int_{\Omega^\varepsilon} |\nabla u^{\varepsilon,k}|^2 + \int_{\Omega^\varepsilon} (u^{\varepsilon,k})^2 \\ &= (1 + \lambda^{\varepsilon,k}) \int_{\Omega^\varepsilon} (u^{\varepsilon,k})^2, \\ \left\| u_2^{\varepsilon,k} \right\|_{H^1(\Omega_2)}^2 &= \int_{\Omega_2} |\nabla u_2^{\varepsilon,k}|^2 + \int_{\Omega_2} (u_2^{\varepsilon,k})^2 \\ &\leq \int_{\Omega_2} \left(\left(\frac{\partial u_2^{\varepsilon,k}}{\partial x_1} \right)^2 + \frac{1}{\varepsilon^2} \left(\frac{\partial u_2^{\varepsilon,k}}{\partial x_2} \right)^2 \right) + \int_{\Omega_2} (u_2^{\varepsilon,k})^2 \\ &= \varepsilon^{2q-1} \left(\int_{\Omega_2^\varepsilon} |\nabla u^{\varepsilon,k}|^2 + \int_{\Omega_2^\varepsilon} (u^{\varepsilon,k})^2 \right) \leq \varepsilon^{2q-1} \left(\int_{\Omega^\varepsilon} |\nabla u^{\varepsilon,k}|^2 + \int_{\Omega^\varepsilon} (u^{\varepsilon,k})^2 \right) \\ &= \varepsilon^{2q-1} (1 + \lambda^{\varepsilon,k}) \int_{\Omega^\varepsilon} (u^{\varepsilon,k})^2. \end{aligned}$$

Ainsi, on trouve

$$\left\| \left(u_1^{\varepsilon,k}, u_2^{\varepsilon,k} \right) \right\|_{H^1(\Omega_1) \times H^1(\Omega_2)}^2 \leq (1 + \varepsilon^{2q-1}) (1 + \lambda^{\varepsilon,k}) \int_{\Omega^\varepsilon} (u^{\varepsilon,k})^2 = (1 + \varepsilon^{2q-1}) (1 + \lambda^{\varepsilon,k}) \varepsilon^{2p}$$

qui est borné si $p \geq 0$ et $p + q \geq 1/2$, puisque la famille $\{\lambda^{\varepsilon,k}\}_{\varepsilon>0}$ l'est, par le lemme 2. \square

Proposition 4 *Si $p \geq 0$ et $p + q \geq 1/2$, il existe une sous-famille de $\{\varepsilon > 0\}$, toujours notée $\{\varepsilon > 0\}$, telle que, pour tout $k \geq 1$, la famille associée $\left\{ \left(\lambda^{\varepsilon,k}, u_1^{\varepsilon,k}, u_2^{\varepsilon,k} \right) \right\}_{\varepsilon>0}$ vérifie, lorsque $\varepsilon \rightarrow 0$,*

$$\lambda^{\varepsilon,k} \rightarrow \lambda^k \text{ dans } \mathbb{R},$$

$$\left(u_1^{\varepsilon,k}, u_2^{\varepsilon,k} \right) \rightarrow \left(u_1^k, u_2^k \right) \text{ dans } H^1(\Omega_1) \times H^1(\Omega_2).$$

Preuve

Cette proposition est un corollaire direct des lemmes 2 et 3 (on procède par extractions successives). \square

Dans la suite, nous ne considérerons que cette sous-famille, sans notation supplémentaire ; nous verrons à la fin du développement qu'en fait toute la famille converge. Nous nous permettrons de plus d'omettre parfois les indices k des fonctions et valeurs propres, pour plus de lisibilité. Dès maintenant, fixons $p = 0$ et $q = 1/2$, soit les valeurs minimales de p et q pour que la proposition 4 assure la convergence. La raison à cela est que, sinon, $u_1^{\varepsilon,k}$ convergerait vers 0 dans $H^1(\Omega_1)$, ce qui ne permettrait pas d'identifier de problème limite sur Ω_1 .

4.2.3 Identification du problème limite sur Ω_1

Distinguons deux cas.

1er cas : il existe $\varepsilon > 0$ tel que $\Gamma^\varepsilon \subset \partial\Omega_1^D$, i.e. les conditions au bord de Ω_1 à l'interface sont de Dirichlet, ou encore $\partial\Omega_1^{\varepsilon,N} = \partial\Omega_1^N$.

Alors on a, pour tout $v_1 \in H^1(\Omega_1)$

$$\begin{aligned} \left| \int_{\partial\Omega_1^D} u_1 v_1 \right| &= \lim_{\varepsilon \rightarrow 0} \left| \int_{\partial\Omega_1^D} u_1^\varepsilon v_1 \right| \leq \lim_{\varepsilon \rightarrow 0} \left| \int_{\partial\Omega_1^D \setminus \Gamma^\varepsilon} u_1^\varepsilon v_1 \right| + \lim_{\varepsilon \rightarrow 0} \left| \int_{\Gamma^\varepsilon} u_1^\varepsilon v_1 \right| \\ &= \lim_{\varepsilon \rightarrow 0} \left| \int_{\Gamma^\varepsilon} u_1^\varepsilon v_1 \right| \stackrel{(C-S)}{\leq} \lim_{\varepsilon \rightarrow 0} \|u_1^\varepsilon\|_{L^2(\Gamma^\varepsilon)} \lim_{\varepsilon \rightarrow 0} \|v_1\|_{L^2(\Gamma^\varepsilon)} \\ &\leq C \lim_{\varepsilon \rightarrow 0} \|u_1^\varepsilon\|_{H^1(\Omega_1)} \lim_{\varepsilon \rightarrow 0} \|v_1\|_{L^2(\Gamma^\varepsilon)} = 0, \end{aligned}$$

puisque $\|u_1^\varepsilon\|_{H^1(\Omega_1)}$ est borné (par le lemme 3).

Autrement dit, $u_1 \in V(\Omega_1)$. Ainsi, u_1 vérifie, pour tout $v_1 \in V(\Omega_1)$,

$$\begin{aligned} \left| \int_{\Omega_1} \langle \nabla u_1, \nabla v_1 \rangle - \lambda \int_{\Omega_1} u_1 v_1 \right| &= \left| \lim_{\varepsilon \rightarrow 0} \left(\int_{\Omega_1} \langle \nabla u_1^\varepsilon, \nabla v_1 \rangle - \lambda^\varepsilon \int_{\Omega_1} u_1^\varepsilon v_1 \right) \right| \\ &= \left| \lim_{\varepsilon \rightarrow 0} \int_{\partial\Omega_1} \frac{\partial u_1^\varepsilon}{\partial n} v_1 \right| \\ &\leq \left| \lim_{\varepsilon \rightarrow 0} \int_{\partial\Omega_1^D} \frac{\partial u_1^\varepsilon}{\partial n} v_1 \right| + \left| \lim_{\varepsilon \rightarrow 0} \int_{\partial\Omega_1^N} \frac{\partial u_1^\varepsilon}{\partial n} v_1 \right| \\ &= \left| \lim_{\varepsilon \rightarrow 0} \int_{\partial\Omega_1^{\varepsilon, N}} \frac{\partial u_1^\varepsilon}{\partial n} v_1 \right| = 0, \end{aligned}$$

donc u_1 est solution au problème

$$(P_1) \quad \begin{cases} \text{Trouver } u_1 \in V(\Omega_1) \text{ et } \lambda \in \mathbb{R} \text{ tels que, pour tout } v_1 \in V(\Omega_1) : \\ \int_{\Omega_1} \langle \nabla u_1, \nabla v_1 \rangle = \lambda \int_{\Omega_1} u_1 v_1 . \end{cases}$$

2e cas : il existe $\varepsilon > 0$ tel que $\Gamma^\varepsilon \subset \partial\Omega_1^N$, i.e. les conditions au bord de Ω_1 à l'interface sont de Neumann, ou encore $\partial\Omega_1^{\varepsilon, D} = \partial\Omega_1^D$.

Alors on a, pour tout $v_1 \in V(\Omega_1)$,

$$\begin{aligned} \int_{\Omega_1} \langle \nabla u_1, \nabla v_1 \rangle - \lambda \int_{\Omega_1} u_1 v_1 &= \lim_{\varepsilon \rightarrow 0} \left(\int_{\Omega_1} \langle \nabla u_1^\varepsilon, \nabla v_1 \rangle - \lambda^\varepsilon \int_{\Omega_1} u_1^\varepsilon v_1 \right) \\ &= - \lim_{\varepsilon \rightarrow 0} \int_{\partial\Omega_1} \frac{\partial u_1^\varepsilon}{\partial n} v_1 \\ &= - \lim_{\varepsilon \rightarrow 0} \int_{\partial\Omega_1 \setminus \Gamma^\varepsilon} \frac{\partial u_1^\varepsilon}{\partial n} v_1 - \lim_{\varepsilon \rightarrow 0} \int_{\Gamma^\varepsilon} \frac{\partial u_1^\varepsilon}{\partial n} v_1 \\ &= - \lim_{\varepsilon \rightarrow 0} \int_{\partial\Omega_1^D} \frac{\partial u_1^\varepsilon}{\partial n} v_1 - \lim_{\varepsilon \rightarrow 0} \int_{\partial\Omega_1^{\varepsilon, N}} \frac{\partial u_1^\varepsilon}{\partial n} v_1 - \lim_{\varepsilon \rightarrow 0} \int_{\Gamma^\varepsilon} \frac{\partial u_1^\varepsilon}{\partial n} v_1 \\ &= - \lim_{\varepsilon \rightarrow 0} \int_{\Gamma^\varepsilon} \frac{\partial u_1^\varepsilon}{\partial n} v_1 . \end{aligned}$$

Voyons que ce dernier terme est nul. Notons

$$\begin{aligned} l : V(\Omega_1) &\longrightarrow \mathbb{R} \\ v_1 &\longmapsto l(v_1) = - \lim_{\varepsilon \rightarrow 0} \int_{\Gamma^\varepsilon} \frac{\partial u_1^\varepsilon}{\partial n} v_1 . \end{aligned}$$

Par le calcul précédent, l est bien définie, linéaire et continue. Par Riesz, il existe $w \in V(\Omega_1)$ tel que, pour tout $v_1 \in V(\Omega_1)$, $l(v_1) = \langle v_1, w \rangle_{V(\Omega_1)}$.

Mais alors, par définition de l , on a $l(v_1) = 0$ pour tout $v_1 \in V(\Omega_1)$ tel qu'il existe $\varepsilon > 0$ avec v_1 nulle sur $B(x_0, \varepsilon) \cap \Omega_1$, i.e. $\langle v_1, w \rangle_{V(\Omega_1)} = 0$ pour tout $v_1 \in V(\Omega_1)$ tel que $\text{Supp}(v_1)$ ne contient pas x_0 . Donc $w = 0$ presque partout sur Ω_1 , d'où $w = 0$ dans $V(\Omega_1)$ et donc $l = 0$.

Ainsi, u_1 est solution au même problème (P_1) que dans le premier cas.

4.2.4 Convergence forte des fonctions propres sur Ω_1

On a vu que u_1^ε converge faiblement dans $H^1(\Omega_1)$ vers u_1 . Ainsi, par Rellich, u_1^ε converge fortement dans $L^2(\Omega_1)$ vers u_1 ; voyons encore que u_1^ε converge fortement dans $H^1(\Omega_1)$ vers u_1 .

On a

$$\|u_1^\varepsilon\|_{H^1(\Omega_1)}^2 = \|\nabla u_1^\varepsilon\|_{L^2(\Omega_1)}^2 + \|u_1^\varepsilon\|_{L^2(\Omega_1)}^2 = (1 + \lambda^\varepsilon) \|u_1^\varepsilon\|_{L^2(\Omega_1)}^2 + \int_{\Gamma^\varepsilon} \frac{\partial u_1^\varepsilon}{\partial n} u_1^\varepsilon.$$

Or, $\int_{\Gamma^\varepsilon} \frac{\partial u_1^\varepsilon}{\partial n} u_1^\varepsilon \rightarrow 0$ lorsque $\varepsilon \rightarrow 0$, d'où

$$\|u_1^\varepsilon\|_{H^1(\Omega_1)}^2 \xrightarrow{\varepsilon \rightarrow 0} (1 + \lambda) \|u_1\|_{L^2(\Omega_1)}^2 = \|u_1\|_{H^1(\Omega_1)}^2.$$

Ainsi, on a convergence faible ainsi que convergence des normes, ce qui implique la convergence forte.

4.2.5 Identification du problème limite sur Ω_2

Lemme 4 *La limite $\frac{\partial u_2}{\partial x_2}$ dans $L^2(\Omega_2)$ de $\frac{\partial u_2^\varepsilon}{\partial x_2}$ est la fonction identiquement nulle. En particulier, u_2 ne dépend pas de x_2 .*

Preuve

On a

$$\begin{aligned} \left\| \frac{\partial u_2}{\partial x_2} \right\|_{L^2(\Omega_2)}^2 &\leq \liminf_{\varepsilon \rightarrow 0} \left\| \frac{\partial u_2^\varepsilon}{\partial x_2} \right\|_{L^2(\Omega_2)}^2 \\ &\leq \liminf_{\varepsilon \rightarrow 0} \left(\varepsilon^2 \int_{\Omega_2} \left(\left(\frac{\partial u_2^\varepsilon}{\partial x_1} \right)^2 + \frac{1}{\varepsilon^2} \left(\frac{\partial u_2^\varepsilon}{\partial x_2} \right)^2 \right) \right) = 0. \end{aligned}$$

□

Lemme 5 *La limite $u_2|_\Gamma$ dans $H^{1/2}(\Gamma) \subset L^2(\Gamma)$ de $u_2^\varepsilon|_\Gamma$ est la fonction identiquement nulle.*

Preuve

Le temps de la preuve, notons $\Omega_2^* = \{(x_1, x_2) \in \Omega_2 : x_1 < 0\}$. On a

$$\begin{aligned} \|u_2^\varepsilon\|_{H^1(\Omega_2^*)}^2 &= \int_{\Omega_2^*} |\nabla u_2^\varepsilon|^2 + \int_{\Omega_2^*} (u_2^\varepsilon)^2 = \varepsilon \int_{\Omega_2^*} |\nabla u_1^\varepsilon \circ h^\varepsilon|^2 + \varepsilon \int_{\Omega_2^*} (u_1^\varepsilon \circ h^\varepsilon)^2 \\ &\leq \int_{h^\varepsilon(\Omega_2^*)} |\nabla u_1^\varepsilon|^2 + \int_{h^\varepsilon(\Omega_2^*)} (u_1^\varepsilon)^2. \end{aligned}$$

Ainsi, on a pour tout $\varepsilon_0 > 0$ fixé

$$\begin{aligned} \|u_2\|_{H^1(\Omega_2^*)}^2 &\leq \liminf_{\varepsilon \rightarrow 0} \|u_2^\varepsilon\|_{H^1(\Omega_2^*)}^2 \leq \lim_{\varepsilon \rightarrow 0} \left(\int_{h^\varepsilon(\Omega_2^*)} |\nabla u_1^\varepsilon|^2 + \int_{h^\varepsilon(\Omega_2^*)} (u_1^\varepsilon)^2 \right) \\ &\leq \lim_{\varepsilon \rightarrow 0} \left(\int_{h^{\varepsilon_0}(\Omega_2^*)} |\nabla u_1^\varepsilon|^2 + \int_{h^{\varepsilon_0}(\Omega_2^*)} (u_1^\varepsilon)^2 \right) \\ &= \int_{h^{\varepsilon_0}(\Omega_2^*)} |\nabla u_1|^2 + \int_{h^{\varepsilon_0}(\Omega_2^*)} u_1^2, \end{aligned}$$

car $u_1^\varepsilon \xrightarrow{\varepsilon \rightarrow 0} u_1$ dans $H^1(\Omega_1)$. Donc

$$\|u_2\|_{H^1(\Omega_2^*)} \leq \|u_1\|_{H^1(h^{\varepsilon_0}(\Omega_2^*))}$$

qui est aussi petit qu'on le veut, puisque c'est le cas de $h^{\varepsilon_0}(\Omega_2^*)$.

Ainsi, $u_2^\varepsilon \xrightarrow{\varepsilon \rightarrow 0} 0$ dans $H^1(\Omega_2^*)$, donc $u_2^\varepsilon \xrightarrow{\varepsilon \rightarrow 0} 0$ dans $H^{1/2}(\Gamma) \subset L^2(\Gamma)$. □

Ces deux lemmes permettent de déterminer le problème limite. En effet, ils montrent que u_2 appartient à $V_0(\Omega_2)$. On a de plus, pour tout $v_2 \in V_0(\Omega_2)$,

$$\int_{\Omega_2} \frac{\partial u_2}{\partial x_1} \frac{\partial v_2}{\partial x_1} - \lambda \int_{\Omega_2} u_2 v_2 = \lim_{\varepsilon \rightarrow 0} \left(\int_{\Omega_2} \left(\frac{\partial u_2^\varepsilon}{\partial x_1} \frac{\partial v_2}{\partial x_1} + \frac{1}{\varepsilon^2} \frac{\partial u_2^\varepsilon}{\partial x_2} \frac{\partial v_2}{\partial x_2} \right) - \lambda^\varepsilon \int_{\Omega_2} u_2^\varepsilon v_2 \right) = 0.$$

Ainsi, u_2 est solution au problème

$$(P_2) \quad \begin{cases} \text{Trouver } u_2 \in V_0(\Omega_2) \text{ et } \lambda \in \mathbb{R} \text{ tels que, pour tout } v_2 \in V_0(\Omega_2) : \\ \int_{\Omega_2} \langle \nabla u_2, \nabla v_2 \rangle = \lambda \int_{\Omega_2} u_2 v_2. \end{cases}$$

4.2.6 Convergence forte des fonctions propres sur Ω_2

On a vu que u_2^ε converge faiblement dans $H^1(\Omega_2)$ vers u_2 . Par Rellich, u_2^ε converge fortement dans $L^2(\Omega_2)$ vers u_2 . De plus, on a vu au lemme 4 que $\frac{\partial u_2^\varepsilon}{\partial x_2}$ converge fortement dans $L^2(\Omega_2)$ vers $\frac{\partial u_2}{\partial x_2}$.

4.2.7 Identification des solutions limites

On a vu que les $\{u^{\varepsilon,k}\}_{\varepsilon>0}$ et les $\{\lambda^{\varepsilon,k}\}_{\varepsilon>0}$ convergent vers des solutions du problème limite. Il s'agit maintenant de voir que toute solution du problème limite est bien limite des $\{u^{\varepsilon,k}\}_{\varepsilon>0}$ et $\{\lambda^{\varepsilon,k}\}_{\varepsilon>0}$.

Nous allons voir que c'est le cas, en construisant des suites de fonctions test sur Ω^ε , (presque) orthonormées et dont leurs quotients de Rayleigh convergent vers tous les λ^k . Le théorème du Min-Max permet alors de conclure.

Voyons tout d'abord les deux lemmes techniques suivants :

Lemme 6 *Soit D un domaine relativement compact de \mathbb{R}^2 et soient, pour tout $\varepsilon > 0$ petit, $B = B(a, \varepsilon)$ et $B' = B(a, \sqrt{\varepsilon})$ des boules centrées en $a \in \mathbb{R}^2$. On considère l'application $\chi : \mathbb{R}^2 \rightarrow [0, 1]$ définie par*

$$\chi(x) = \begin{cases} 0 & \text{si } x \in \overline{B} \\ -\frac{2}{\ln \varepsilon} (\ln |x - a| - \ln \varepsilon) & \text{si } x \in B' \setminus B \\ 1 & \text{si } x \notin B' . \end{cases}$$

Alors, si f est une fonction-test sur D , $\tilde{f} = \chi f$ en est aussi une, est nulle sur la boule \overline{B} et converge vers f dans $H^1(D)$ lorsque $\varepsilon \rightarrow 0$.

Lemme 7 *Soit E un espace vectoriel de dimension supérieure à k muni d'un produit scalaire $\langle \cdot, \cdot \rangle_E$ et d'une norme associée $\|\cdot\|_E$. Soit encore, pour tout $\varepsilon > 0$ petit, un système libre $\{v_1, \dots, v_k\} = \{v_1^\varepsilon, \dots, v_k^\varepsilon\}$ presque orthonormé, i.e. tel que $|\langle v_i, v_j \rangle_E - \delta_{ij}| < \varepsilon$. Alors, pour tout $j \leq k$, $\|w_j - v_j\|_E \xrightarrow{\varepsilon \rightarrow 0} 0$, où $\{w_1, \dots, w_k\} = \{w_1^\varepsilon, \dots, w_k^\varepsilon\}$ est le résultat de l'orthonormalisation de Gram-Schmidt de $\{v_1, \dots, v_k\}$.*

Preuve lemme 6

Notons $M = \max_D |f|$ et $N = \max_D |\nabla f|$. On a

$$\begin{aligned} \|f - \tilde{f}\|_{H^1(D)}^2 &= \int_D |f - \tilde{f}|^2 + \int_D |\nabla f - \nabla \tilde{f}|^2 \\ &= \int_{B'} |f - \tilde{f}|^2 + \int_{B'} |\nabla f - \nabla \tilde{f}|^2 \\ &= \int_B f^2 + \int_{B' \setminus B} |f - \tilde{f}|^2 + \int_B |\nabla f|^2 + \int_{B' \setminus B} |\nabla f - \nabla \tilde{f}|^2 \\ &\leq M^2 \pi \varepsilon + N^2 \pi \varepsilon^2 + \int_{B' \setminus B} |\nabla f - \nabla \tilde{f}|^2 . \end{aligned}$$

Et on a, pour $x \in B' \setminus B$,

$$\frac{\partial}{\partial x_i} \chi(x) = -\frac{2}{\ln \varepsilon} \frac{x_i - a_i}{|x - a|^2}$$

pour $i \in \{1, 2\}$, d'où

$$\nabla \tilde{f}(x) = -\frac{2}{\ln \varepsilon} (\ln |x - a| - \ln \varepsilon) \nabla f(x) - \frac{2}{\ln \varepsilon} \frac{f(x)}{|x - a|^2} (x - a).$$

Donc

$$\begin{aligned} \left| \nabla f(x) - \nabla \tilde{f}(x) \right| &\leq \left| \nabla f(x) + \frac{2}{\ln \varepsilon} (\ln |x - a| - \ln \varepsilon) \nabla f(x) \right| + \left| \frac{2}{\ln \varepsilon} \frac{f(x)}{|x - a|^2} (x - a) \right| \\ &\leq |\nabla f(x)| + \frac{2|f(x)|}{|\ln \varepsilon| |x - a|} \leq N + \frac{2M}{|\ln \varepsilon| |x - a|}. \end{aligned}$$

Ainsi,

$$\begin{aligned} \int_{B' \setminus B} \left| \nabla f - \nabla \tilde{f} \right|^2 &\leq \int_{B' \setminus B} \left(N^2 + \frac{4M^2}{(\ln \varepsilon |x - a|)^2} \right) = N^2 \pi \varepsilon + \frac{8\pi M^2}{(\ln \varepsilon)^2} (\ln \sqrt{\varepsilon} - \ln \varepsilon) \\ &\leq N^2 \pi \varepsilon - \frac{4\pi M^2}{\ln \varepsilon}, \end{aligned}$$

d'où

$$\left\| f - \tilde{f} \right\|_{H^1(D)}^2 \xrightarrow{\varepsilon \rightarrow 0} 0.$$

□

Preuve lemme 7

Par induction sur $j \geq 1$.

Ancrage ($j=1$) : on a

$$\|w_1 - v_1\|_E = \left| \frac{1}{\|v_1\|_E} - 1 \right| \|v_1\|_E \xrightarrow{\varepsilon \rightarrow 0} 0.$$

Supposons à présent que $\|w_i - v_i\|_E \xrightarrow{\varepsilon \rightarrow 0} 0$ pour tout $i < j$ et voyons que $\|w_j - v_j\|_E \xrightarrow{\varepsilon \rightarrow 0} 0$.

On a

$$\langle v_j, w_i \rangle_E = \langle v_j, v_i \rangle_E + \langle v_j, w_i - v_i \rangle_E \stackrel{C-S}{\leq} \langle v_j, v_i \rangle_E + \|v_j\|_E \|w_i - v_i\|_E \xrightarrow{\varepsilon \rightarrow 0} 0.$$

Donc

$$\sum_{i=1}^{j-1} \langle v_j, w_i \rangle_E w_i \xrightarrow{\varepsilon \rightarrow 0} 0$$

et ainsi

$$\|w_j - v_j\|_E \xrightarrow{\varepsilon \rightarrow 0} 0.$$

□

Distinguons à présent deux cas.

1er cas : soit λ une valeur propre du problème limite provenant de Ω_1 , i.e. telle qu'il existe $u_{1,\lambda} \in V(\Omega_1) \setminus \{0\}$ tel que, pour tout $v_1 \in V(\Omega_1)$,

$$\int_{\Omega_1} \langle \nabla u_{1,\lambda}, \nabla v_1 \rangle = \lambda \int_{\Omega_1} u_{1,\lambda} v_1.$$

Considérons alors $u_\lambda^\varepsilon : \Omega^\varepsilon \rightarrow \mathbb{R}$ définie par

$$u_\lambda^\varepsilon(x) = \begin{cases} 0 & \text{si } x \in \Omega_2^\varepsilon \\ u_{1,\lambda}^\varepsilon(x) & \text{si } x \in \Omega_1, \end{cases}$$

où $u_{1,\lambda}^\varepsilon$ est construite à partir de $u_{1,\lambda}$ et nulle sur $B(0, \varepsilon) \cap \Omega_1$ de la même manière que \tilde{f} est construite à partir de f dans le lemme 6. Alors $u_\lambda^\varepsilon \in V(\Omega^\varepsilon)$, avec

$$\|u_\lambda^\varepsilon\|_{H^1(\Omega^\varepsilon)} = \|u_\lambda^\varepsilon\|_{H^1(\Omega_1)} \xrightarrow{\varepsilon \rightarrow 0} \|u_{1,\lambda}\|_{H^1(\Omega_1)}.$$

On a par construction $u_\lambda^\varepsilon|_{\Omega_1} \xrightarrow{\varepsilon \rightarrow 0} u_{1,\lambda}$ dans $V(\Omega^\varepsilon)$, et on trouve donc $\lambda = \lim_{\varepsilon \rightarrow 0} QR(u_\lambda^\varepsilon)$ et $\lim_{\varepsilon \rightarrow 0} \langle u_{\lambda_1}^\varepsilon, u_{\lambda_2}^\varepsilon \rangle_{H^1(\Omega^\varepsilon)} = \delta_{\lambda_1, \lambda_2}$.

2e cas : soit λ une valeur propre du problème limite provenant de Ω_2 , i.e. telle qu'il existe $u_{2,\lambda} \in V_0(\Omega_2) \setminus \{0\}$ tel que, pour tout $v_2 \in V_0(\Omega_2)$,

$$\int_{\Omega_2} \frac{\partial u_{2,\lambda}}{\partial x_1} \frac{\partial v_2}{\partial x_1} = \lambda \int_{\Omega_1} u_{2,\lambda} v_2.$$

Considérons alors $u_\lambda^\varepsilon : \Omega^\varepsilon \rightarrow \mathbb{R}$ définie par

$$u_\lambda^\varepsilon(x) = \begin{cases} 0 & \text{si } x \in \Omega_1 \\ \varepsilon^{-1/2} u_{2,\lambda}(h^{\varepsilon-1}(x)) & \text{si } x \in \Omega_2^\varepsilon. \end{cases}$$

Alors $u_\lambda^\varepsilon \in V(\Omega^\varepsilon)$ avec

$$\|u_\lambda^\varepsilon\|_{H^1(\Omega^\varepsilon)} = \|u_{2,\lambda}\|_{H^1(\Omega_2)}.$$

De plus, $\lambda = QR(u_\lambda^\varepsilon)$ pour tout $\varepsilon > 0$ et $\lim_{\varepsilon \rightarrow 0} \langle u_{\lambda_1}^\varepsilon, u_{\lambda_2}^\varepsilon \rangle_{H^1(\Omega^\varepsilon)} = \delta_{\lambda_1, \lambda_2}$.

Ainsi, tous les $\{u_\lambda^\varepsilon\}_{\varepsilon > 0}$ construits, quelle que soit λ valeur propre du problème limite, vérifient $\lambda = \lim_{\varepsilon \rightarrow 0} QR(u_\lambda^\varepsilon)$ et $\lim_{\varepsilon \rightarrow 0} \langle u_{\lambda_1}^\varepsilon, u_{\lambda_2}^\varepsilon \rangle_{H^1(\Omega^\varepsilon)} = \delta_{\lambda_1, \lambda_2}$.

On construit alors, pour tout λ et pour tout $\varepsilon > 0$, w_λ^ε comme dans le lemme 7 et on obtient que $QR(w_\lambda^\varepsilon) \rightarrow \lambda$ lorsque $\varepsilon \rightarrow 0$. Par le théorème du Min-Max, cela montre que toute solution du problème limite est limite de solutions des problèmes sur Ω^ε .

4.2.8 Convergence de la suite entière de valeurs et fonctions propres

On a vu qu'il existe une sous-suite de $\{u^{\varepsilon,k}\}_{\varepsilon>0}$ qui converge. Il s'agit maintenant de voir que toute la suite converge.

Distinguons deux cas.

1er cas : soit $\{u^\varepsilon\}_{\varepsilon>0}$ la suite de $H^1(\Omega^\varepsilon)$ dont une sous-suite $\{u^{\varepsilon_j}\}_{j>0}$ converge vers u , fonction propre associée à une valeur propre simple λ sur Ω^ε . Soit alors $\{u^{\varepsilon_l}\}_{l>0}$ une sous-suite de $\{u^\varepsilon\}_{\varepsilon>0}$ distincte de la précédente. Alors, en faisant le même développement que pour la première sous-suite, on trouve que la limite est solution du même problème limite, de même valeur propre (par le théorème du Min-Max et la section précédente). Comme λ est simple, la limite est donc la même que pour la première suite, donc toute la suite $\{u^\varepsilon\}_{\varepsilon>0}$ converge.

2e cas : soit $\{(u_1^\varepsilon, \dots, u_m^\varepsilon)\}_{\varepsilon>0}$ une suite de $H^1(\Omega^\varepsilon)^m$ dont la sous-suite $\{(u_1^{\varepsilon_j}, \dots, u_m^{\varepsilon_j})\}_{j>0}$ converge vers (u_1, \dots, u_m) , fonctions propres associées à une valeur propre λ de multiplicité m exactement. En reprenant les arguments du premier cas, on trouve que l'espace propre engendré par $\{(u_1^\varepsilon, \dots, u_m^\varepsilon)\}_{\varepsilon>0}$ converge vers celui associé à λ .

On trouve donc dans tous les cas que la suite complète $\{u^\varepsilon\}_{\varepsilon>0}$ converge, ce qui termine la preuve du théorème sous hypothèses fortes.

4.3 Preuve du théorème sous hypothèses faibles

Nous allons à présent passer en revue les incidences sur les calculs qu'impliquent l'affaiblissement des hypothèses faites dans la section précédente afin de terminer la preuve du théorème.

1re étape : traitons le cas où la courbe γ ne définit pas un segment de droite, mais satisfait aux autres hypothèses faites lors de la preuve.

Il s'agit, dans un premier temps, de vérifier que la courbe

$$h^\varepsilon : \begin{array}{ccc} \Omega_2 & \longrightarrow & \mathbb{R}^2 \\ (x_1, x_2) & \longmapsto & \gamma(x_1) + \varepsilon x_2 n(x_1), \end{array}$$

avec $n = (-\gamma'_2, \gamma'_1)$, est un difféomorphisme sur son image si $\varepsilon > 0$ est suffisamment petit. Voyons l'injectivité de h^ε . On a

$$Dh^\varepsilon(x_1, x_2) = (\gamma' + \varepsilon x_2 n, \varepsilon n) = \begin{pmatrix} \gamma'_1 - \varepsilon x_2 \gamma''_2 & -\varepsilon \gamma'_2 \\ \gamma'_2 + \varepsilon x_2 \gamma''_1 & \varepsilon \gamma'_1 \end{pmatrix}.$$

Donc

$$\det(Dh^\varepsilon(x_1, x_2)) = \varepsilon(\gamma_1'^2 + \gamma_2'^2) + \varepsilon^2 x_2(\gamma_1''\gamma_2' - \gamma_1'\gamma_2'') = \varepsilon(1 + \varepsilon x_2 k(x_1)),$$

avec k la courbure de γ . Ainsi, si $\varepsilon < 1/|k|$ (ce qui est possible puisque k est bornée par hypothèse), on trouve que $\det(Dh^\varepsilon)$ est toujours non nul. Le théorème d'inversion locale montre l'injectivité locale de h^ε . Finalement, comme γ est injective et Ω_2 compact, on trouve que h^ε est injective.

Il faut à présent munir Ω_2 de la métrique associée à Ω_2^ε via h^ε , soit

$$g^\varepsilon(x_1, x_2) = \begin{pmatrix} (1 + \varepsilon x_2 k(x_1))^2 & 0 \\ 0 & \varepsilon^2 \end{pmatrix} \text{ au lieu de } \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon^2 \end{pmatrix}.$$

Il s'agit alors d'adapter les calculs la preuve, ce qui se fait sans mal. La remarque suivante suffit à se convaincre :

$$g^\varepsilon(x_1, x_2) - \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon^2 \end{pmatrix} \xrightarrow{\varepsilon \rightarrow 0} 0.$$

2e étape : supposons à présent que γ n'est pas nécessairement orthogonale à $\partial\Omega_1$ (mais non parallèle) et que $\partial\Omega_1$ n'est pas nécessairement rectiligne de part et d'autre de x_0 .

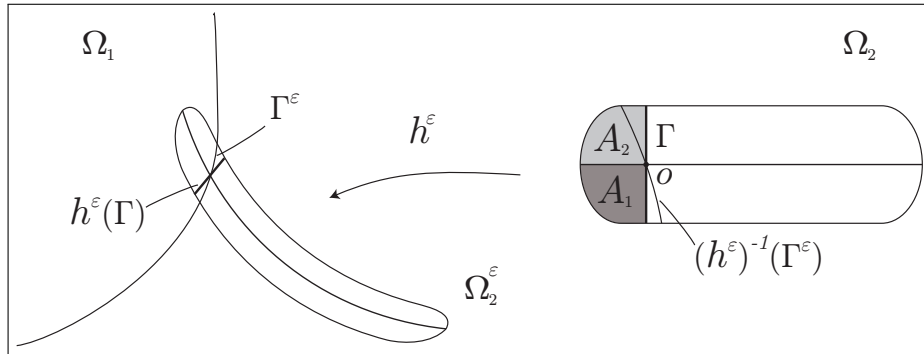
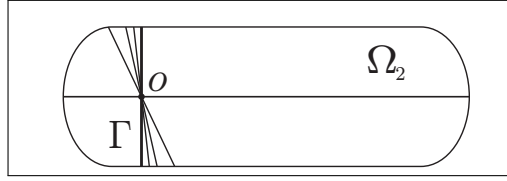


FIGURE 4.6 – Chemin non orthogonal à $\partial\Omega_1$.

Nous ne pouvons ainsi pas garantir que $h^\varepsilon(\Gamma) = \Gamma^\varepsilon$ et la preuve du lemme 5 n'est plus valable (mais le reste de la preuve reste correct) ; voyons donc les modifications à apporter afin de démontrer le lemme sous ces conditions.

Reprenons les notations adoptées lors de la preuve du lemme et notons encore les parties $A_1 = \{(x_1, x_2) \in \Omega_2^* : x_2 \leq 0\}$ et $A_2 = \{(x_1, x_2) \in \Omega_2^* : x_2 < 0\}$, de sorte que, pour ε petit, $h^\varepsilon(A_i) \subset \Omega_1$ pour $i = 1$ ou $i = 2$. Ainsi, pour ce i , on trouve que u_ε^i converge vers 0 dans $H^1(A_i)$.

FIGURE 4.7 – Itérés de $(h^\varepsilon)^{-1}(\gamma^\varepsilon)$.

Mais on a de plus, par le lemme 4, que $\frac{\partial u_2^\varepsilon}{\partial x_2}$ converge vers $\frac{\partial u_2}{\partial x_2} = 0$ dans $L^2(\Omega_2^*)$, donc u_2^ε converge vers 0 dans $H^1(\Omega_2^*)$, ce qui permet de conclure de la même manière que sous hypothèses fortes.

4.4 Métrique non euclidienne

Remarquons qu'on peut munir Ω^ε d'une métrique non euclidienne et obtenir un résultat similaire. Plus précisément, si on munit Ω^ε d'une métrique g et qu'on munit Ω_2 de celle associée à Ω_2^ε via h^ε , soit

$$g^\varepsilon = \begin{pmatrix} g_{11} & \varepsilon g_{12} \\ \varepsilon g_{12} & \varepsilon^2 g_{22} \end{pmatrix} \circ h^\varepsilon,$$

alors on voit facilement que le problème limite sur Ω_2 correspond à trouver $u_2 \in V_0(\Omega_2)$ et $\lambda \in \mathbb{R}$ tels que, pour tout $v_2 \in V_0(\Omega_2)$,

$$\int_{\Omega_2} \sqrt{\det g(x_1, 0)} g^{11}(x_1, 0) \frac{\partial u_2}{\partial x_1} \frac{\partial v_2}{\partial x_1} = \lambda \int_{\Omega_2} \sqrt{\det g(x_1, 0)} u_2 v_2,$$

avec $(g^{ij})_{i,j=1,2} = ((g_{ij})_{i,j=1,2})^{-1}$, i.e. $g^{11} = g_{22}/\det g$.

Chapitre 5

Optimisation de forme

Une question classique de géométrie spectrale est de déterminer, à volume fixé, les domaines d'une variété riemannienne qui rendent une valeur propre donnée optimale. Faber et Krahn ont par exemple montré que la première valeur propre de Dirichlet d'un domaine du plan est plus grande ou égale à celle d'un disque de même aire (voir par exemple [15]). Peu de résultats existent par contre concernant les valeurs suivantes ; aussi une détermination numérique de domaines optimaux constitue une solution intermédiaire bienvenue.

L'objectif de ce chapitre est de proposer une méthode numérique de minimisation des valeurs propres de Dirichlet pour les domaines du plan, qui soit éventuellement généralisable aux domaines de surfaces. L'approche retenue est inspirée du travail d'Oudet ([35]) : il s'agit d'une méthode de variation du bord du domaine basée sur la condition d'optimalité au premier ordre d'une *fonction-coût* J . Il est à noter qu'Oudet dit avoir abandonné rapidement la méthode de variation du bord, en dressant une liste de trois inconvénients selon lui majeurs :

- Le risque d'atteindre un minimum local de la fonction-coût.
- La difficulté de modifier la topologie du domaine.
- La difficulté d'automatiser le remaillage fréquent du domaine.

Si le problème du minimum local est bien réel, les deux autres points relevés par Oudet sont contournés ici par des procédures de redéfinition du bord permettant la modification de la topologie et par l'utilisation de l'algorithme automatisé de maillage de Delaunay présenté précédemment. L'algorithme d'optimisation de forme ainsi obtenu, conceptuellement simple, a de plus de bonnes chances d'être généralisable aux domaines de surfaces.

Les résultats numériques obtenus à l'aide des algorithmes présentés dans ce chapitre sont illustrés à la section 6.5. Nous verrons qu'ils sont validés par quelques résultats théoriques et qu'ils consolident certaines conjectures concernant les domaines optimaux, en étant de plus compétitifs par rapport aux résultats numériques actuels. En particulier, les résultats d'Oudet sont améliorés (à partir du sixième domaine optimal).

5.1 Méthode de variation du bord

Soit $k > 0$ un entier fixé; est présentée dans ce qui suit une méthode numérique de détermination du domaine de \mathbb{R}^2 dont la k^e valeur propre de Dirichlet est minimale (comparativement aux domaines de même aire).

Le point essentiel de la méthode de variation du bord est la définition d'une *dérivée* sur l'espace des formes. La dérivabilité des valeurs propres relativement à la forme du domaine ne sera toutefois pas traitée ici et nous nous contenterons de donner une preuve formelle de la formule obtenue; pour plus de détails, voir par exemple [28] ou [47]. Pour cela, on introduit sur \mathbb{R}^2 un champ de vecteurs V , appelé *champ de déformation*, et on pose

$$d(J(\Omega))(\Omega, V) = \lim_{t \rightarrow 0} \frac{J(\Omega_t) - J(\Omega)}{t},$$

avec

$$\Omega_t = \{x + tV(x) : x \in \Omega\}.$$

Chercher un extremum de la fonction-coût J revient donc à chercher à satisfaire

$$d(J(\Omega))(\Omega, V) = 0,$$

pour tout champ de vecteurs V .

Remarque : Il est facile de montrer que la dérivée $d(J(\Omega))(\Omega, V)$ ne dépend que de la composante normale de V sur le bord $\partial\Omega$ de Ω , lorsque Ω est suffisamment lisse.

Théorème 11 (Hadamard)

Soient Ω ouvert de classe \mathcal{C}^2 et $k > 0$ un entier fixé. On suppose que $\lambda_k(\Omega)$ est simple. Alors

$$d(\lambda_k(\Omega))(\Omega, V) = - \int_{\partial\Omega} \left(\frac{\partial u_k}{\partial n} \right)^2 V \cdot n \, d\sigma,$$

où u_k est la k^e fonction propre, $\|u_k\|_{L^2(\Omega)} = 1$.

Preuve formelle (tirée de [35])

Notons $u(t, x)$ la k^e fonction propre de Ω_t et $\lambda(\Omega_t)$ la valeur propre associée. Les conditions au bord de Ω_t s'écrivent

$$u(t, (Id + tV)x) = 0,$$

pour tout $x \in \partial\Omega$. En dérivant cette relation en $t = 0$, on obtient donc

$$u' + \sum_{i=1}^2 \frac{\partial u}{\partial x_i} \cdot \frac{d(Id + tV)}{dt} = u' + \sum_{i=1}^2 \frac{\partial u}{\partial x_i} V_i = 0. \quad (5.1)$$

Or,

$$\sum_{i=1}^2 \frac{\partial u}{\partial x_i} V_i = \nabla u \cdot V = \frac{\partial u}{\partial n} V \cdot n \quad (5.2)$$

puisque $u = 0$ sur $\partial\Omega$. En injectant (5.2) dans (5.1), on obtient donc

$$u' = - \sum_{i=1}^2 \frac{\partial u}{\partial x_i} V_i = - \frac{\partial u}{\partial n} V \cdot n. \quad (5.3)$$

De plus, on a par définition de u et $\lambda(\Omega_t)$

$$-\Delta u = \lambda(\Omega_t) \cdot u.$$

Ainsi, on obtient par dérivation

$$-\Delta u' = \lambda(\Omega_t) \cdot u' + \lambda'(\Omega_t) \cdot u.$$

On trouve ainsi pour $t = 0$, en injectant (5.3),

$$- \int_{\Omega} u \Delta u' = \lambda(\Omega) \int_{\Omega} u u' + \lambda'(\Omega) \int_{\Omega} u^2.$$

Donc, comme $\|u\|_{L^2(\Omega)} = 1$,

$$\lambda'(\Omega) = - \int_{\Omega} u \Delta u' - \lambda(\Omega) \int_{\Omega} u u'. \quad (5.4)$$

Or, par Green,

$$- \int_{\Omega} u \Delta u' = - \int_{\partial\Omega} \frac{\partial u'}{\partial n} u + \int_{\partial\Omega} \frac{\partial u}{\partial n} u' - \int_{\Omega} u' \Delta u = \int_{\partial\Omega} \frac{\partial u}{\partial n} u' + \lambda(\Omega) \int_{\Omega} u u'. \quad (5.5)$$

Ainsi, en injectant (5.5) dans (5.4), on obtient

$$\lambda'(\Omega) = \int_{\partial\Omega} \frac{\partial u}{\partial n} u' \stackrel{(5.3)}{=} - \int_{\partial\Omega} \left(\frac{\partial u_k}{\partial n} \right)^2 V \cdot n \, d\sigma.$$

□

Pour tenir compte du volume de Ω , comme $\lambda_k(l \cdot \Omega) = \frac{1}{l^2} \lambda_k(\Omega)$, pour tout $l > 0$, on remarque que

$$\min_{|\Omega|=1} \lambda_k(\Omega) = \min(|\Omega| \cdot \lambda_k(\Omega)).$$

On va donc considérer pour fonction-coût

$$J(\Omega) = |\Omega| \cdot \lambda_k(\Omega).$$

Proposition 5 *La dérivée de la fonction volume de Ω est donnée par*

$$d(|\Omega|)(\Omega, V) = \int_{\partial\Omega} V \cdot n \, d\sigma.$$

Preuve formelle

Considérons le domaine Ω comme paramétrisation locale de Ω_t via

$$\begin{aligned} \varphi = \varphi_t : U &\longrightarrow U_t \\ x &\longmapsto x + tV(x), \end{aligned}$$

avec $\Omega \subset U \subset \mathbb{R}^2$ et donc $\Omega_t \subset U_t \subset \mathbb{R}^N$. On trouve ainsi, en notant g_φ la métrique induite par φ sur Ω ,

$$\begin{aligned} (|\Omega_t|)' &= \left(\int_{\Omega_t} dx_t \right)' = \left(\int_{\Omega} \sqrt{\det(g_\varphi)} dx \right)' \\ &= \int_{\Omega} \left(\sqrt{\det(g_\varphi)} \right)' dx = \int_{\Omega} \frac{(\det(g_\varphi))'}{2\sqrt{\det(g_\varphi)}}. \end{aligned} \quad (5.6)$$

Calculons donc g_φ . Comme $D\varphi(x) = Id + tDV(x)$, alors

$$\begin{aligned} \left\langle \frac{\partial \varphi}{\partial x_i}, \frac{\partial \varphi}{\partial x_j} \right\rangle &= \left\langle e_i + t \frac{\partial V}{\partial x_i}(x), e_j + t \frac{\partial V}{\partial x_j}(x) \right\rangle \\ &= \delta_{ij} + t \left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i} \right) + t^2 \frac{\partial V}{\partial x_i} \frac{\partial V}{\partial x_j}. \end{aligned}$$

Ainsi, en notant $f_{ij} = \frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i}$ et $h_{ij} = \frac{\partial V}{\partial x_i} \frac{\partial V}{\partial x_j}$, on trouve

$$g_\varphi(x) = \left(\delta_{ij} + t f_{ij}(x) + t^2 h_{ij}(x) \right)_{ij}.$$

En $t = 0$, on trouve donc immédiatement $\sqrt{\det(g_\varphi)} = 1$. Ainsi, par (5.6), on trouve en $t = 0$

$$(|\Omega_t|)' = \frac{1}{2} \int_{\Omega} (\det(g_\varphi))' dx. \quad (5.7)$$

Il reste donc à calculer $(\det(g_\varphi))'$ en $t = 0$. Comme chaque coefficient de g_φ est un polynôme en t , le déterminant en sera aussi un. Et comme on veut dériver ce dernier et l'évaluer en $t = 0$, seuls les termes linéaires en t déterminent le résultat. Le calcul du déterminant montre que les termes linéaires en t de $\det(g_\varphi)$ sont $t \cdot \sum_{j=1}^2 f_{jj}$. Ainsi, on trouve

$$(|\Omega_t|)' \stackrel{(5.7)}{=} \frac{1}{2} \int_{\Omega} (\det(g_\varphi))' dx = \int_{\Omega} \operatorname{div} V dx = \int_{\partial \Omega} V \cdot n d\sigma,$$

par le théorème de la divergence. □

Ainsi, en considérant comme fonction-coût $J(\Omega) = |\Omega| \cdot \lambda_k(\Omega)$, la dérivée associée est

$$\begin{aligned} d(J(\Omega))(\Omega, V) &= d(|\Omega|)(\Omega, V) \cdot \lambda_k(\Omega) + |\Omega| \cdot d(\lambda_k(\Omega))(\Omega, V) \\ &= \left(\int_{\partial\Omega} V \cdot n \, d\sigma \right) \cdot \lambda_k + |\Omega| \cdot \left(- \int_{\partial\Omega} \left(\frac{\partial u_k}{\partial n} \right)^2 V \cdot n \, d\sigma \right) \\ &= \int_{\partial\Omega} \left(\lambda_k - |\Omega| \left(\frac{\partial u_k}{\partial n} \right)^2 \right) \cdot V \cdot n \, d\sigma . \end{aligned}$$

La fonction-coût J et sa dérivée étant définies, il est à présent possible de donner l'algorithme d'optimisation de forme suivant :

Soit Ω un domaine initial. Faire en boucle :

1. Discrétisation du bord : on sélectionne des points P_i du bord $\partial\Omega$ de Ω , appelés *points de contrôle*, idéalement tous équidistants de leurs voisins.
2. Calcul des dérivées de forme : pour chaque point P_i sélectionné, on calcule la *dérivée de forme* d_i définie par

$$d_i = d(J(\Omega))(\Omega, V_i) = \frac{d}{dt} J(\Omega_t^i) = \lim_{t \rightarrow 0} \frac{J(\Omega_t^i) - J(\Omega)}{t} ,$$

avec $J(\Omega) = |\Omega| \cdot \lambda_k(\Omega)$, $\Omega_t^i = \{x + tV_i(x) : x \in \Omega\}$ et $V_i(x) = \varphi_i(x) \cdot n(P_i)$, où $n(P_i)$ est le vecteur normal extérieur de Ω en P_i et φ_i est la fonction de $\mathcal{C}(\partial\Omega)$, affine sur chaque segment $P_j P_k$ de $\partial\Omega$ et définie par la relation $\varphi_i(P_j) = \delta_{ij}$.

3. Critère de sortie : si le vecteur $d = (d_i)$ est petit, l'algorithme stoppe.
4. Variation du bord : les P_i sont translatés de $\alpha \cdot d_i \cdot n(P_i)$, où $\alpha > 0$ est un facteur bien choisi.
5. Evaluation de la fonction-coût : après avoir contrôlé que les P_i forment toujours un polygone ne s'auto-intersectant pas (ou après avoir utilisé un algorithme traitant une telle situation), on évalue $J(\Omega)$ et on ajuste éventuellement le facteur α .

5.2 Dérivée normale d'une fonction propre de Dirichlet

Dans le calcul des dérivées de forme intervient, selon le théorème d'Hadamard, la dérivée normale d'une fonction propre de Dirichlet, i.e. d'une solution u du problème

$$\begin{cases} \text{Trouver } u \in H_0^1(\Omega) \text{ et } \lambda \in \mathbb{R} \text{ tels que, pour tout } v \in H_0^1(\Omega) : \\ \int_{\Omega} \langle \nabla u, \nabla v \rangle = \lambda \int_{\Omega} uv . \end{cases}$$

Or, si on considère $v \in H^1(\Omega)$, on remarque que le couple (u, λ) satisfait la relation

$$\int_{\Omega} \langle \nabla u, \nabla v \rangle - \int_{\partial\Omega} \frac{\partial u}{\partial n} v = \lambda \int_{\Omega} uv .$$

Ainsi, une fois le couple (u, λ) connu, la dérivée normale de u peut être déterminée comme unique solution du problème variationnel

$$(P_n) \quad \begin{cases} \text{Trouver } \omega \in L^2(\partial\Omega) \text{ tel que, pour tout } v \in H^1(\Omega) : \\ a(\omega, v) = b(v) , \end{cases}$$

avec

$$\begin{aligned} a(\omega, v) &= \int_{\partial\Omega} \omega v , \\ b(v) &= \int_{\Omega} \langle \nabla u, \nabla v \rangle - \lambda \int_{\Omega} uv . \end{aligned}$$

5.2.1 Calcul matriciel

Soit τ une triangulation de Ω , domaine que l'on suppose polygonal. Notons N_0 le nombre de nœuds de la triangulation, N le nombre de ceux n'appartenant pas à $\partial\Omega$ et $\{P_i\}_{i=1}^{N_0}$ les nœuds de la triangulation, avec $P_i \in \partial\Omega$ si et seulement si $i > N$. Considérons encore la base étendue $\{\varphi_i\}_{i=1}^{N_0}$ des *fonctions formes* : il s'agit des fonctions de $\mathcal{C}(\bar{\Omega})$, affines sur chaque élément de la triangulation et définies par la relation $\varphi_i(P_j) = \delta_{ij}$. Notons de même sans distinction φ_i la trace de ces fonctions sur $\partial\Omega$ (pour $i > N$).

La résolution numérique du problème aux valeurs propres préalablement effectuée a permis d'écrire la fonction propre u dans la base $\{\varphi_i\}_{i=1}^{N_0}$, disons

$$u = \sum_{i=1}^N u_i \varphi_i$$

(le fait que les coefficients de φ_i avec $i > N$ soient nuls provient des conditions au bord). Il s'agit donc de chercher ω de la forme

$$\omega = \sum_{i=N+1}^{N_0} \omega_i \varphi_i$$

(en voyant les φ_i comme des fonctions sur $\partial\Omega$).

Ainsi, le problème (P_n) est équivalent à trouver $\{\omega_j\}_{j=N+1}^{N_0}$ tel que

$$\sum_{j=N+1}^{N_0} \omega_j a(\varphi_j, \varphi_i) = b(\varphi_i),$$

pour tout $i \in \{1, \dots, N_0\}$.

Autrement dit, en notant $a_{ij} = a(\varphi_j, \varphi_i)$, $b_i = b(\varphi_i)$, $\eta = (\omega_j)_{j=N+1}^{N_0}$, $A = (a_{ij})_{i,j=N+1}^{N_0}$ et $b = (b_i)_{i=N+1}^{N_0}$, le problème (P_n) est équivalent à chercher $\eta \in \mathbb{R}^{N_0-N}$ vérifiant l'équation matricielle

$$A\eta = b. \quad (5.8)$$

Pour résoudre numériquement cette équation, on remarque que A est symétrique définie positive. Il s'agit alors d'effectuer une décomposition de Cholesky de A , disons $A = N^t N$, puis de résoudre successivement les équations $N^t \xi = b$ et $N\eta = \xi$ (N et N^t étant triangulaires).

5.2.2 Construction des matrices

La construction des éléments matriciels A et b de l'équation (5.8) suit une logique similaire à celle des matrices de rigidité et de masse du problème aux valeurs propres initial (section 3.5), c'est-à-dire qu'elles se font progressivement triangle par triangle, respectivement segment par segment, via des matrices locales. Quelques remarques s'imposent toutefois.

Concernant la matrice A , ses coefficients sont définis par

$$a_{ij} = \int_{\partial\Omega} \varphi_i \varphi_j.$$

pour $i, j > N$. Ils sont non nuls si et seulement si P_i et P_j sont soit confondus, soit des nœuds successifs du bord $\partial\Omega$ de Ω . Un rapide calcul montre que, dans le premier cas, la contribution d'un segment $P_i P_k$ de $\partial\Omega$ au coefficient a_{ii} vaut le tiers de la longueur du segment. Dans le deuxième cas, a_{ij} vaut le sixième de la longueur du segment $P_i P_j$ de $\partial\Omega$.

Quant au vecteur b , on remarque tout d'abord que ses coefficients peuvent s'écrire

$$b_j = \sum_{k=1}^N u_k \left(\int_{\Omega} \langle \nabla \varphi_k, \nabla \varphi_j \rangle - \lambda \int_{\Omega} \varphi_k \varphi_j \right),$$

pour $j > N$. On y retrouve des éléments similaires à ceux des matrices de masse et de rigidité du problème initial, à la différence près que les nœuds P_j associés appartiennent au bord du domaine Ω , alors que les nœuds P_k appartiennent à l'intérieur de Ω . Ainsi, la contribution d'un triangle de la triangulation au vecteur b est non nulle seulement si le triangle contient un sommet sur le bord de Ω .

5.3 Changement de topologie des domaines

Au fil des itérations de l'algorithme d'optimisation de forme et des déformations successives de la courbe polygonale définissant le bord du domaine, il peut arriver que celle-ci s'auto-intersecte, ne définissant plus alors de domaine. Il s'agit alors de traiter les différentes situations d'auto-intersections possibles de la courbe, par modification de cette dernière, de manière à ce qu'elle définisse à nouveau un domaine. Le corollaire à cette redéfinition est bien souvent une modification de la topologie du domaine et de son bord.

Pour illustrer les différentes situations possibles, considérons le domaine de la figure 5.1 et supposons que son bord soit déformé selon une des variantes ci-dessous.

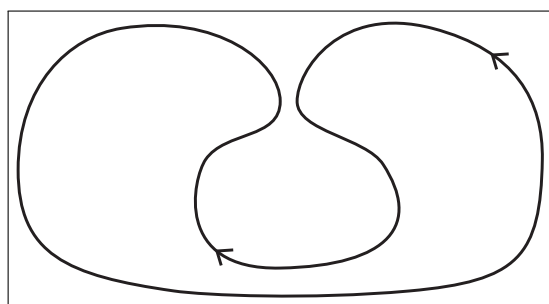


FIGURE 5.1 – Domaine initial.

- a) Contact ponctuel du bord du domaine (voir figure 5.2, à gauche).
- b) Contact non ponctuel du bord du domaine (voir figure 5.2, à droite).
- c) Recoupement du domaine (voir figure 5.3, à gauche).
- d) Séparation du domaine (voir figure 5.3, à droite).

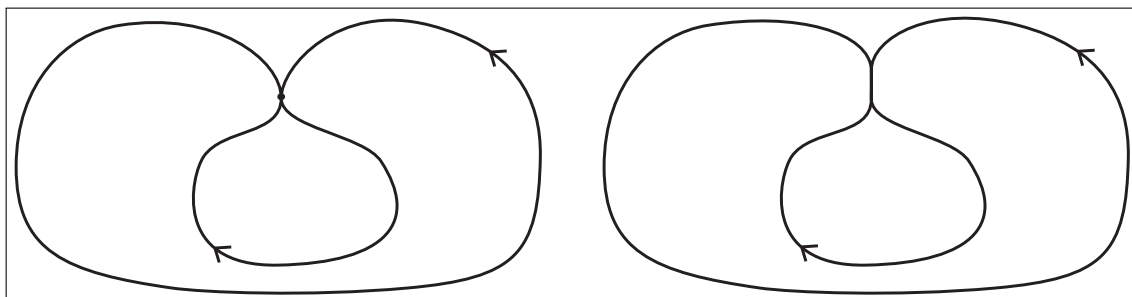


FIGURE 5.2 – Contacts du bord du domaine.

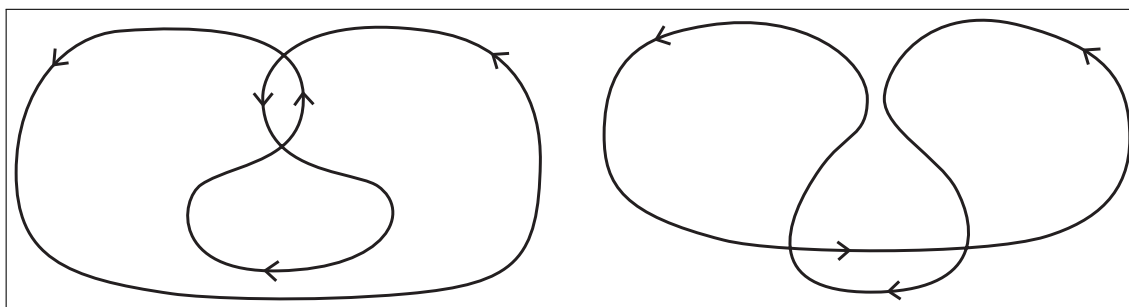


FIGURE 5.3 – Recouplement et séparation du domaine.

L'algorithme de traitement de ces situations, consistant essentiellement en la suppression des parties superflues de la courbe, est le suivant :

1. Suppression des contacts non ponctuels de la courbe (les allés-retours).
2. Création de nœuds doubles aux intersections de la courbe.
3. Ré-aiguillage de la courbe au voisinage des nœuds doubles.
4. Suppression des boucles superflues (ces dernières étant déterminées par la valeur de leur indice par rapport à un point milieu d'un de leurs segments).
5. Écartement des nœuds doubles restants (élargissement du contact ponctuel de deux parties du domaine).

Reprenons les illustrations précédentes de situations de mauvaise définition du domaine ; les résultats du traitement de ces courbes sont illustrés ci-dessous.

- a) Les contacts ponctuels de bord de domaine sont traités par écartement des points de contact.

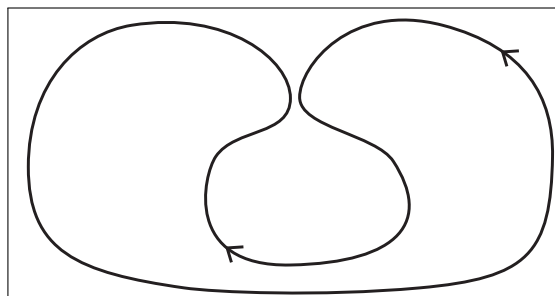


FIGURE 5.4 – Traitement des contacts ponctuels.

- b) Les contacts non ponctuels de bord de domaine sont traités par suppression des allés-retours.

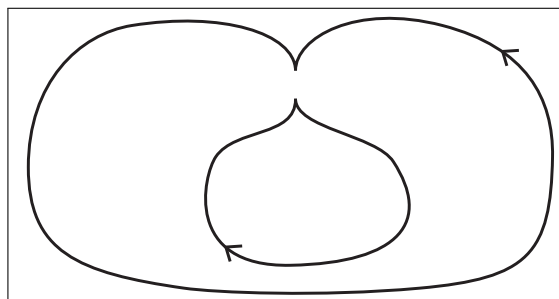


FIGURE 5.5 – Traitement des contacts non ponctuels.

- c) Les recouvrements de domaine sont traités par ré-aiguillage de la courbe au voisinage des points d'intersections et écartement des points de contact (voir figure 5.6, à gauche), puis par suppression des boucles superflues (voir figure 5.6, à droite).

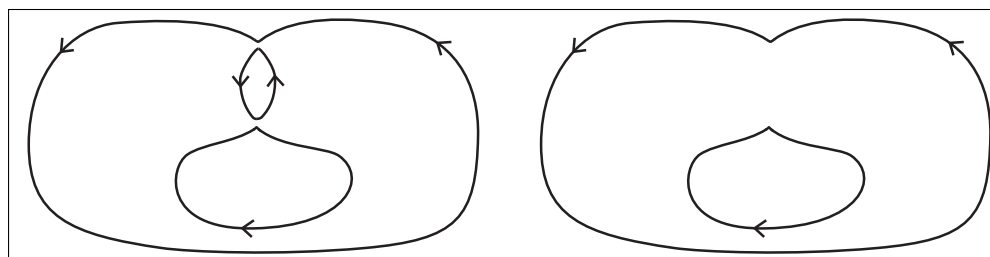


FIGURE 5.6 – Traitement des recouvrements.

- d) Les séparations de domaine sont traités de la même manière que les recouvrements de domaine, c'est-à-dire par ré-aiguillage de la courbe au voisinage des points d'intersections et écartement des points de contact (voir figure 5.7, à gauche), puis par suppression des boucles superflues (voir figure 5.7, à droite).

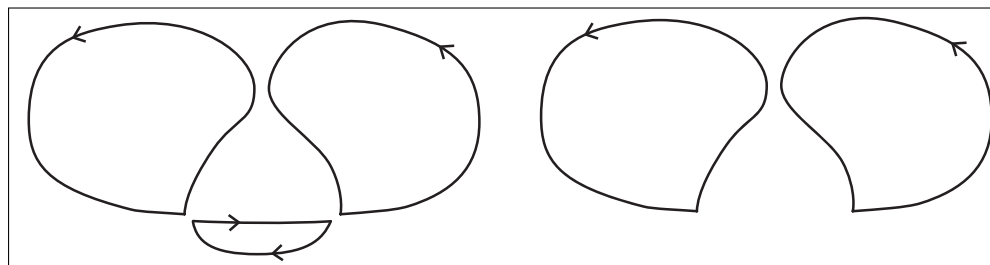


FIGURE 5.7 – Traitement des séparations.

Chapitre 6

Exemples et résultats numériques

Dans les pages qui suivent se trouvent des exemples de valeurs et fonctions propres calculées à l'aide des programmes développés pour ce travail et basés sur les algorithmes décrits précédemment, notamment la méthode des éléments finis pour des domaines de surfaces, les algorithmes de maillage par macro-éléments et par Delaunay ainsi que l'algorithme de Lanczos pour la résolution de problèmes matriciels aux valeurs propres. Les illustrations des domaines et fonctions propres ont été réalisés à l'aide du logiciel de visualisation *Paraview* (©Sandia Corporation, Kitware Inc.).

Les programmes développés permettront de vérifier numériquement quelques résultats théoriques bien connus. Nous présenterons de plus quelques modélisations de domaines à anses fines, particulièrement délicates à effectuer mais qui illustrent pourtant avec une surprenante efficacité (étant données les largeurs modestes des anses) le théorème 10 du chapitre 4 à propos des domaines à anses planes et en indiquent une apparente généralisation aux domaines à anses cylindriques. En point d'orgue, quelques domaines obtenus grâce à l'algorithme d'optimisation de forme seront illustrés et discutés.

A noter que le mass-lumping a généralement été utilisé lors de l'intégration numérique, accélérant ainsi les calculs; en contrepartie, les valeurs propres obtenues ne sont pas systématiquement supérieures aux valeurs propres exactes. De plus, les erreurs engendrées par l'algorithme de Lanczos sont négligeables proportionnellement à celles provenant de la discrétisation du problème (i.e. de la méthode des éléments finis) et de l'intégration numérique (en cas de métrique non triviale). Les différences observables entre les résultats numériques obtenus et les éventuelles solutions théoriques dépendent donc essentiellement de la résolution des triangulations considérées pour les différents domaines. La taille des matrices obtenues, mesure de la résolution de la triangulation, sera donc indiquée pour chaque calcul numérique. La largeur de bande de la matrice de rigidité de chaque système, en lien avec la numérotation des noeuds de la triangulation utilisée, sera également précisée, de même que le nombre d'itérations de l'algorithme de résolution du problème matriciel aux valeurs propres associé, mesure avec la résolution de la triangulation du temps de calcul nécessaire à la résolution du problème. A remarquer toutefois que le nombre d'itérations

dépend beaucoup du nombre de solutions recherchées et qu'il est parfois préférable de demander davantage de solutions que souhaitées afin d'accélérer l'algorithme. La librairie *ARPACK* est par exemple bien plus performante lorsqu'il lui est demandé de déterminer toutes les multiplicités d'une même valeur propre, ou du moins tout un ensemble de valeurs propres proches, plutôt qu'une partie seulement (voir [32] ou [34]).

Concernant l'échelle de couleurs employée dans ce qui suit pour représenter les fonctions propres, le vert correspond à une valeur nulle, alors que le bleu et le rouge correspondent à des valeurs opposées, de valeurs absolues plus grandes à mesure que les couleurs sont plus vives.



FIGURE 6.1 – Echelle de couleurs employée.

6.1 Premiers exemples

6.1.1 Carré, cylindre et tores

Le carré constitue le premier exemple à traiter. Il possède en effet plusieurs avantages, dont celui d'être très facilement modélisable (via un seul macro-élément) tout en servant de modèle à des domaines progressivement plus complexes, tels le cylindre (vu comme quotient du carré), le tore plat (idem) et le tore de \mathbb{R}^3 (idem, mais avec une métrique non triviale). De plus, la résolution manuelle du problème aux valeurs propres du carré, du cylindre et du tore plat est aisée et permet de comparer les solutions théoriques avec les résultats numériques obtenus à partir de maillages plus ou moins fins.

Carré

Pour déterminer manuellement le spectre de Dirichlet du carré $[0, 1]^2$, employons la méthode de *séparation des variables*, i.e. supposons que les fonctions propres sont de la forme

$$u(x) = u_1(x_1) \cdot u_2(x_2).$$

Ainsi, l'équation $-\Delta u = \lambda u$ devient

$$-u_1'' u_2 - u_1 u_2'' = \lambda u_1 u_2.$$

En divisant par $u_1 u_2$ (les cas où u_1 et u_2 sont nuls étant inintéressants), on trouve la relation

$$-\frac{u_1''}{u_1} = \lambda + \frac{u_2''}{u_2}.$$

Or, le terme de gauche est indépendant de x_2 et celui de droite est indépendant de x_1 ; ils sont donc égaux à une constante. On peut de plus affirmer que cette constante est positive, de par la nature des conditions au bord.

Ainsi, le problème est équivalent à chercher u_1 , u_2 , λ et c solutions respectives des problèmes

$$(P_1) \quad \begin{cases} \text{Trouver } u_1 : [0, 1] \rightarrow \mathbb{R} \text{ et } c \in \mathbb{R} \text{ tels que :} \\ -u_1'' = c^2 u_1 \quad \text{dans }]0, 1[\\ u_1 = 0 \quad \text{en } 0 \text{ et } 1, \end{cases}$$

$$(P_2) \quad \begin{cases} \text{Trouver } u_2 : [0, 1] \rightarrow \mathbb{R} \text{ et } c, \lambda \in \mathbb{R} \text{ tels que :} \\ -u_2'' = (\lambda - c^2)u_2 \quad \text{dans }]0, 1[\\ u_2 = 0 \quad \text{en } 0 \text{ et } 1. \end{cases}$$

La résolution du problème (P_1) donne

$$u_{1,n}(x_1) = c_1 \sin(n\pi x_1),$$

avec $c = n\pi$, $n \in \mathbb{N} \setminus \{0\}$ et $c_1 \in \mathbb{R}$ une constante.

La résolution du problème (P_2) donne quant à elle

$$u_{2,m}(x_2) = c_2 \sin(m\pi x_2),$$

avec $\sqrt{\lambda - c^2} = m\pi$, $m \in \mathbb{N} \setminus \{0\}$ et $c_2 \in \mathbb{R}$ une constante.

Au total, on trouve les valeurs et fonctions propres

$$\begin{aligned} \lambda_{n,m} &= (n^2 + m^2)\pi^2, \\ u_{n,m}(x) &= C \sin(n\pi x_1) \sin(m\pi x_2), \end{aligned}$$

avec $n, m \in \mathbb{N} \setminus \{0\}$ et $C \in \mathbb{R}$ une constante. En particulier, les premières valeurs propres sont

1. $\lambda_{1,1} = 2\pi^2$
2. $\lambda_{1,2} = \lambda_{2,1} = 5\pi^2$ (valeur double)
3. $\lambda_{2,2} = 8\pi^2$
4. $\lambda_{1,3} = \lambda_{3,1} = 10\pi^2$ (valeur double)

La méthode de séparation des variable fonctionne de manière semblable avec des conditions de Neumann sur $x_1 = 0$, $x_1 = 1$, $x_2 = 0$ et/ou $x_2 = 1$ (ainsi que sur des domaines rectangulaires). On trouve par exemple les valeurs et fonctions propres de Neumann

$$\begin{aligned} \lambda_{n,m} &= (n^2 + m^2)\pi^2, \\ u_{n,m}(x) &= C \cos(n\pi x_1) \cos(m\pi x_2), \end{aligned}$$

avec $n, m \in \mathbb{N}$ et $C \in \mathbb{R}$ une constante, d'où en particulier les premières valeurs propres

1. $\lambda_{0,0} = 0$
2. $\lambda_{0,1} = \lambda_{1,0} = \pi^2$ (valeur double)
3. $\lambda_{1,1} = 2\pi^2$
4. $\lambda_{0,2} = \lambda_{2,0} = 4\pi^2$ (valeur double)
5. $\lambda_{1,2} = \lambda_{2,1} = 5\pi^2$ (valeur double)
6. $\lambda_{2,2} = 8\pi^2$

Concernant la résolution numérique du problème, le maillage du carré ne nécessite qu'un seul macro-élément. Ci-dessous sont présentées quelques valeurs propres de Dirichlet et de Neumann obtenues numériquement pour différentes résolutions de triangulations ainsi que les solutions exactes. Les fonctions propres associées (obtenues avec la plus haute de ces résolutions) sont également représentées.

Carré (Dirichlet)	Valeurs théoriques	1 macro-él. 10 × 10	1 macro-él. 20 × 20	1 macro-él. 40 × 40
Taille des matrices		81	361	1521
Largeur de bande		11	21	41
Nombre d'itérations		133	432	926
λ_1	19.74	19.58	19.70	19.73
λ_2	49.35	47.99	49.00	49.26
λ_3	49.35	47.99	49.00	49.26
λ_4	78.96	76.39	78.31	78.79
λ_5	98.70	92.23	97.04	98.28
λ_6	98.70	92.23	97.04	98.28

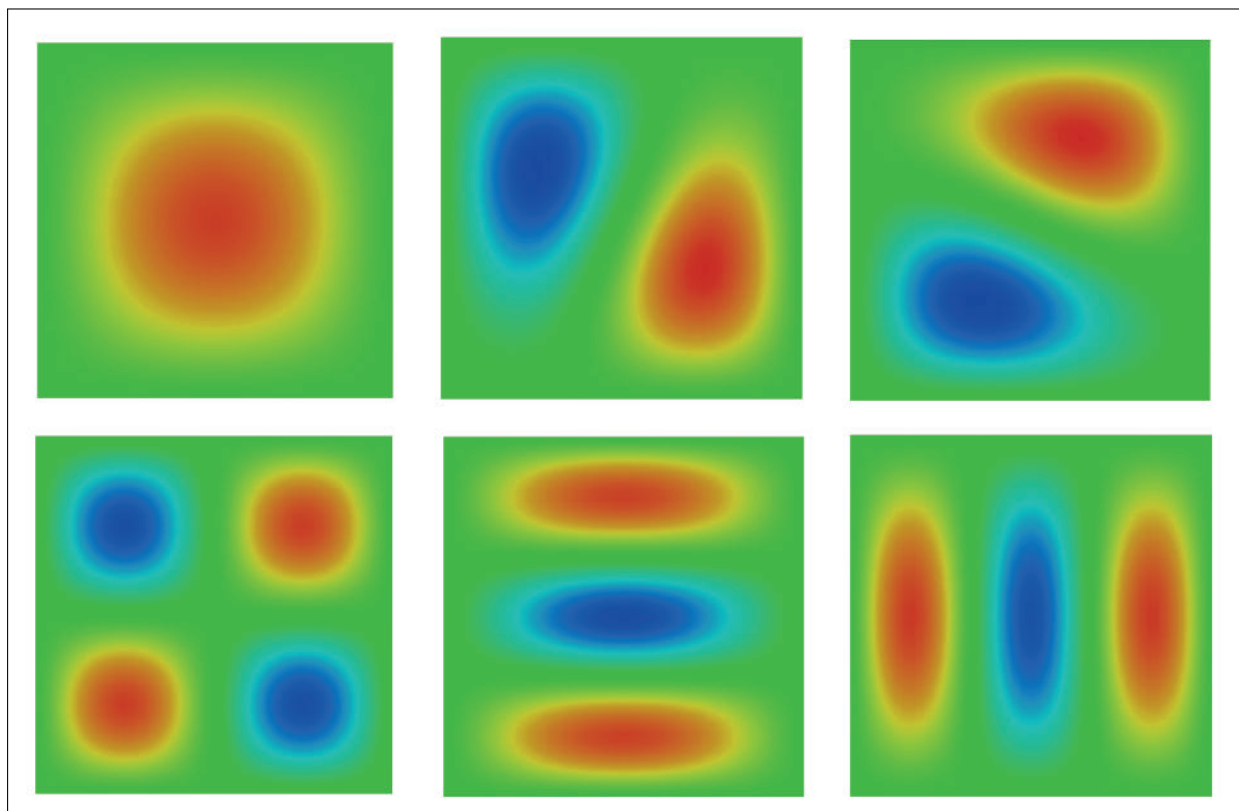


FIGURE 6.2 – Premières fonctions propres du carré, conditions de Dirichlet.

Carré (Neumann)	Valeurs théoriques	1 macro-él. 10 × 10	1 macro-él. 20 × 20	1 macro-él. 40 × 40
Taille des matrices		121	441	1681
Largeur de bande		13	23	43
Nombre d'itérations		180	541	2829
λ_0	0	0.00	0.00	0.00
λ_1	9.87	9.72	9.83	9.86
λ_2	9.87	9.85	9.87	9.87
λ_3	19.74	19.57	19.70	19.73
λ_4	39.48	38.18	39.15	39.40
λ_5	39.48	38.20	39.15	39.40
λ_6	49.35	47.32	48.84	49.22

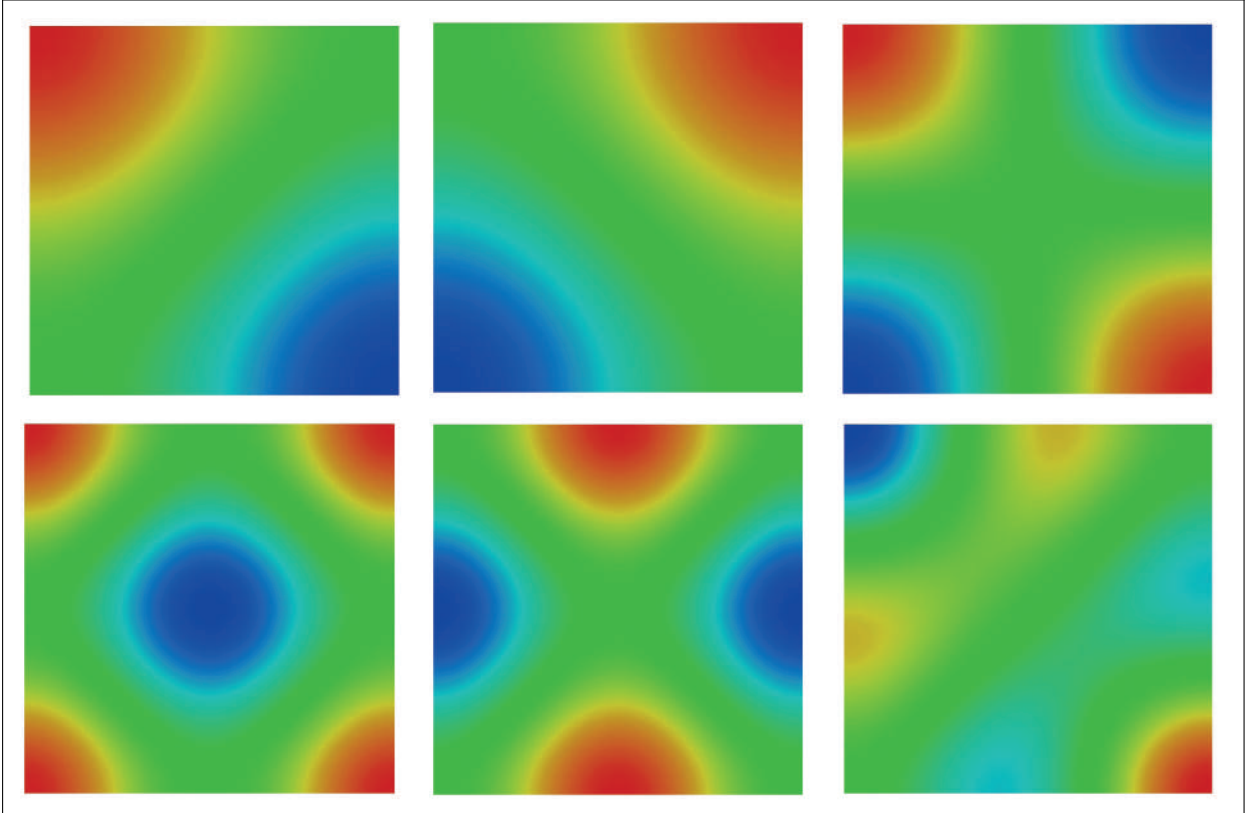


FIGURE 6.3 – Premières fonctions propres non triviales du carré, conditions de Neuman.

Cylindre

Plutôt que de considérer le cylindre $\mathbb{S}^1 \times [0, 1]$, nous voyons ici le cylindre comme $[0, 1]^2 / \mathbb{Z}$, i.e. le quotient du carré $[0, 1]^2$ par la relation d'équivalence \sim définie par

$$x \sim y \text{ si et seulement si } x_1 - y_1 \in \mathbb{Z} \text{ et } x_2 - y_2 = 0.$$

Autrement dit, on recolle le carré le long de son bord vertical. Ainsi, le problème aux valeurs propres s'écrit, par exemple si on considère les conditions au bord de Dirichlet,

$$\left\{ \begin{array}{l} \text{Trouver } u : [0, 1]^2 \rightarrow \mathbb{R} \text{ et } \lambda \in \mathbb{R} \text{ tels que :} \\ \quad -\Delta u = \lambda u \quad \text{dans }]0, 1[^2 \\ \quad \frac{\partial^k u}{\partial x_1^k}(0, x_2) = \frac{\partial^k u}{\partial x_1^k}(1, x_2) \quad \text{pour } x_2 \in [0, 1] \text{ et } k \in \mathbb{N} \\ \quad u = 0 \quad \text{sur } [0, 1] \times \{0, 1\}. \end{array} \right.$$

Pour déterminer manuellement le spectre du cylindre, il s'agit alors de reprendre la stratégie utilisée pour le carré, à savoir la méthode de séparation des variables. On trouve alors, de manière semblable au carré, que les solutions $u(x) = u_1(x_1) \cdot u_2(x_2)$ vérifient

$$u_{1,n}(x_1) = c_{1,1} \cos(2n\pi x_1) + c_{1,2} \sin(2n\pi x_1),$$

avec $c = 2n\pi$, $n \in \mathbb{N}$ et $c_{1,1}, c_{1,2} \in \mathbb{R}$ des constantes, ainsi que

$$u_{2,m}(x_2) = c_2 \sin(m\pi x_2),$$

avec $\sqrt{\lambda - c^2} = m\pi$, $m \in \mathbb{N} \setminus \{0\}$ et $c_2 \in \mathbb{R}$ une constante.

Au total, on trouve les valeurs et fonctions propres

$$\begin{aligned} \lambda_{n,m} &= (4n^2 + m^2)\pi^2, \\ u_{n,m}(x) &= (C_1 \cos(2n\pi x_1) + C_2 \sin(2n\pi x_1)) \sin(m\pi x_2), \end{aligned}$$

avec $n \in \mathbb{N}$, $m \in \mathbb{N} \setminus \{0\}$ et $C_1, C_2 \in \mathbb{R}$ des constantes. En particulier, les premières valeurs propres sont

- | | |
|---|--|
| 1. $\lambda_{0,1} = \pi^2$ | 6. $\lambda_{1,3} = 13\pi^2$ (valeur double) |
| 2. $\lambda_{0,2} = 4\pi^2$ | 7. $\lambda_{0,4} = 16\pi^2$ |
| 3. $\lambda_{1,1} = 5\pi^2$ (valeur double) | 8. $\lambda_{2,1} = 17\pi^2$ (valeur double) |
| 4. $\lambda_{1,2} = 8\pi^2$ (valeur double) | 9. $\lambda_{1,4} = \lambda_{2,2} = 20\pi^2$ (valeur quadruple) |
| 5. $\lambda_{0,3} = 9\pi^2$ | 10. $\lambda_{0,5} = \lambda_{2,3} = 25\pi^2$ (valeur quadruple) |

La résolution numérique du problème est basée sur celle du carré et ne nécessite en surplus que l'identification des faces verticales de l'unique macro-élément employé pour le maillage du domaine. Une renumérotation des nœuds des différentes triangulations s'avère ici nécessaire du fait des recolllements effectués. Ci-dessous sont présentées quelques valeurs propres de Dirichlet obtenues numériquement pour différentes résolutions de triangulations ainsi que les solutions exactes. Les fonctions propres associées (obtenues avec la plus haute de ces résolutions) sont également représentées.

Cylindre (Dirichlet)	Valeurs théoriques	1 macro-él. 10 × 10	1 macro-él. 20 × 20	1 macro-él. 40 × 40
Taille des matrices		90	380	1560
Largeur de bande		17	32	62
Nombre d'itérations		117	355	1150
λ_1	9.87	9.79	9.85	9.86
λ_2	39.48	38.20	39.15	39.40
λ_3	49.35	47.99	49.00	49.26
λ_4	49.35	47.99	49.00	49.26
λ_5	78.96	76.39	78.31	78.79
λ_6	78.96	76.39	78.31	78.79

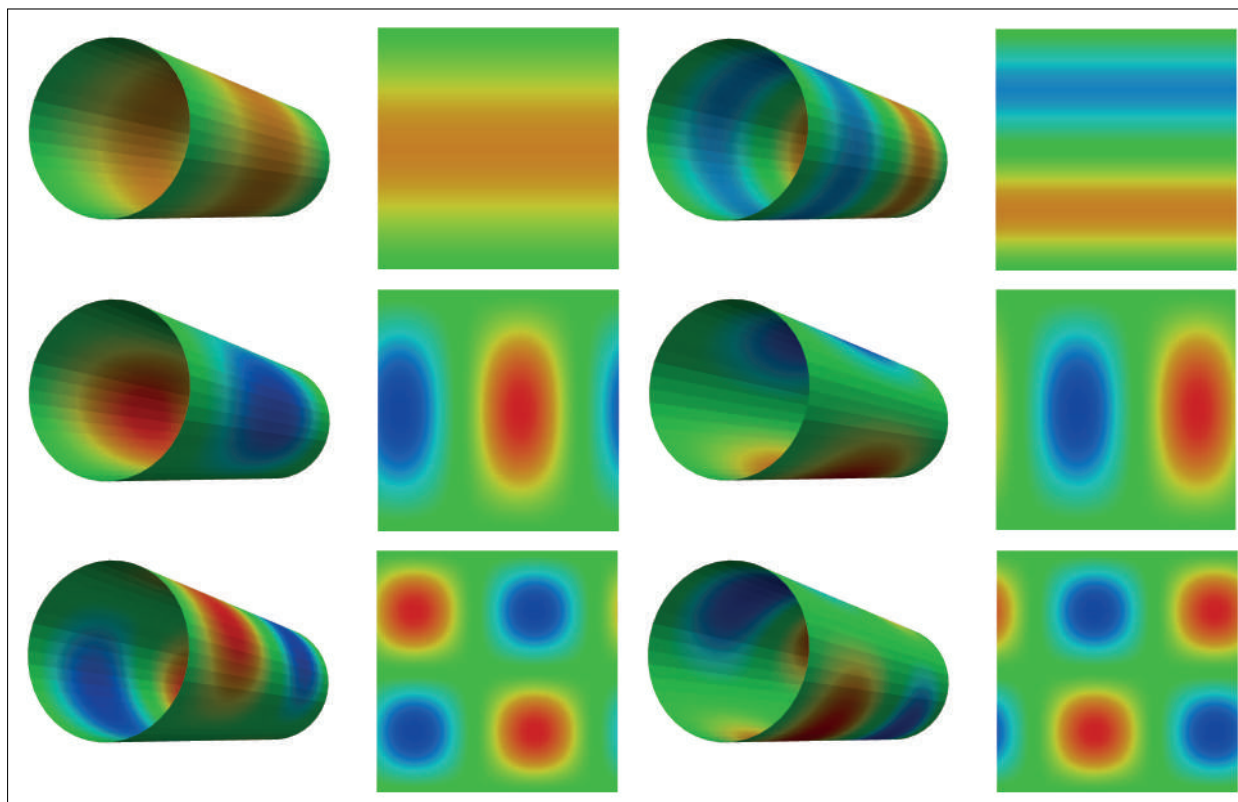


FIGURE 6.4 – Premières fonctions propres du cylindre, conditions de Dirichlet.

Tore plat

De manière semblable au cylindre et plutôt que de considérer le tore plat $\mathbb{S}^1 \times \mathbb{S}^1$, nous le voyons ici comme $[0, 1]^2 / \mathbb{Z}^2$, i.e. le quotient du carré $[0, 1]^2$ par la relation d'équivalence \sim définie par

$$x \sim y \text{ si et seulement si } x_i - y_i \in \mathbb{Z}.$$

Autrement dit, on recolle le carré le long de son bord vertical, puis le long de son bord horizontal. Ainsi, le problème aux valeurs propres s'écrit

$$\left\{ \begin{array}{l} \text{Trouver } u : [0, 1]^2 \rightarrow \mathbb{R} \text{ et } \lambda \in \mathbb{R} \text{ tels que :} \\ -\Delta u = \lambda u \quad \text{dans }]0, 1[^2 \\ \frac{\partial^k u}{\partial x_1^k}(0, x_2) = \frac{\partial^k u}{\partial x_1^k}(1, x_2) \quad \text{pour } x_2 \in [0, 1] \text{ et } k \in \mathbb{N} \\ \frac{\partial^k u}{\partial x_2^k}(x_1, 0) = \frac{\partial^k u}{\partial x_2^k}(x_1, 1) \quad \text{pour } x_1 \in [0, 1] \text{ et } k \in \mathbb{N}. \end{array} \right.$$

On reprend alors à nouveau la méthode de séparation des variables afin de déterminer manuellement le spectre du tore et on trouve que les solutions $u(x) = u_1(x_1) \cdot u_2(x_2)$

vérifient

$$u_{1,n}(x_1) = c_{1,1} \cos(2n\pi x_1) + c_{1,2} \sin(2n\pi x_1),$$

avec $c = 2n\pi$, $n \in \mathbb{N}$ et $c_{1,1}, c_{1,2} \in \mathbb{R}$ des constantes, ainsi que

$$u_{2,m}(x_2) = c_{2,1} \cos(2m\pi x_2) + c_{2,2} \sin(2m\pi x_2),$$

avec $\sqrt{\lambda - c^2} = 2m\pi$, $m \in \mathbb{N}$ et $c_{2,1}, c_{2,2} \in \mathbb{R}$ des constantes.

Au total, on trouve les valeurs et fonctions propres

$$\begin{aligned} \lambda_{n,m} &= 4(n^2 + m^2)\pi^2, \\ u_{n,m}(x) &= (C_1 \cos(2n\pi x_1) + C_2 \sin(2n\pi x_1)) \cdot (C_3 \cos(2m\pi x_2) + C_4 \sin(2m\pi x_2)), \end{aligned}$$

avec $n, m \in \mathbb{N}$ et $C_1, C_2, C_3, C_4 \in \mathbb{R}$ des constantes. En particulier, les premières valeurs propres sont

1. $\lambda_{0,0} = 0$
2. $\lambda_{0,1} = \lambda_{1,0} = 4\pi^2$ (valeur quadruple)
3. $\lambda_{1,1} = 8\pi^2$ (valeur quadruple)
4. $\lambda_{0,2} = \lambda_{2,0} = 16\pi^2$ (valeur quadruple)

La résolution numérique du problème est une fois de plus basée sur celle du carré et ne nécessite en surplus que l'identification des faces opposées de l'unique macro-élément employé pour le maillage du domaine. Une renumérotation des nœuds des différentes triangulations s'avère ici nécessaire du fait des recollements effectués. Il s'agit également de prendre garde à calculer un nombre relativement élevé de valeurs propres à la fois afin de garantir la validité des premières valeurs, l'algorithme risquant sinon d'en manquer quelques unes dans les cas de grande multiplicité (4 en l'occurrence). Ci-dessous sont présentées quelques valeurs propres obtenues numériquement pour différentes résolutions de triangulations ainsi que les solutions exactes. Les fonctions propres associées (obtenues avec la plus haute de ces résolutions) sont également représentées.

Tore plat	Valeurs théoriques	1 macro-él. 10 × 10	1 macro-él. 20 × 20	1 macro-él. 40 × 40
Taille des matrices		100	400	1600
Largeur de bande		29	59	119
Nombre d'itérations		110	260	544
λ_0	0	0.00	0.00	0.00
λ_1	39.48	38.20	39.15	39.40
λ_2	39.48	38.20	39.15	39.40
λ_3	39.48	38.20	39.15	39.40
λ_4	39.48	38.20	39.15	39.40
λ_5	78.96	76.39	78.31	78.79
λ_6	78.96	76.39	78.31	78.79

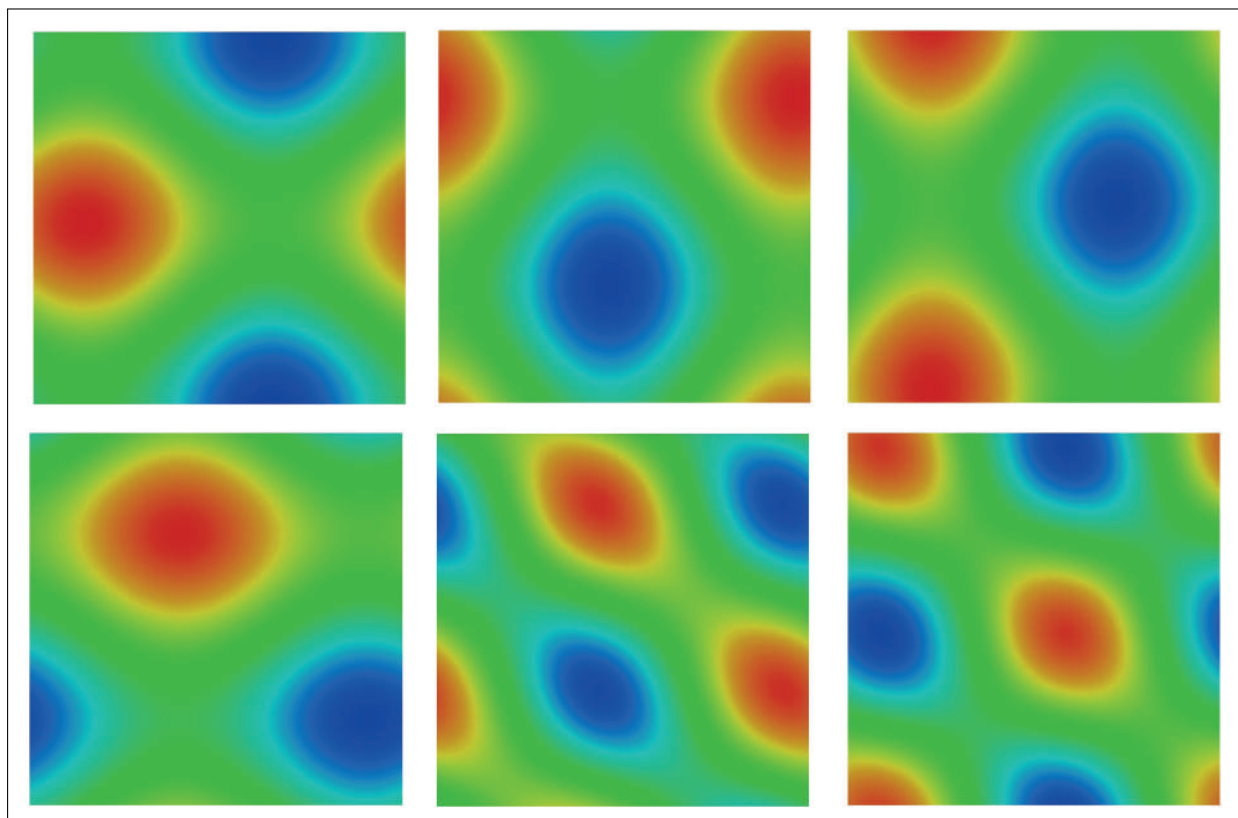


FIGURE 6.5 – Premières fonctions propres non triviales du tore plat.

Tore de \mathbb{R}^3

Le maillage du tore de \mathbb{R}^3 est identique à celui du tore plat. Il s'agit toutefois, pour la résolution du problème à proprement parler, de tenir compte de la métrique associée au tore de \mathbb{R}^3 .

En détails, on considère la fonction

$$\begin{aligned} \varphi : \quad \mathbb{R}^2 &\longrightarrow \mathbb{R}^3 \\ (x_1, x_2) &\longmapsto \begin{pmatrix} (r_1 - r_2 \cos(2\pi x_2)) \cos(2\pi x_1) \\ (r_1 - r_2 \cos(2\pi x_2)) \sin(2\pi x_1) \\ r_2 \sin(2\pi x_2) \end{pmatrix} \end{aligned}$$

L'image de φ est le tore de \mathbb{R}^3 de grand rayon r_1 et de petit rayon r_2 . De plus, φ passe au quotient de \mathbb{R}^2 par \mathbb{Z}^2 .

La métrique à considérer pour la résolution numérique est donc la première forme fon-

damentale associée à φ , c'est-à-dire

$$g(x_1, x_2) = 4\pi^2 \begin{pmatrix} (r_1 - r_2 \cos(2\pi x_2))^2 & 0 \\ 0 & r_2^2 \end{pmatrix}.$$

Ci-dessous sont présentées quelques valeurs propres obtenues numériquement pour diverses résolutions de triangulations du tore de \mathbb{R}^3 de grand rayon $r_1 = 1$ et de petit rayon $r_2 = 0.5$, ainsi que les fonctions propres associées (obtenues avec la plus haute de ces résolutions), représentées sur le tore lui-même ainsi qu'en paramétrisation.

Tore de \mathbb{R}^3 $r_1 = 1, r_2 = 0.5$	1 macro-él. 8×16	1 macro-él. 16×32	1 macro-él. 32×64
Taille des matrices	128	512	2048
Largeur de bande	25	49	97
Nombre d'itérations	430	731	5228
λ_0	0.00	0.00	0.00
λ_1	0.99	1.00	1.00
λ_2	0.99	1.00	1.00
λ_3	3.06	3.15	3.17
λ_4	3.06	3.15	3.17
λ_5	3.73	3.86	3.90
λ_6	4.28	4.44	4.48
λ_7	4.87	5.01	5.04
λ_8	4.87	5.01	5.04

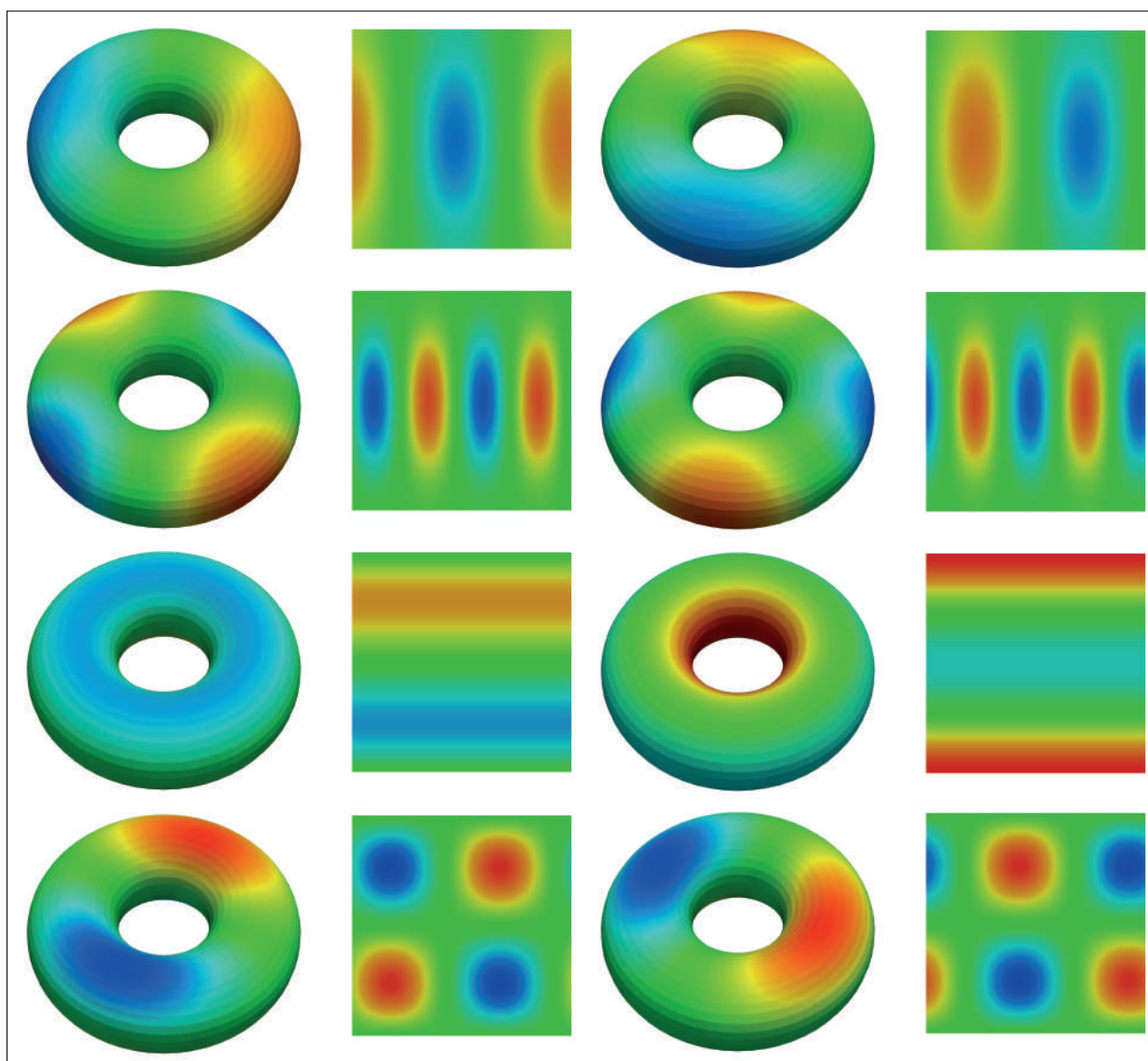


FIGURE 6.6 – Premières fonctions propres non triviales du tore.

6.1.2 Disque et sphères

Le disque débute une deuxième série d'exemples d'utilisation des programmes développés. Si son maillage est plus complexe que celui du carré (son bord n'étant pas polygonal), les résultats numériques obtenus peuvent toutefois être vérifiés puisque les valeurs propres du disque sont connues (de manière approchée), en lien avec les fonctions de Bessel (voir par exemple [54]). De plus, le disque sert de modèle aux sphères plate (vue comme quotient de la réunion disjointe de deux disques) et ronde (idem, mais avec une métrique non triviale).

Disque

La détermination manuelle du spectre du disque unité $B(0,1)$ de \mathbb{R}^2 emploie la méthode de séparation des variables. Toutefois, contrairement au cas du carré (et de ses quotients), il s'agit de considérer les fonctions propres et l'équation associée sous formes polaires.

Ecrivons donc

$$u(r, \theta) = f(r) \cdot h(\theta)$$

et rappelons la forme polaire du Laplacien :

$$\Delta u = u_{r,r} + \frac{1}{r}u_r + \frac{1}{r^2}u_{\theta,\theta},$$

$$\text{avec } u_r = \frac{\partial u}{\partial r}, \quad u_{r,r} = \frac{\partial^2 u}{\partial r^2}, \quad u_\theta = \frac{\partial u}{\partial \theta} \quad \text{et} \quad u_{\theta,\theta} = \frac{\partial^2 u}{\partial \theta^2}.$$

Autrement dit, on peut écrire $u_r = f'h$, $u_{r,r} = f''h$, $u_\theta = fh'$ et $u_{\theta,\theta} = fh''$, l'équation $-\Delta u = \lambda u$ devenant alors

$$\frac{r^2 \left(f'' + \frac{1}{r}f' + \lambda f \right)}{f} = -\frac{h''}{h}$$

(en évacuant les cas où f et h sont nuls).

On remarque alors que le terme de gauche est indépendant de θ , alors que celui de droite est indépendant de r ; ils sont donc égaux à une constante positive (de par la nature des conditions au *bord*).

Ainsi, le terme de droite conduit au problème

$$\left\{ \begin{array}{l} \text{Trouver } h : [0, 2\pi] \rightarrow \mathbb{R} \text{ et } c \in \mathbb{R} \text{ tels que :} \\ -h'' = c^2 h \quad \text{dans }]0, 2\pi[\\ h^{(k)}(0) = h^{(k)}(2\pi) \quad \text{pour } k \in \mathbb{N}, \end{array} \right.$$

d'où

$$h(\theta) = c_1 \cos(n\theta) + c_2 \sin(n\theta),$$

avec $c = n \in \mathbb{N}$ et $c_1, c_2 \in \mathbb{R}$ des constantes. Le terme de gauche conduit alors à l'équation

$$f'' + \frac{1}{r}f' + \left(\lambda - \frac{n^2}{r^2} \right) f = 0.$$

En notant $x = kr$, $y_n(x) = f(kr)$ et $k^2 = \lambda$, on retrouve ainsi les *fonctions de Bessel*, à savoir les solutions de l'équation

$$y_n'' + \frac{1}{x}y_n' + \left(1 - \frac{n^2}{x^2} \right) y_n = 0.$$

De plus, si on considère le spectre de Dirichlet, la condition au bord $f(1) = 0$ se traduit par $y_n(k) = 0$, i.e. $y_n(\sqrt{\lambda}) = 0$. Autrement dit, les valeurs propres de Dirichlet sont les carrés des zéros des fonctions de Bessel. En particulier, les premières valeurs propres sont

- | | |
|---|---|
| 1. $\lambda_1 \cong 5.78$ (pour $n = 0$) | 6. $\lambda_6 \cong 49.22$ (pour $n = 1$; val. double) |
| 2. $\lambda_2 \cong 14.68$ (pour $n = 1$; val. double) | 7. $\lambda_7 \cong 57.58$ (pour $n = 4$; val. double) |
| 3. $\lambda_3 \cong 26.37$ (pour $n = 2$; val. double) | 8. $\lambda_8 \cong 70.86$ (pour $n = 2$; val. double) |
| 4. $\lambda_4 \cong 30.47$ (pour $n = 0$) | 9. $\lambda_9 \cong 74.89$ (pour $n = 0$) |
| 5. $\lambda_5 \cong 40.71$ (pour $n = 3$; val. double) | 10. $\lambda_{10} \cong 76.94$ (pour $n = 5$; val. double) |

De même, si on considère le spectre de Neumann, la condition au bord $f'(1) = 0$ se traduit par $y'_n(k) = 0$, i.e. $y'_n(\sqrt{\lambda}) = 0$. Autrement dit, les valeurs propres de Neumann sont les carrés des zéros des dérivées des fonctions de Bessel. En particulier, les premières valeurs propres sont

- | | |
|--|---|
| 1. $\lambda_1 = 0$ (pour $n = 0$) | 5. $\lambda_5 \cong 17.65$ (pour $n = 3$; val. double) |
| 2. $\lambda_2 \cong 3.39$ (pour $n = 1$; val. double) | 6. $\lambda_6 \cong 28.28$ (pour $n = 4$; val. double) |
| 3. $\lambda_3 \cong 9.33$ (pour $n = 2$; val. double) | 7. $\lambda_7 \cong 28.42$ (pour $n = 1$; val. double) |
| 4. $\lambda_4 \cong 14.68$ (pour $n = 0$) | 8. $\lambda_8 \cong 41.16$ (pour $n = 5$; val. double) |

Concernant la résolution numérique du problème, bien qu'un maillage de Delaunay à partir d'une liste des nœuds du bord du domaine soit envisageable, la méthode employée ici consiste en une décomposition en un macro-élément central carré et quatre macro-éléments ayant chacun une face courbe. Dans les deux cas, une renumérotation des nœuds de la triangulation s'avère nécessaire. Ci-dessous sont présentées quelques valeurs propres exactes et numériques, ces dernières provenant de différentes résolutions de triangulations obtenues par maillage par macro-éléments. Les fonctions propres associées (obtenues avec la plus haute de ces résolutions) sont également représentées.

Disque (Dirichlet)	Valeurs théoriques	5 macro-él. 6 × 6 / 3 × 6	5 macro-él. 12 × 12 / 6 × 6	5 macro-él. 24 × 24 / 12 × 24
Taille des matrices		97	409	1681
Largeur de bande		16	31	61
Nombre d'itérations		148	531	2362
λ_1	5.78	5.78	5.78	5.78
λ_2	14.68	14.53	14.64	14.67
λ_3	14.68	14.62	14.67	14.68
λ_4	26.37	25.35	26.12	26.31
λ_5	26.37	26.49	26.41	26.39
λ_6	30.47	29.29	30.17	30.40

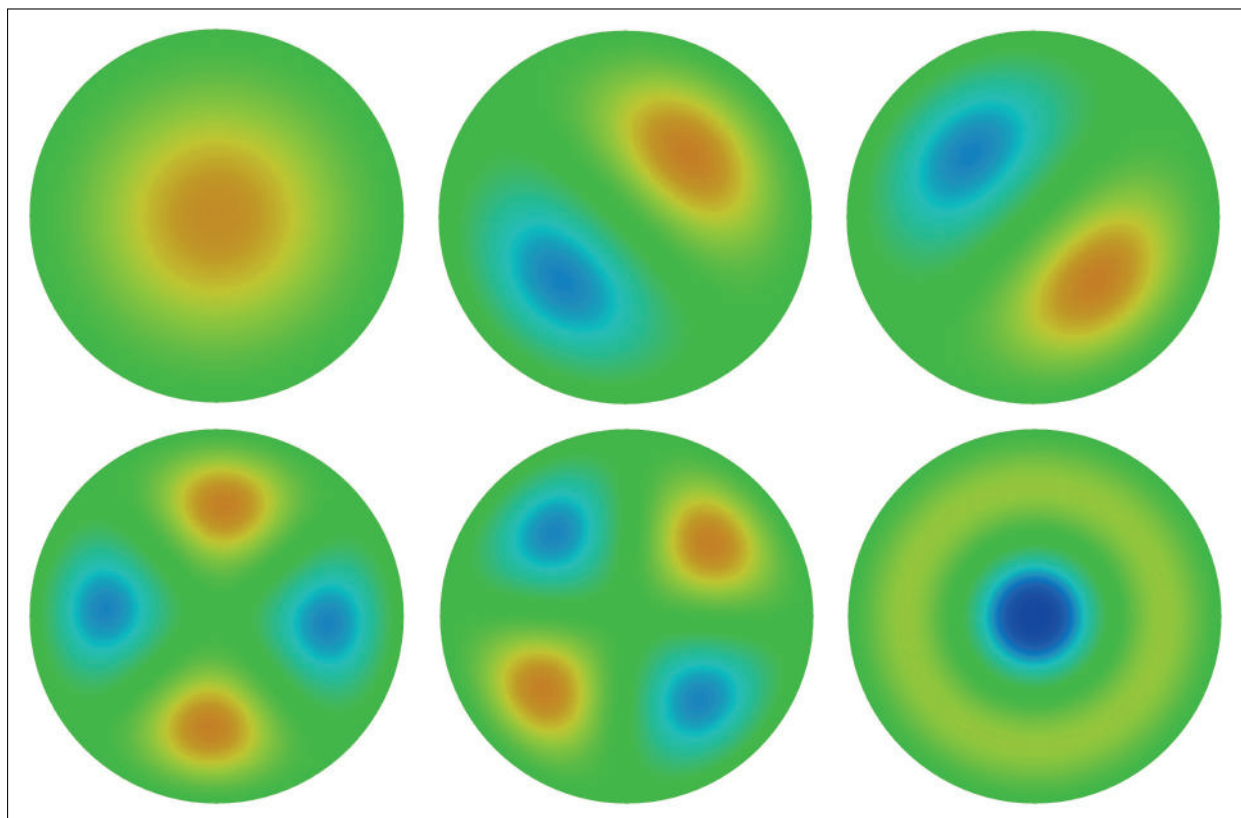


FIGURE 6.7 – Premières fonctions propres du disque, conditions de Dirichlet.

Disque (Neumann)	Valeurs théoriques	5 macro-él. 6 × 6 / 3 × 6	5 macro-él. 12 × 12 / 6 × 6	5 macro-él. 24 × 24 / 12 × 24
Taille des matrices		121	457	1777
Largeur de bande		18	33	63
Nombre d'itérations		233	933	3996
λ_0	0	0.00	0.00	0.00
λ_1	3.39	3.38	3.39	3.39
λ_2	3.39	3.40	3.39	3.39
λ_3	9.33	9.06	9.26	9.31
λ_4	9.33	9.16	9.29	9.32
λ_5	14.68	14.40	14.61	14.66
λ_6	17.65	16.62	17.40	17.59

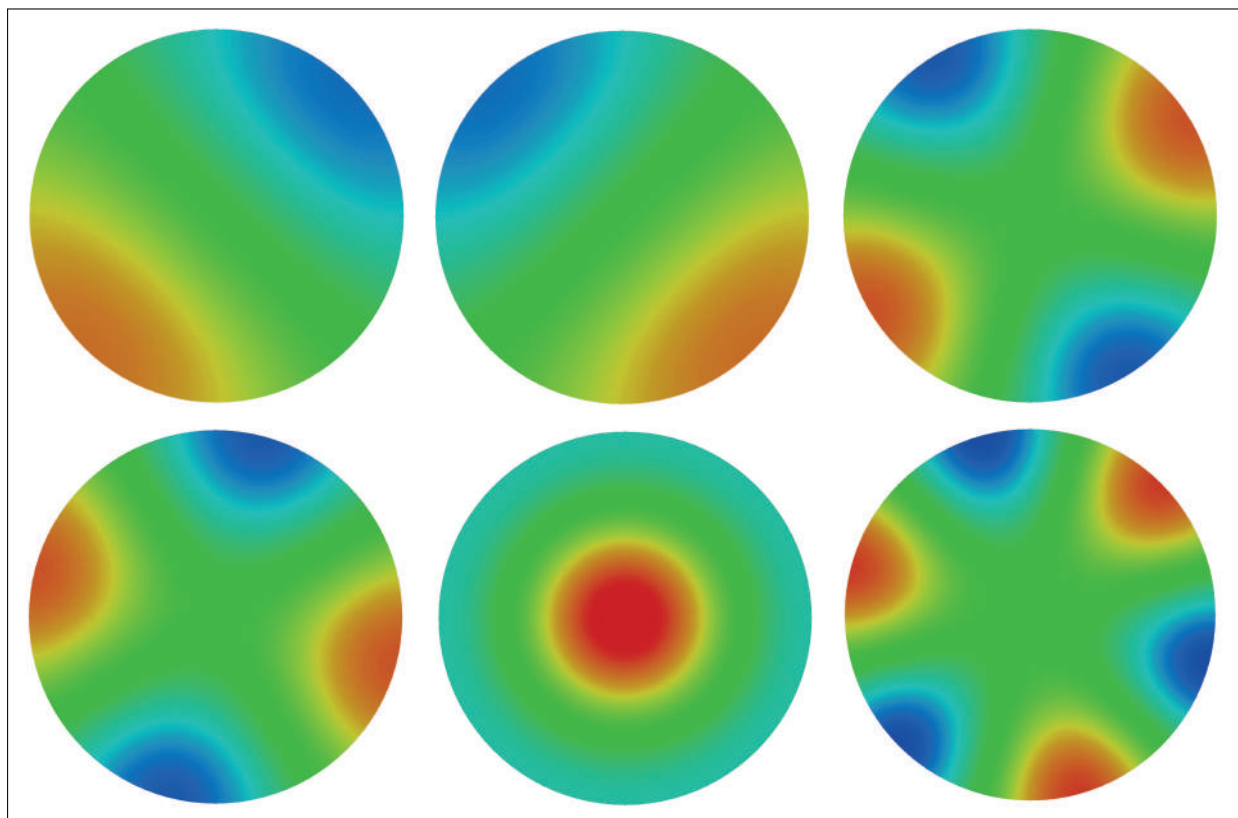


FIGURE 6.8 – Premières fonctions propres non triviales du disque, conditions de Neumann.

Sphère plate

La surface considérée ici et nommée *sphère plate* ne peut pas être plongée dans \mathbb{R}^3 . Il s'agit du quotient de la réunion disjointe de deux disques unité D_1, D_2 par la relation d'équivalence \sim définie par

$$x \sim y \text{ si et seulement si } x = y \text{ ou } x \in D_i, y \in D_{3-i}, (x_1, x_2) = (y_1, -y_2).$$

Autrement dit, on colle le long de leurs bords deux disques superposés (en prenant garde à l'orientation).

La détermination manuelle du spectre de la sphère plate, employant la propriété de symétrie donnée par la proposition 6 et les valeurs propres du disque, sera détaillée à la section 6.2.2.

La résolution numérique du problème est basée sur celle du disque, à appliquer à deux copies disjointes de ce domaine. Il s'agit en surplus d'identifier les bords des disques. Ci-dessous sont présentées quelques valeurs propres obtenues numériquement pour différentes résolutions de triangulations ainsi que les solutions exactes, ainsi que les fonctions propres

associées (obtenues avec la plus haute de ces résolutions).

Sphère plate	Valeurs théoriques	10 macro-él. 6 × 6 / 3 × 6	10 macro-él. 12 × 12 / 6 × 6	10 macro-él. 24 × 24 / 12 × 24
Taille des matrices		218	866	3458
Largeur de bande		32	62	122
Nombre d'itérations		247	1386	5140
λ_0	0	0.00	0.00	0.00
λ_1	3.39	3.39	3.39	3.39
λ_2	3.39	3.39	3.39	3.39
λ_3	5.78	5.78	5.78	5.78
λ_4	9.33	9.09	9.27	9.31
λ_5	9.33	9.14	9.28	9.32
λ_6	14.68	14.40	14.61	14.66
λ_7	14.68	14.57	14.66	14.68
λ_8	14.68	14.57	14.66	14.68

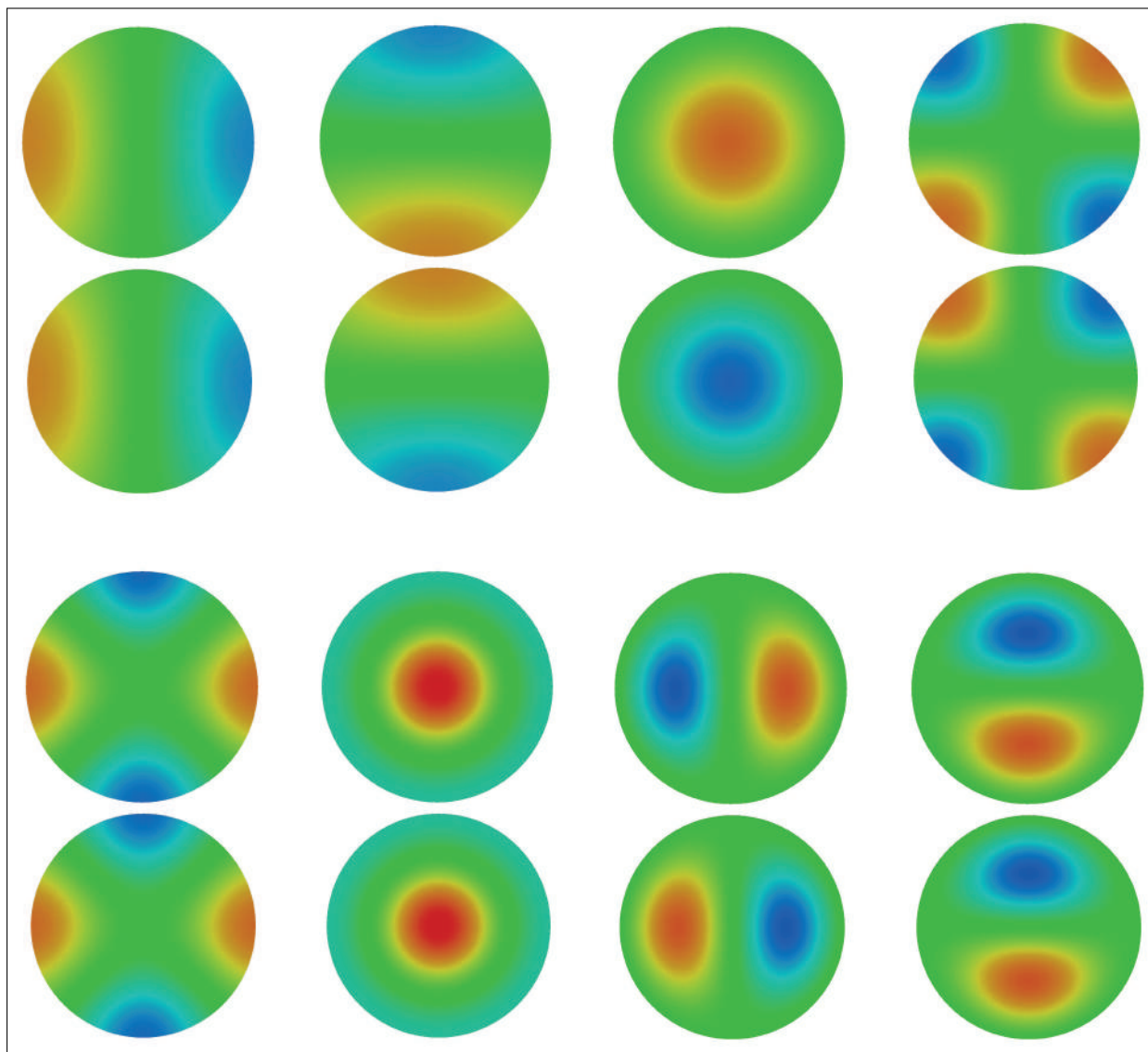


FIGURE 6.9 – Premières fonctions propres non triviales de la sphère plate.

Sphère ronde

La détermination manuelle du spectre de la sphère ronde \mathbb{S}^2 , mettant à profit les symétries de cette surface, fait appel à des notions d'algèbre comme les groupes de Lie ; voir par exemple [36]. On peut montrer que les valeurs propres sont

$$\lambda_n = n(n + 1),$$

avec $n \in \mathbb{N}$, de multiplicité $2n + 1$. De plus, les fonctions propres associées à λ_n , appelées *harmoniques sphériques*, sont les restrictions des polynômes harmoniques homogènes de degré n .

Le maillage de la sphère ronde est identique à celui de la sphère plate. Il s'agit toutefois, pour la résolution du problème à proprement parler, de tenir compte de la métrique associée à \mathbb{S}^2 .

En détails, on considère pour atlas de la sphère les *projections stéréographiques*, c'est-à-dire les paramétrisations locales φ_1, φ_2 définies par

$$\begin{aligned} \varphi_i : \quad \mathbb{R}^2 &\longrightarrow \mathbb{R}^3 \\ (x_1, x_2) &\longmapsto \frac{1}{1 + x_1^2 + x_2^2} (2x_1, 2x_2, (-1)^i(x_1^2 + x_2^2 - 1)) . \end{aligned}$$

Comme les images $\varphi_1(B(0, 1))$, respectivement $\varphi_2(B(0, 1))$, sont les hémisphères nord, respectivement sud, et que $\varphi_1(u, v) = \varphi_2(u, -v)$ sur \mathbb{S}^1 , alors les premières formes fondamentales associées à φ_1 et φ_2 constituent la métrique à considérer pour la résolution numérique ; il s'agit de

$$g(x_1, x_2) = \frac{4}{(1 + x_1^2 + x_2^2)^2} Id .$$

Lors de la résolution numérique du problème à proprement parler, il s'agit de prendre garde à calculer un nombre relativement élevé de valeurs propres à la fois afin de garantir la validité des premières valeurs, l'algorithme risquant sinon d'en manquer quelques unes dans les cas de grande multiplicité. Ci-dessous sont présentées quelques valeurs propres obtenues numériquement pour différentes résolutions de triangulations ainsi que les solutions exactes. Les fonctions propres associées (obtenues avec la plus haute de ces résolutions) sont également représentées sur la sphère elle-même ainsi qu'en paramétrisation.

Sphère ronde	Valeurs théoriques	10 macro-él. 6 × 6 / 3 × 6	10 macro-él. 12 × 12 / 6 × 12	10 macro-él. 24 × 24 / 12 × 24
Taille des matrices		218	866	3458
Largeur de bande		32	62	122
Nombre d'itérations		331	1209	4522
λ_0	0	0.00	0.00	0.00
λ_1	2	1.99	2.00	2.00
λ_2	2	2.01	2.00	2.00
λ_3	2	2.01	2.00	2.00
λ_4	6	5.88	5.97	5.99
λ_5	6	5.89	5.97	5.99
λ_6	6	5.91	5.98	5.99
λ_7	6	5.91	5.98	5.99
λ_8	6	5.92	5.98	5.99

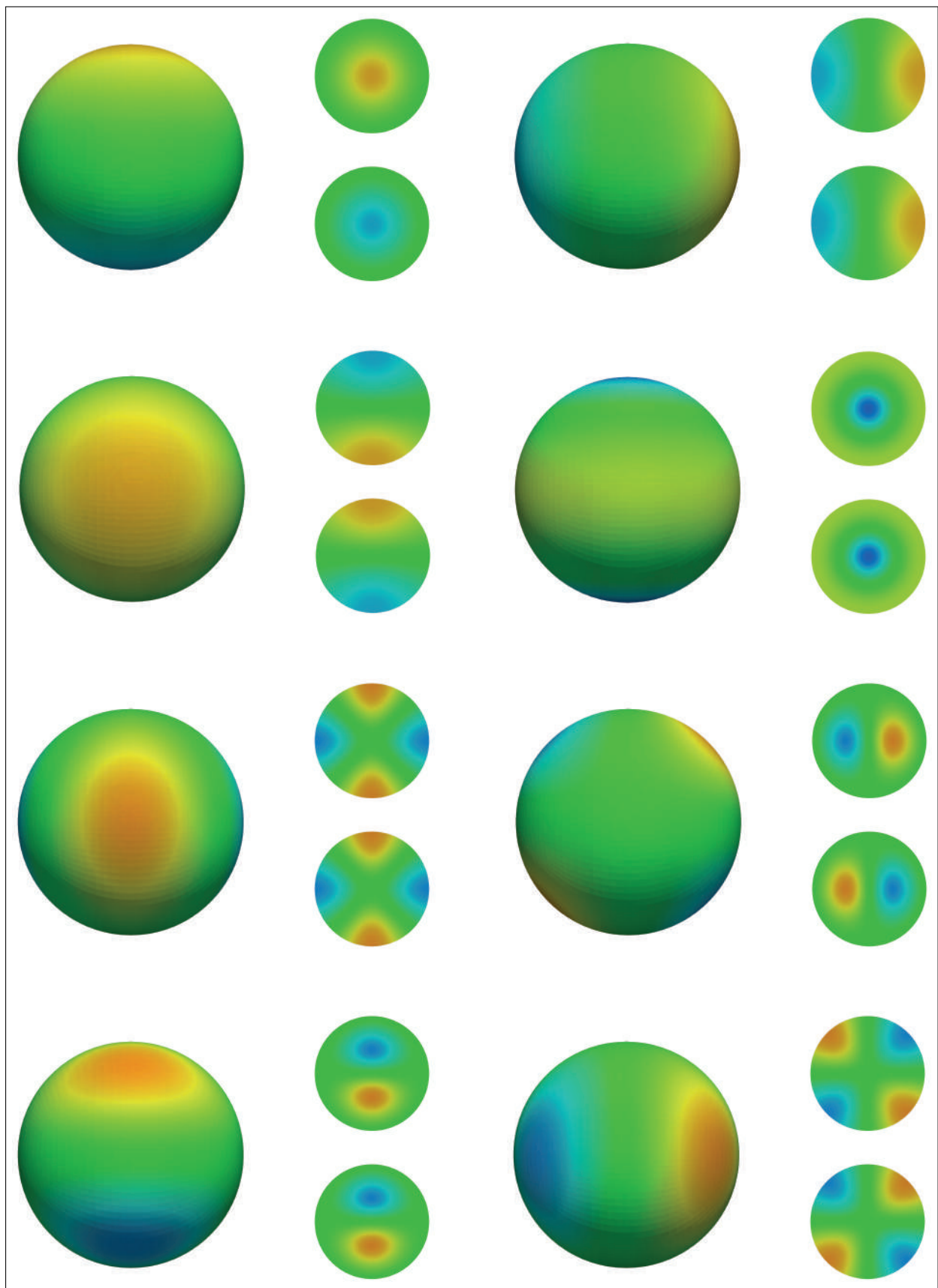


FIGURE 6.10 – Premières fonctions propres non triviales de la sphère.

6.2 Vérification d'une propriété de symétrie

Dans cette section, nous employerons les programmes développés à la vérification numérique de la propriété de symétrie donnée par la proposition ci-dessous.

Proposition 6 *Soient Ω_1, Ω_2 des domaines d'une surface S tels que $S = \bar{\Omega}_1 \cup \bar{\Omega}_2$ et $\partial\Omega_1 = \partial\Omega_2 = \Gamma$. Supposons encore qu'il existe une isométrie $\psi : S \rightarrow S$ telle que $\psi|_\Gamma = Id$ et $\psi(\Omega_i) = \Omega_{3-i}$. En particulier, les Ω_i sont isométriques, donc isospectraux. Alors les spectres vérifient*

$$\sigma(S) = \sigma_N(\Omega_i) \cup \sigma_D(\Omega_i).$$

Preuve

Voyons déjà que $\sigma(S) \subset \sigma_N(\Omega_i) \cup \sigma_D(\Omega_i)$. Pour cela, considérons f une fonction propre de S . Soient alors les fonctions $f_N = f + f \circ \psi$ et $f_D = f - f \circ \psi$: lorsqu'elles ne sont pas nulles (et elles ne le sont pas toutes les deux), il s'agit encore de fonctions propres de S , toujours de même valeur propre. De plus, leurs restrictions $f_{N,i} = f_N|_{\Omega_i}$ et $f_{D,i} = f_D|_{\Omega_i}$ vérifient $\frac{\partial}{\partial n} f_{N,i} = 0$ et $f_{D,i} = 0$ sur Γ ; il s'agit donc de fonctions propres de Neumann, respectivement de Dirichlet de Ω_i .

Voyons à présent que $\sigma(S) \supset \sigma_N(\Omega_i) \cup \sigma_D(\Omega_i)$. Pour cela, considérons f_i une fonction propre de Ω_i . Soit alors f le prolongement de f_i sur S défini par $f|_{\Omega_{3-i}} = f_i \circ \psi$ si f_i est une fonction propre de Neumann et par $f|_{\Omega_{3-i}} = -f_i \circ \psi$ si f_i est une fonction propre de Dirichlet. : il s'agit d'une fonction propre de S , de même valeur propre. □

6.2.1 Demi-tores

Les tores et leurs demi-tores constituent des situations auxquelles la proposition 6 s'applique. Considérons donc, par exemple, le tore de \mathbb{R}^3 défini à la section 6.1.1 et reprenons la fonction de paramétrisation φ . Deux types de demi-tores peuvent être considérés :

1. Les demi-tores difféomorphes à des anneaux, définis comme images par la fonction φ de $[0, 1] \times]0, 0.5[$ et de $[0, 1] \times]0.5, 1[$. Il s'agit des surfaces obtenues par coupe du tore le long des *parallèles* donnés par $x_3 = 0$ (*équateurs*). L'isométrie ψ à considérer est alors l'inversion de l'axe vertical Ox_3 : $\psi(x_1, x_2, x_3) = (x_1, x_2, -x_3)$.
2. Les demi-tores difféomorphes à des cylindres, par exemple définis comme images par la fonction φ de $]0, 0.5[\times [0, 1]$ et de $]0.5, 1[\times [0, 1]$. Il s'agit des surfaces obtenues par coupe du tore le long des *méridiens* donnés par $x_2 = 0$. L'isométrie ψ à considérer est alors l'inversion de l'axe Ox_2 : $\psi(x_1, x_2, x_3) = (x_1, -x_2, x_3)$.

Le tableau ci-dessous, résumant les résultats numériques obtenus pour le tore et ses demi-tores pour une résolution de triangulation raisonnable, met en évidence la propriété de symétrie donnée par la proposition 6.

Tore de \mathbb{R}^3 $r_1 = 1, r_2 = 0.5$ 32×64	<i>Anneau</i> (Dirichlet) 16×64	<i>Anneau</i> (Neumann) 16×64	<i>Cylindre</i> (Dirichlet) 32×32	<i>Cylindre</i> (Neumann) 32×32
0.00		0.00		0.00
1.00		1.00		1.00
1.00		1.00	1.00	
3.17		3.17		3.17
3.17		3.17	3.17	
3.90	3.90			3.90
4.48		4.48		4.48
5.04	5.04			5.04
5.04	5.04		5.04	

Les illustrations ci-dessous représentent quand à elles les fonctions propres associées aux différents demi-tores, avec les conditions adéquates. Elles sont à mettre en correspondance avec les fonctions propres du tore de la section 6.1.1 (voir figure 6.6).

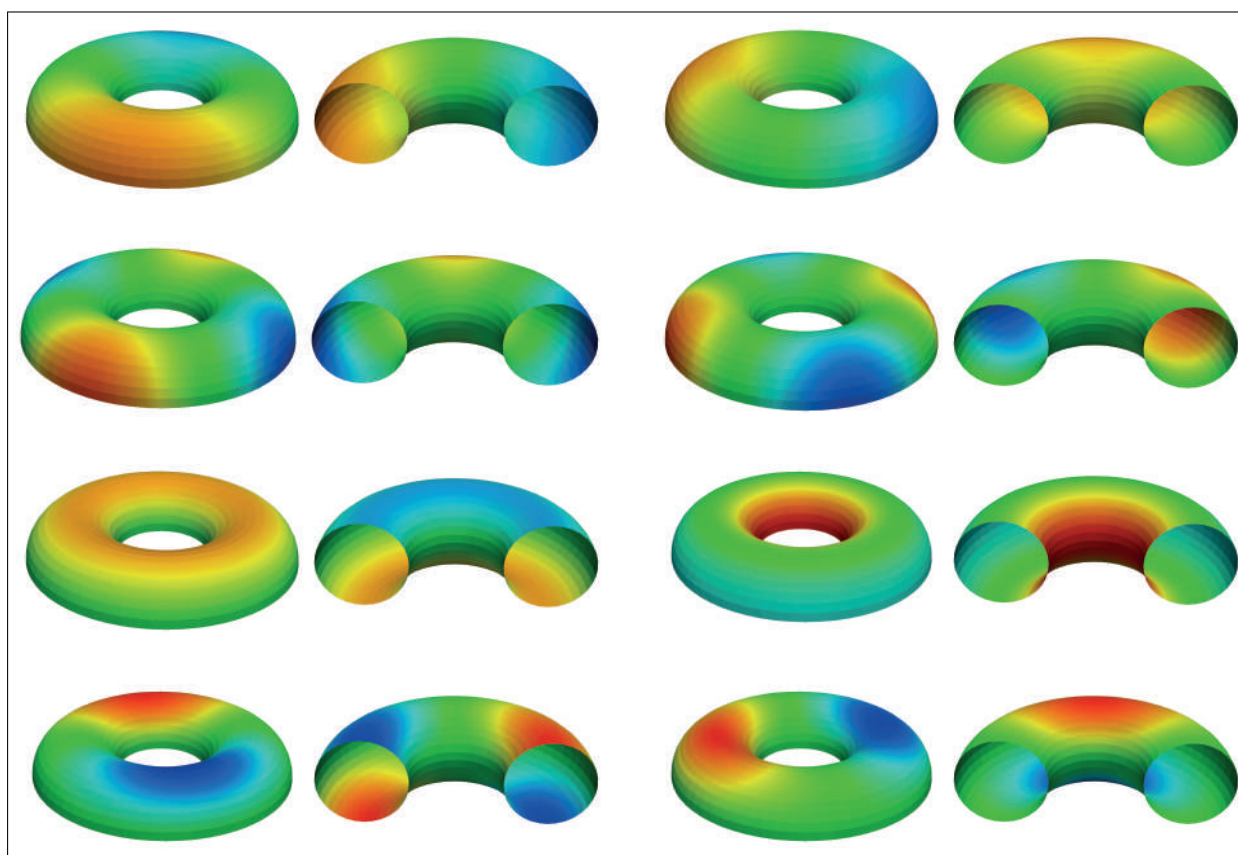


FIGURE 6.11 – Fonctions propres non triviales des demi-tores correspondant aux premières fonctions propres non triviales du tore.

6.2.2 Hémisphères

La proposition 6 s'applique indépendamment aux deux sphères définies à la section 6.1.2 à savoir aux sphères plate et ronde, ainsi qu'à leurs hémisphères (D_1 et D_2 , pour reprendre les notations employées précédemment). Il s'agit en effet de définir ψ comme la fonction associant à un point (x_1, x_2) de D_i le point $(x_1, -x_2)$ de D_{3-i} . Autrement dit et pour reprendre la représentation usuelle de la sphère dans \mathbb{R}^3 , ψ est l'inversion de l'axe Ox_3 , i.e. la fonction associant tout point (x_1, x_2, x_3) à $(x_1, x_2, -x_3)$.

Remarquons que la propriété de symétrie permet de déterminer le spectre de la sphère plate à partir des spectres de Dirichlet et de Neumann du disque : conformément au raisonnement exposé à la section 6.1.2, il s'agit des carrés des zéros des fonctions de Bessel et de leurs dérivées.

Les tableaux ci-dessous, résumant les résultats numériques obtenus pour chacune des sphères considérées et leurs hémisphères respectives pour une résolution de triangulation

raisonnable, met en évidence la propriété de symétrie donnée par la proposition 6.

Sphère plate $24 \times 24 / 12 \times 24$	Disque (Dirichlet)	Disque (Neumann)
0.00		0.00
3.39		3.39
3.39		3.39
5.78	5.78	
9.31		9.31
9.32		9.32
14.66		14.66
14.68	14.67	
14.68	14.68	

Les fonctions propres associées à l'hémisphère plate, c'est-à-dire au disque, sont représentées dans les figures 6.7 et 6.8; elles sont à mettre en correspondance avec celles de la sphère plate (figure 6.9).

Sphère ronde $24 \times 24 / 12 \times 24$	Hémisphère (Dirichlet)	Hémisphère (Neumann)
0.00		0.00
2.00	2.00	
2.00		2.00
2.00		2.00
5.99	5.99	
5.99	5.99	
5.99		5.99
5.99		6.00
5.99		6.00

Il est facile de se représenter les fonctions propres de l'hémisphère à partir de la figure 6.10, illustrant les premières fonctions propres non triviales de la sphère; aussi ce travail est-il laissé au soin du lecteur.

6.3 Domaines isospectraux

Dans cette section sera illustré le calcul numérique des premières valeurs propres de deux paires de domaines polygonaux isospectraux (tant pour les conditions de Dirichlet que pour celles de Neumann) non isométriques. Ces domaines sont dûs à Gordon, Webb et Wolpert

(voir [24]), comme réponse pour les domaines du plan à la question de géométrie spectrale de Kac (voir [29])

Peut-on entendre la forme d'un tambour ?

à comprendre comme

Le spectre d'un domaine définit-il le domaine, à isométrie près ?

Pour expliquer la construction des exemples que j'ai choisi de modéliser, reprenons la stratégie détaillée par Buser, Conway, Doyle et Semmler (voir [11]). Partons de la paire de domaines isométriques illustrés par la figure 6.12 et déformons les triangles équilatéraux constituant ces domaines en triangles isocèles rectangles de sorte que chacune de ces briques puisse être obtenue comme image de chacune de ses voisines par la symétrie d'axe donné par leur côté commun. Les domaines polygonaux (non isométriques) obtenus sont illustrés par la figure 6.13.

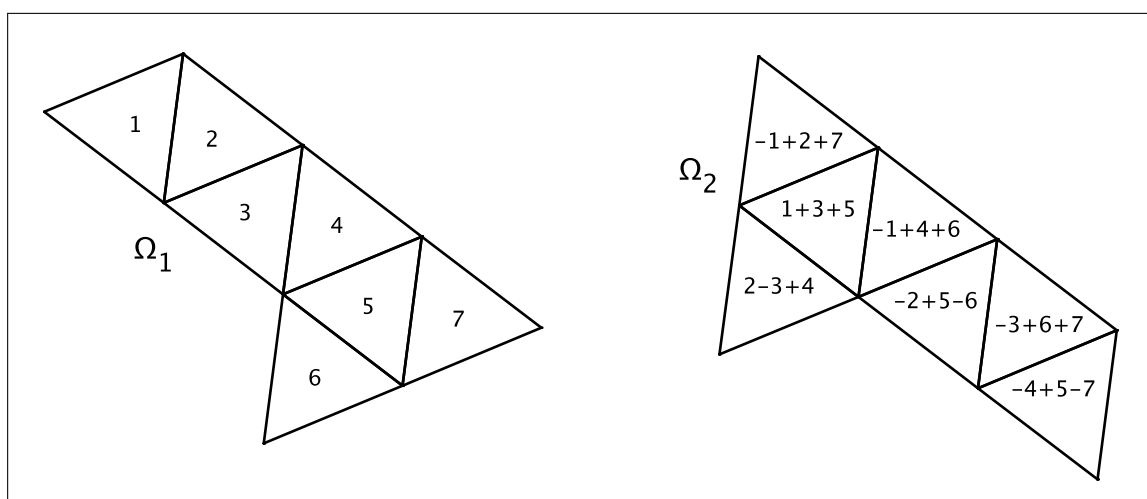


FIGURE 6.12 – Une paire de domaines isométriques servant à la construction de domaines isospectraux non isométriques.

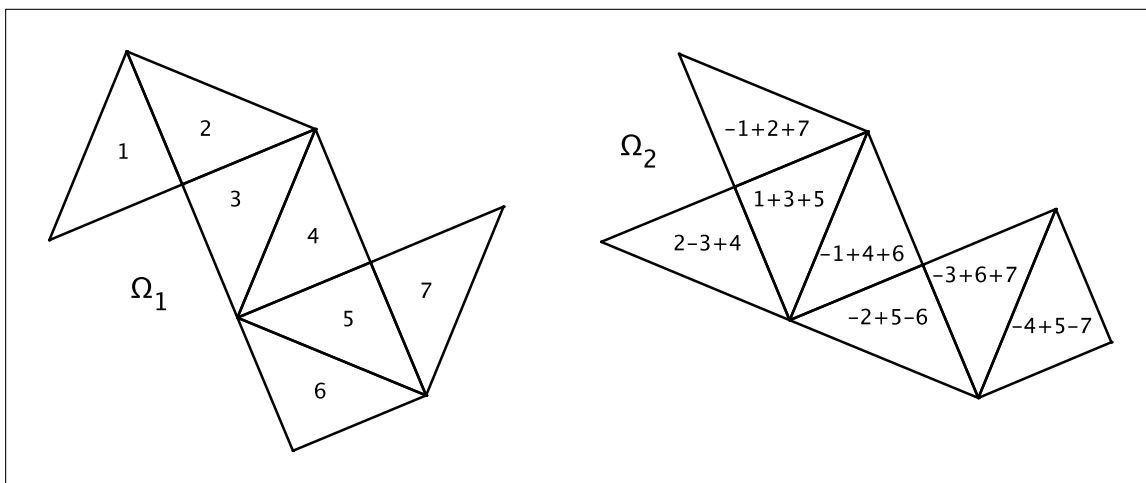


FIGURE 6.13 – Une première paire de domaines isospectraux non isométriques.

Pour la deuxième paire de domaines isospectraux, il s'agit de déformer les briques jusqu'à alors triangulaires en le polygone illustré à la figure 6.14, conformément à ce dernier schéma ; les domaines polygonaux (non isométriques) obtenus sont illustrés par la figure 6.15.

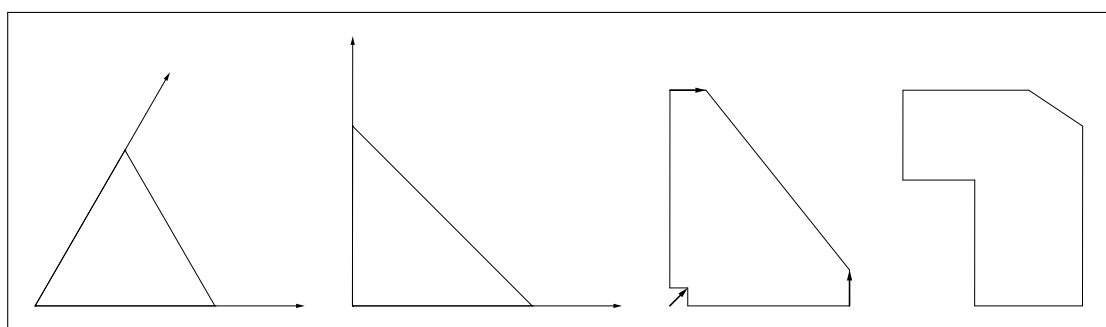


FIGURE 6.14 – Déformation d'une brique.

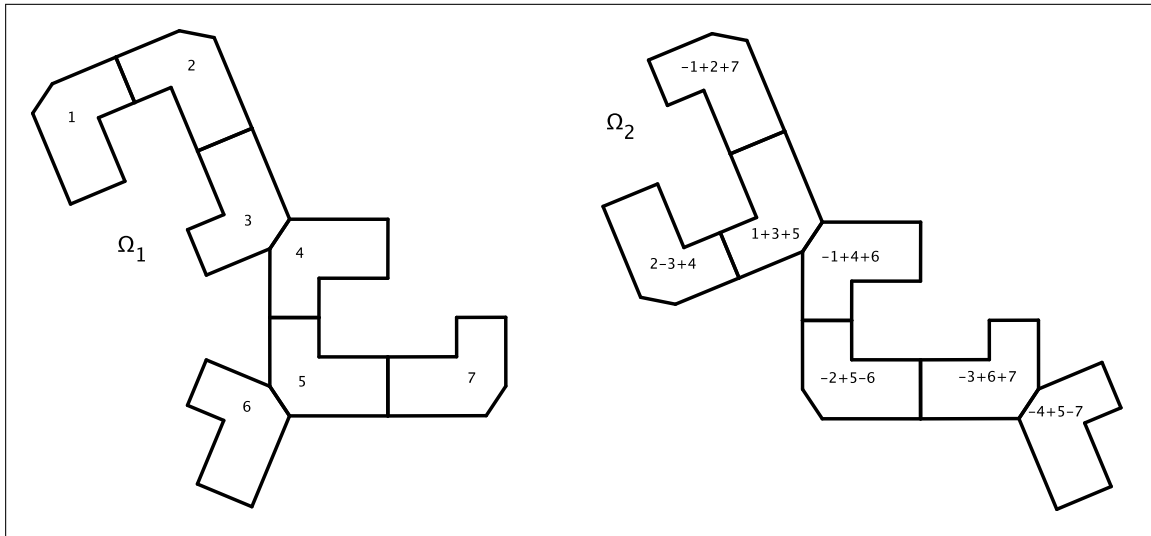


FIGURE 6.15 – Une deuxième paire de domaines isospectraux non isométriques.

La preuve de l'isospectralité est identique pour chacune de ces paires de domaines. Elle est basée sur la méthode de transplantation des fonctions propres (voir [11]), que Bérard applique aux domaines qui nous intéressent de la manière suivante (voir [8]). Il s'agit de montrer que, à toute fonction propre sur un des domaines, correspond une et une seule fonction propre du domaine dual. Soit donc u une fonction propre (de Dirichlet ou de Neumann) pour le domaine Ω_1 . Pour tout $i = 1, \dots, 7$, notons u_i les restrictions de u aux briques constituant Ω_1 , conformément aux figures 6.12, 6.13 ou 6.15 (selon la paire de domaines considérés).

On considère alors sur Ω_2 la fonction dont les restrictions aux différentes briques sont données par $-u_1 + u_2 + u_7$, $u_1 + u_3 + u_5$, \dots , $-u_4 + u_5 - u_7$, conformément aux figures 6.12, 6.13 ou 6.15. Il est alors facile de voir que cette fonction est une fonction propre (de Dirichlet ou de Neumann, selon la nature de u), et en particulier qu'elle est lisse.

Concernant le calcul numérique des valeurs propres de ces domaines, le maillage de la première paire de domaine ne pose aucune difficulté, les macro-éléments correspondant exactement aux briques constituant les domaines. Pour la deuxième paire de domaines, il s'agit de décomposer chaque brique en 3 macro-éléments, à savoir deux quadrilatères et un triangle. Il est également utile de procéder à un raffinement des triangulations obtenues au voisinage des angles rentrants, afin d'améliorer la convergence des valeurs approchées.

Les tableaux ci-dessous résument les résultats numériques obtenus avec des conditions de Neumann pour chaque paire de domaines isospectraux. Les triangulations, de différentes tailles, ont chacune été complétées d'une itération de raffinement au voisinage des angles rentrants. De par la nature de la preuve de l'isospectralité de ces domaines (par transplan-

tation), nous pouvions prédire une isospectralité numérique très précise, ce qui a bien été vérifié. Les illustrations ci-dessous permettent de plus de comparer les fonctions propres des deux domaines (pour la plus haute des résolutions considérées); toutefois, à nouveau de par la nature de la preuve de l'isospectralité de ces domaines (par transplantation), le lien entre les fonctions propres des deux domaines n'est pas visuellement évident.

Briques triangulaires (Neumann)	7 macro-él. 5×5		7 macro-él. 10×10		7 macro-él. 20×20	
	Ω_1	Ω_2	Ω_1	Ω_2	Ω_1	Ω_2
Taille des matrices	156	156	561	561	2121	2121
Largeur de bande	34	29	65	55	125	105
Nombre d'itérations	248	257	525	531	1283	1475
λ_0	0.00	0.00	0.00	0.00	0.00	0.00
λ_1	0.85	0.85	0.85	0.85	0.85	0.85
λ_2	3.31	3.31	3.26	3.26	3.25	3.25
λ_3	4.21	4.21	4.23	4.23	4.23	4.23
λ_4	7.37	7.37	7.43	7.43	7.44	7.44
λ_5	9.64	9.64	9.83	9.83	9.86	9.86
λ_6	10.58	10.58	10.83	10.83	10.89	10.89

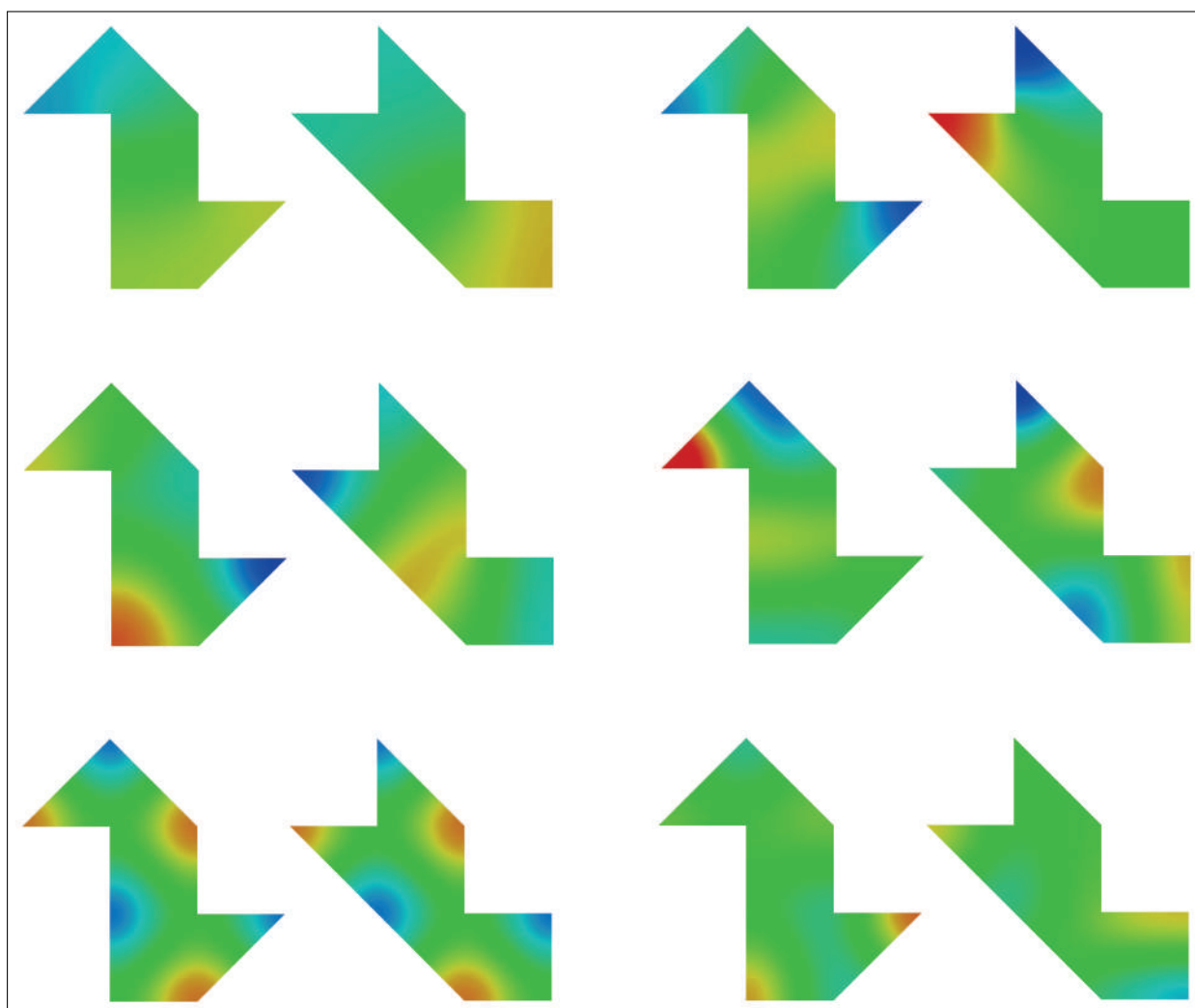


FIGURE 6.16 – Premières fonctions propres non triviales de la paire de domaines isospectraux à base triangulaire, conditions de Neumann.

Briques heptagonales (Neumann)	21 macro-él. $6 \times 6 / 3 \times 6$		21 macro-él. $12 \times 12 / 12 \times 6$		21 macro-él. $24 \times 24 / 24 \times 12$	
	Ω_1	Ω_2	Ω_1	Ω_2	Ω_1	Ω_2
Taille des matrices	1415	1415	4907	4909	18222	18224
Largeur de bande	40	38	72	73	128	146
Nombre d'itérations	2231	2172	7993	8363	36029	34898
λ_0	0.00	0.00	0.00	0.00	0.00	0.00
λ_1	0.13	0.13	0.13	0.13	0.13	0.13
λ_2	0.49	0.49	0.48	0.48	0.48	0.48
λ_3	0.67	0.67	0.67	0.67	0.66	0.66
λ_4	1.36	1.36	1.35	1.35	1.35	1.35
λ_5	2.59	2.60	2.58	2.58	2.57	2.57
λ_6	2.87	2.87	2.85	2.85	2.84	2.84

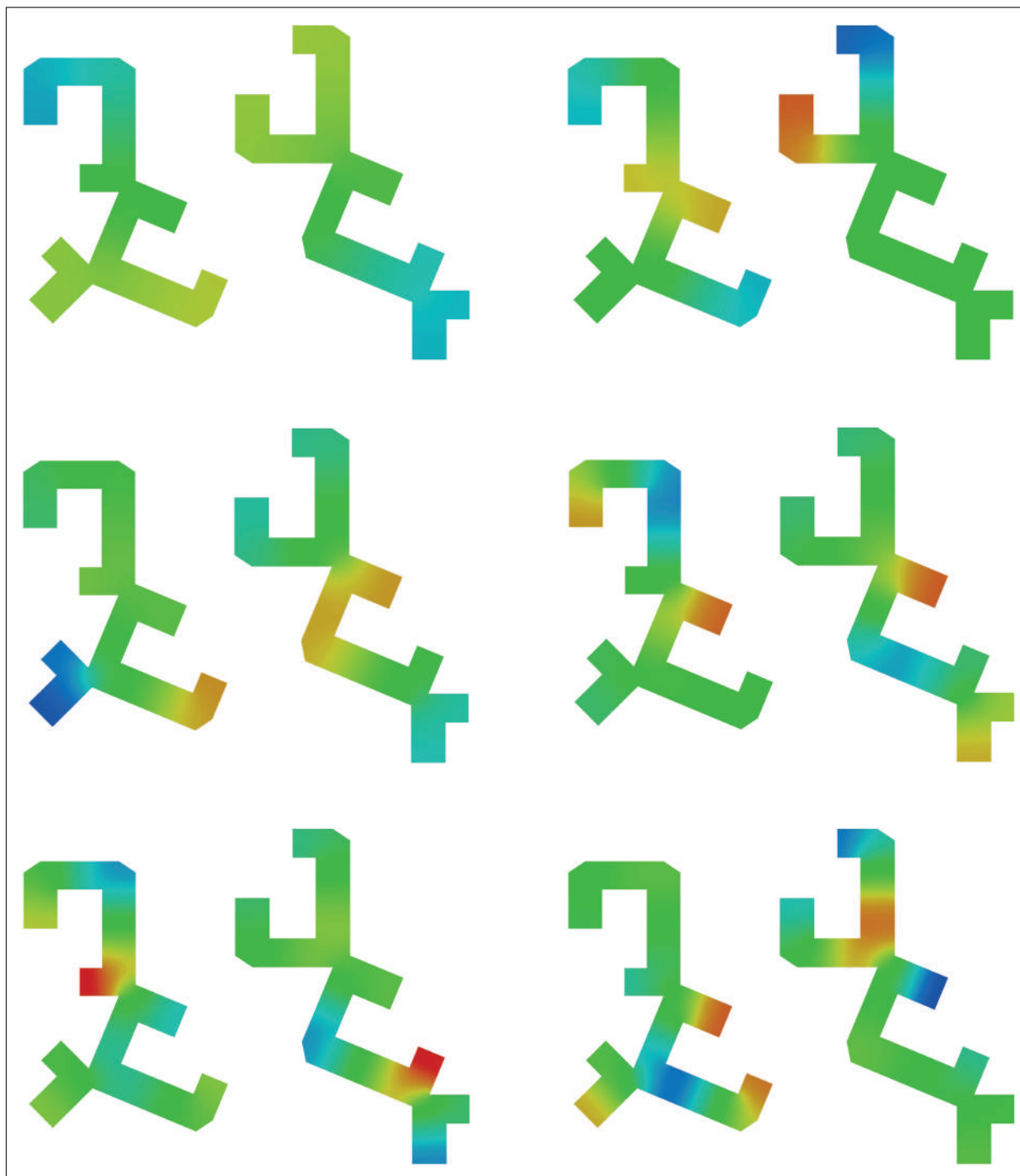


FIGURE 6.17 – Premières fonctions propres non triviales de la paire de domaines isospectraux à base heptagonale, conditions de Neumann.

6.4 Domaines à anses fines

Dans cette section seront présentées quelques unes des expérimentations numériques ayant notamment permis de conjecturer le résultat donné par le théorème 10 du chapitre 4 à propos des domaines à anses fines. Elles indiquent de plus une possible généralisation de ce résultat aux anses cylindriques. Bien que des conditions au bord de Dirichlet ou mixtes aient été considérées, seules les conditions de Neumann seront illustrées ici.

Les domaines présentés consistent en des carrés de côté 1 auxquels sont ajoutées des anses droites de longueur 1, tantôt planes, tantôt cylindriques. Dans les deux cas, des raccords bruts (\mathcal{C}^0) et lisses (\mathcal{C}^1) sont considérés entre l'anse et la partie épaisse. Les résultats numériques obtenus sont associés à ceux du carré $[0, 1]^2$ et du segment $[0, 1]$ conformément au théorème 10, afin de mettre en évidence la convergence des spectres. Ainsi, les conditions au bord examinées sur le carré sont de Neumann, alors que celles du segment sont mixtes (conditions de Neumann à l'extrémité correspondant au bout de l'anse et conditions de Dirichlet à celle correspondant au raccord entre le carré et l'anse).

Rappelons que le calcul manuel du spectre de Neumann du carré $[0, 1]^2$ (voir section 6.1.1) donne les valeurs et vecteurs propres

$$\begin{aligned}\lambda_{n,m} &= (n^2 + m^2) \pi^2, \\ u_{n,m}(x) &= C \cos(n\pi x_1) \cos(m\pi x_2),\end{aligned}$$

avec $n, m \in \mathbb{N}$ et $C \in \mathbb{R}$ une constante. En particulier, les premières valeurs propres sont

- | | |
|--|---|
| 1. $\lambda_{0,0} = 0$ | 4. $\lambda_{0,2} = \lambda_{2,0} = 4\pi^2$ (valeur double) |
| 2. $\lambda_{0,1} = \lambda_{1,0} = \pi^2$ (valeur double) | 5. $\lambda_{1,2} = \lambda_{2,1} = 5\pi^2$ (valeur double) |
| 3. $\lambda_{1,1} = 2\pi^2$ | 6. $\lambda_{2,2} = 8\pi^2$ |

Un rapide calcul manuel du spectre mixte du segment $[0, 1]$, c'est-à-dire la résolution de l'équation différentielle

$$\left\{ \begin{array}{l} \text{Trouver } u : [0, 1] \rightarrow \mathbb{R} \text{ et } \lambda \in \mathbb{R} \text{ tels que :} \\ -u'' = \lambda u \text{ dans }]0, 1[\\ u(0) = 0 \\ u'(1) = 0, \end{array} \right.$$

donne les valeurs et vecteurs propres

$$\begin{aligned}\lambda_n &= \left(\frac{1}{2} + n\right)^2 \pi^2, \\ u_n(x) &= C \sin\left(\left(\frac{1}{2} + n\right) \pi x\right),\end{aligned}$$

avec $n \in \mathbb{N}$ et $C \in \mathbb{R}$ une constante. En particulier, les premières valeurs propres sont

1. $\lambda_1 = \pi^2/4$
2. $\lambda_2 = 9\pi^2/4$
3. $\lambda_3 = 25\pi^2/4$
4. $\lambda_4 = 49\pi^2/4$

Le traitement numérique d'un domaine à anse fine pose des problèmes évidents au niveau du maillage, puisque la taille des mailles dans la partie épaisse doit être sensiblement plus grande que dans l'anse. L'approche que j'ai adoptée consiste essentiellement à mailler indépendamment la partie épaisse et l'anse du domaine, puis à raffiner localement (et convenablement) la triangulation de la partie épaisse près du raccord pour permettre l'assemblage des deux triangulations ainsi obtenues (la compatibilité de ces dernières étant le point essentiel du processus).

6.4.1 Anses planes brutes

La famille de domaines présentée ici consiste en des domaines à anses planes brutes, c'est-à-dire tels que considérés à la section 4.2 (preuve du théorème de convergence sous hypothèses fortes); il s'agit en l'occurrence d'un carré de côté 1, auquel est ajoutée une anse rectangulaire de longueur 1, plus ou moins fine.

Le tableau qui suit indique quelques valeurs propres obtenues pour diverses largeurs d'anse. En ce qui concerne les triangulations appliquées à ces domaines, on trouve sur l'anse une densité de mailles fixe relativement à la largeur de l'anse (9 nœuds) et, sur la partie épaisse, une taille de triangulation essentiellement fixe (à savoir 16 nœuds sur 16, plus ceux ajoutés par un raffinement sur un voisinage de l'anse pour raccorder les deux parties du domaine).

On observe la convergence du spectre de ces domaines vers ceux du carré $[0, 1]^2$ et du segment $[0, 1]$, malgré le maillage relativement grossier de la partie épaisse (qu'il s'agirait d'affiner pour obtenir une meilleure convergence vers les valeurs propres du carré).

	Valeurs théoriques		Domaine à anse		
	Carré (Neumann)	Segment (mixte)	plane brute (Neumann)		
Largeur d'anse			0.13	0.06	0.03
Taille des matrices			1665	3328	7415
Largeur de bande			34	54	96
Nombre d'itérations			977	19262	10725
λ_0	0		0.00	0.00	0.00
λ_1		2.47	2.20	2.26	2.33
λ_2	9.87		9.83	9.82	9.82
λ_3	9.87		9.91	9.87	9.87
λ_4	19.74		19.38	19.72	19.70
λ_5		22.21	19.80	20.08	20.78
λ_6	39.48		38.97	38.97	38.97
λ_7	39.48		39.07	39.03	39.00
λ_8	49.35		46.78	47.49	48.03
λ_9	49.35		48.93	48.88	48.90

Les représentations suivantes illustrent quant à elles la convergence des fonctions propres, pour la largeur d'anse intermédiaire (0.06).

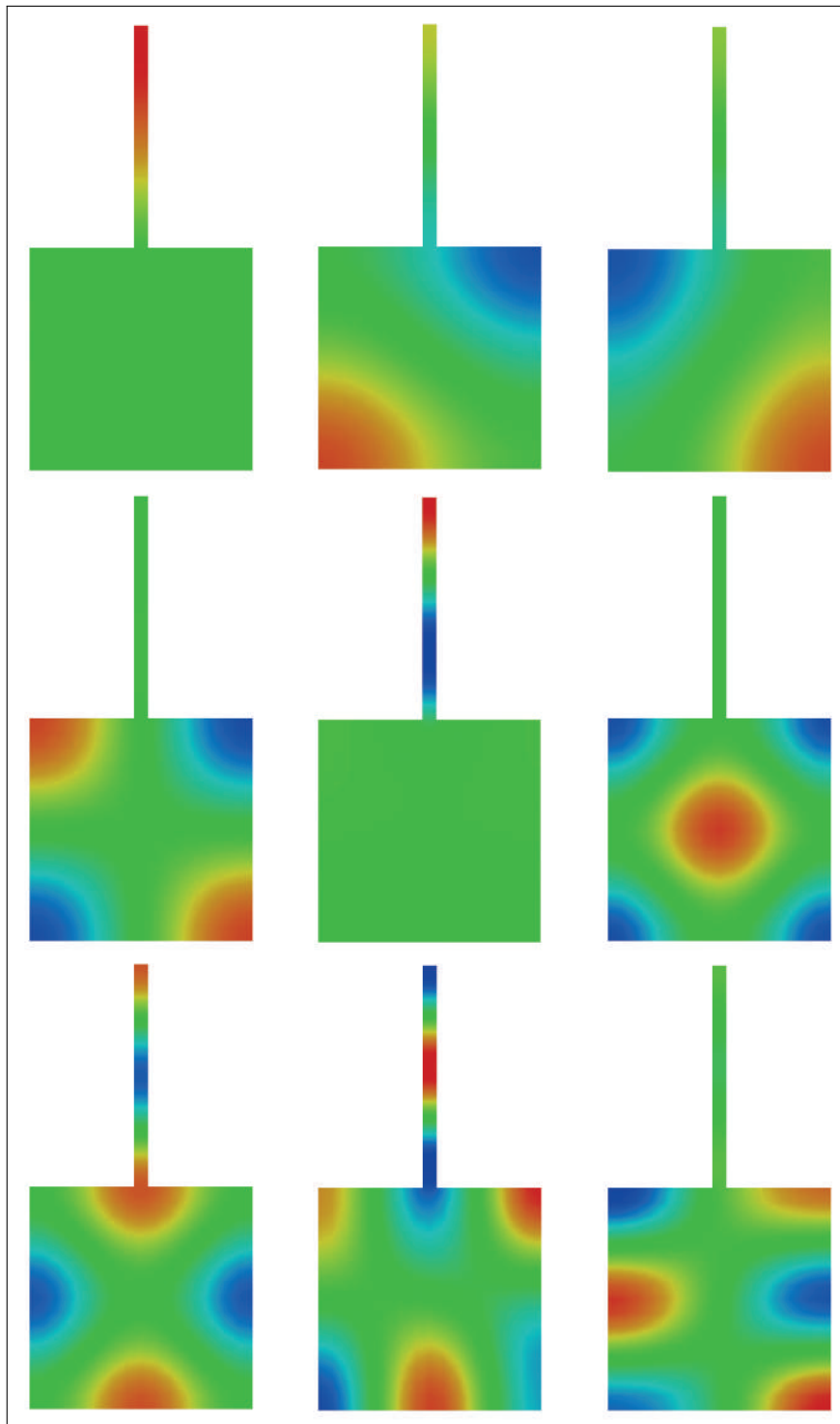


FIGURE 6.18 – Premières fonctions propres non triviales du domaine à anse fine plane brute, conditions de Neumann.

6.4.2 Autres anses

Ci-dessous sont présentés quelques résultats obtenus avec divers types de domaines à anses fines. Le tableau qui suit permet de comparer leurs spectres avec ceux du carré $[0, 1]^2$ et du segment $[0, 1]$. La largeur de l'anse de chacun de ces domaines est fixée à environ 0.03.

Le premier domaine considéré, possédant une anse plane brute, est l'exemple détaillé dans la section précédente. Le deuxième est essentiellement identique au premier, avec toutefois un raccord lisse entre les deux parties du domaine (raccord en arc de cercle, de rayon proportionnel à la largeur de l'anse ; voir figure 6.19) ; les modélisations de ces domaines sont donc très proches (voir section précédente).

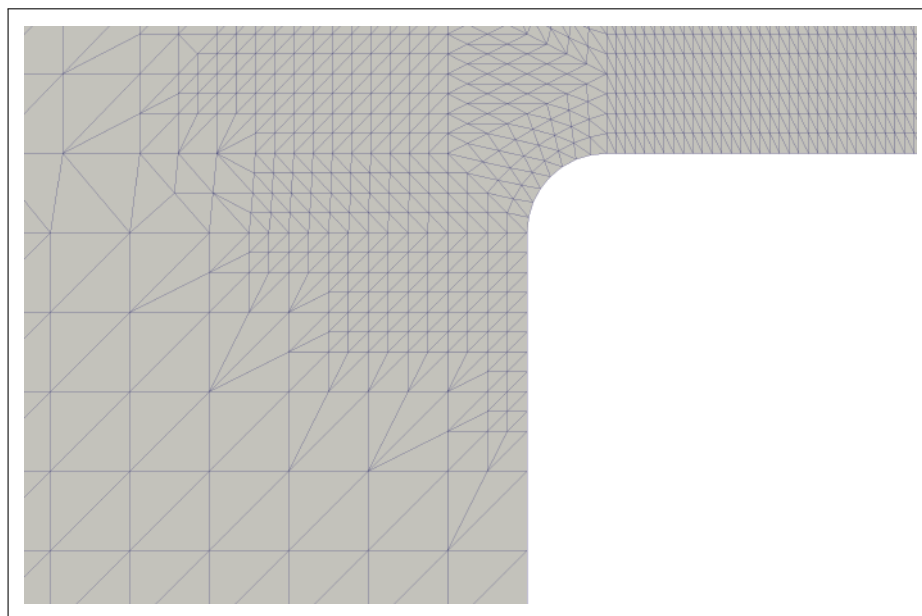


FIGURE 6.19 – Triangulation du raccord lisse de l'anse plane.

Les deux domaines suivants possèdent des anses cylindriques ; le raccord entre l'anse et la partie épaisse du premier est brut, alors que celui du deuxième est lisse (morceau de tore, de rayons proportionnels à la largeur de l'anse ; voir figure 6.20). A noter que la modélisation de ces domaines est plus délicate que celle des domaines à anses planes précédemment considérés, puisqu'il s'agit de trous convenablement la partie épaisse (en son centre) afin de préparer l'insertion de l'anse, de raffiner la triangulation au voisinage de ce trou afin de préparer l'assemblage des triangulations, puis finalement d'assembler les deux parties du domaine.

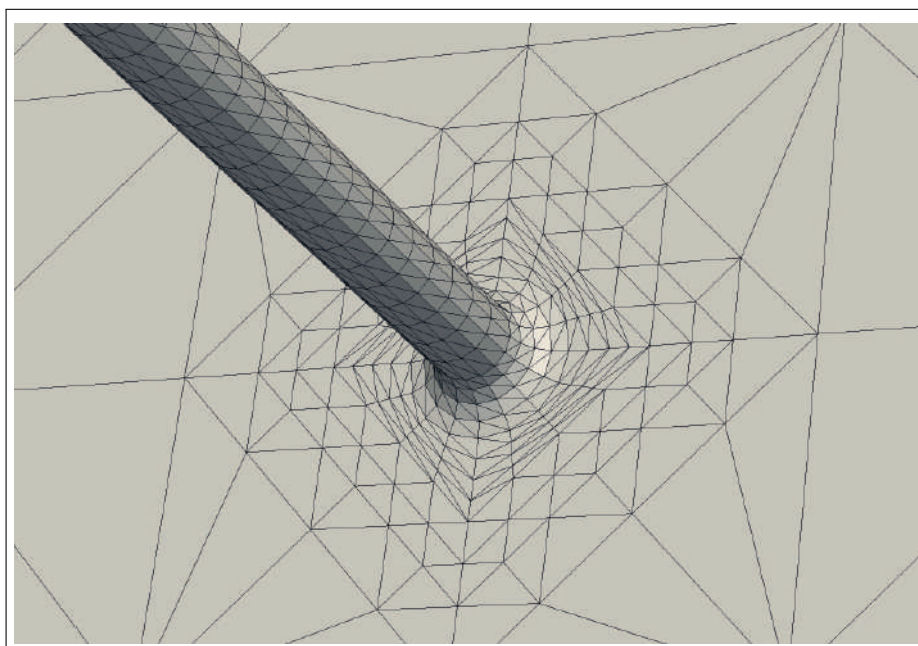


FIGURE 6.20 – Triangulation du raccord lisse de l'anse cylindrique.

	Valeurs théoriques		Domaine à anse			
	Carré (Neumann)	Segment (mixte)	plane		cylindrique	
			brute	lisse	brute	lisse
Taille des matrices			7415	5082	5328	5384
Largeur de bande			96	30	66	66
Nombre d'itérations			10725	60384	941038	276540
λ_0	0		0.00	0.00	0.00	0.00
λ_1		2.47	2.33	2.35	2.44	2.43
λ_2	9.87		9.82	9.81	9.86	9.86
λ_3	9.87		9.87	9.87	9.86	9.86
λ_4	19.74		19.70	19.69	19.75	19.75
λ_5		22.21	20.78	20.94	21.41	21.33
λ_6	39.48		38.97	38.96	39.25	39.25
λ_7	39.48		39.00	38.98	39.46	39.47
λ_8	49.35		48.03	48.01	49.21	49.21
λ_9	49.35		48.90	48.90	49.21	49.21

Les représentations suivantes illustrent quant à elles la convergence des fonctions propres du domaine à anse cylindrique brute, pour une largeur d'anse double de celle considérée ci-dessus, soit 0.06 (pour raison de lisibilité).

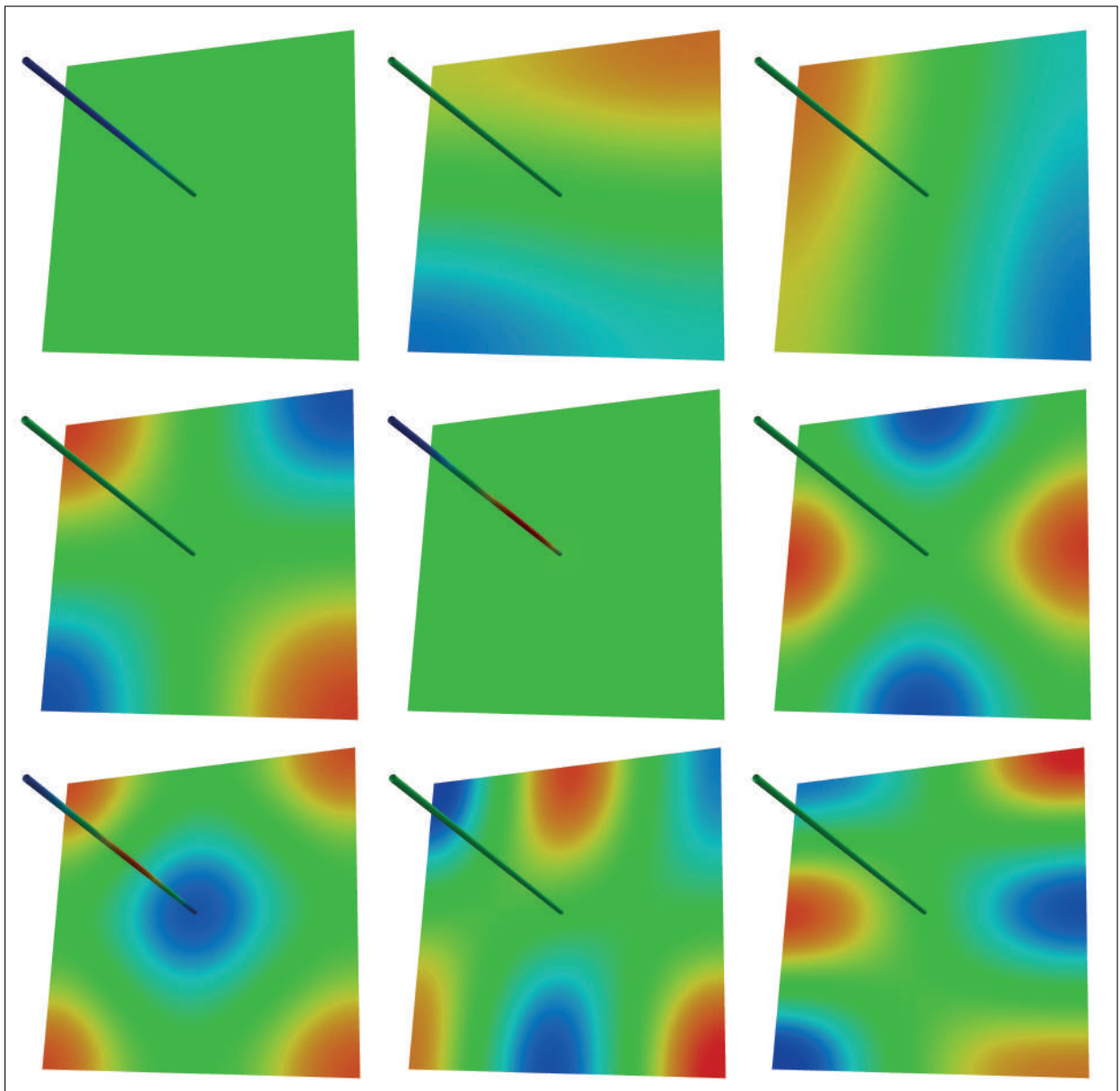


FIGURE 6.21 – Premières fonctions propres non triviales du domaine à anse fine cylindrique brute, conditions de Neumann.

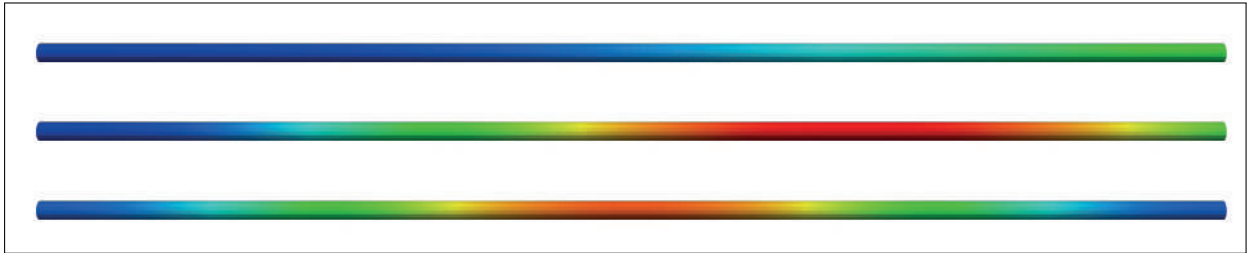


FIGURE 6.22 – Zoom sur l'anse cylindrique pour les première, cinquième et septième fonctions propres non triviales du domaine à anse cylindrique brute, conditions de Neumann.

6.5 Domaines optimaux

Les résultats obtenus à l'aide de l'algorithme décrit au chapitre 5 sont présentés dans cette section. Quelques commentaires s'avèrent toutefois nécessaires en préambule, afin de mettre en évidence la principale difficulté rencontrée ainsi que la manière de la contourner.

Bien que des routines permettant le changement de topologie aient été élaborées, elles n'ont jamais été mises à contribution par l'algorithme de variation de bord (en particulier, elles n'ont pas été mises à l'épreuve du debugging et il est donc possible qu'une certaine quantité de bugs subsiste). Il n'est toutefois pas impossible que les routines de changement de topologie développées soient utiles lors du traitement de valeurs propres d'ordres plus élevés, voir de problèmes connexes. Un ajustement des différents paramètres de la procédure d'optimisation de forme (notamment le facteur de déplacement du bord relativement à la dérivée de la fonction-coût) pourrait également permettre la modification de la topologie du domaine.

Une variation naturelle de la topologie n'ayant pas été observée au cours de l'algorithme (alors qu'il est connu que les domaines optimaux ont des topologies variables), il s'agit donc de considérer des domaines initiaux aux topologies diverses, en particulier au niveau du nombre de composantes connexes. Il est toutefois à noter que, particulièrement en cas de domaine non connexe, l'algorithme est confronté à la non-différentiabilité de valeurs propres multiples, obstacle fondamental de la méthode utilisée (pour la multiplicité des valeurs propres optimales sur les domaines non connexes, voir la preuve du théorème 12 ci-dessous). Dans la pratique, cela se manifeste par une modification cyclique du bord (la variation du bord cherchant tour à tour à optimiser l'une ou l'autre des fonctions propres correspondant à la multiplicité). Bien que ce phénomène n'ait finalement pas empêché l'algorithme d'approcher des domaines optimaux non connexes (notamment grâce à l'adaptation du facteur de déplacement du bord relativement à la dérivée de la fonction-coût), cela a passablement ralenti le processus.

Citons encore le résultat suivant, qui permet de déterminer des domaines optimaux ayant

plusieurs composantes connexes à partir des domaines optimaux précédents. Il nous servira à vérifier les valeurs numériques correspondant à des domaines non connexes.

Théorème 12 (*Wolf-Keller*)

Pour tout $j \in \mathbb{N} \setminus \{0\}$, notons λ_j^* la j^e valeur propre optimale parmi les domaines de \mathbb{R}^n et Ω_j^* un domaine associé, de volume 1. Supposons que, pour un certain $k \in \mathbb{N} \setminus \{0\}$, Ω_k^* soit non connexe. Alors

$$(\lambda_k^*)^{n/2} = \min_{1 \leq j \leq (k-1)/2} ((\lambda_j^*)^{n/2} + (\lambda_{k-j}^*)^{n/2}).$$

De plus, si i est un indice de $1 \leq j \leq (k-1)/2$ vérifiant le minimum ci-dessus, alors il existe un domaine $\tilde{\Omega}_k^*$ associé vérifiant

$$\tilde{\Omega}_k^* = \left(\left(\frac{\lambda_i^*}{\lambda_k^*} \right)^{1/2} \Omega_i^* \right) \amalg \left(\left(\frac{\lambda_{k-i}^*}{\lambda_k^*} \right)^{1/2} \Omega_{k-i}^* \right).$$

Preuve (tirée de [26])

Notons $\Omega_k^* = \Omega_1 \amalg \Omega_2$, avec $|\Omega_1|$ et $|\Omega_2|$ non nuls, et u_k^* une fonction propre associée. Supposons sans nuire à la généralité que u_k^* soit non identiquement nulle sur Ω_1 . Il s'agit donc d'une fonction propre de Ω_1 , disons $\lambda_k^* = \lambda_i(\Omega_1)$ (en cas de valeur propre multiple, considérons i maximal).

Alors on a $i < k$. En effet, sinon $i = k$ et alors le domaine $|\Omega_1|^{-1/n} \Omega_1$ serait de volume 1 et de k^e valeur propre $|\Omega_1|^{2/n} \lambda_k^* < \lambda_k^*$, ce qui contredit la minimalité de λ_k^* .

En corollaire, on a $\lambda_{k-i}(\Omega_2) \leq \lambda_k^*$. Mieux, on a même $\lambda_{k-i}(\Omega_2) = \lambda_k^*$, car sinon on pourrait effectuer une réduction de Ω_2 (complétée d'un agrandissement de Ω_1 afin de conserver un volume unitaire) de sorte à diminuer la k^e valeur propre, ce qui contredit à nouveau la minimalité de λ_k^* .

On a donc montré que la valeur propre λ_k^* a (au moins) une multiplicité correspondant à Ω_1 et (au moins) une autre correspondant à Ω_2 : $\lambda_k^* = \lambda_i(\Omega_1) = \lambda_{k-i}(\Omega_2)$.

Mais $\lambda_i(\Omega_1) = |\Omega_1|^{1/n} \lambda_i^*$, car Ω_1 et $|\Omega_1|^{1/n} \Omega_i^*$ ont même volume et $\lambda_k^* = \lambda_i(\Omega_1)$ est minimal. Ainsi, $|\Omega_1| = \left(\frac{\lambda_i^*}{\lambda_k^*} \right)^{n/2}$. Et on trouve de manière symétrique $|\Omega_2| = \left(\frac{\lambda_{k-i}^*}{\lambda_k^*} \right)^{n/2}$. Or, on a $|\Omega_1| + |\Omega_2| = |\Omega_k^*| = 1$, d'où

$$(\lambda_k^*)^{n/2} = (\lambda_i^*)^{n/2} + (\lambda_{n-i}^*)^{n/2}.$$

Considérons encore, pour tout $1 \leq j \leq n-1$, le domaine de volume 1

$$\tilde{\Omega}_j = \left(\left(\frac{(\lambda_j^*)^{n/2}}{(\lambda_j^*)^{n/2} + (\lambda_{n-j}^*)^{n/2}} \right)^{1/n} \Omega_j^* \right) \amalg \left(\left(\frac{(\lambda_{n-j}^*)^{n/2}}{(\lambda_j^*)^{n/2} + (\lambda_{n-j}^*)^{n/2}} \right)^{1/n} \Omega_{n-j}^* \right).$$

Remarquons qu'on vient de montrer que $\lambda_k(\tilde{\Omega}_i) = \lambda_k^*$.

Comme la j^e valeur propre de sa première composante et la $(k - j)^e$ valeur propre de sa deuxième composante valent

$$((\lambda_j^*)^{n/2} + (\lambda_{n-j}^*)^{n/2})^{2/n},$$

alors c'est aussi le cas de $\lambda_k(\tilde{\Omega}_j)$. Autrement dit, on a

$$(\lambda_k(\tilde{\Omega}_j))^{n/2} = (\lambda_j^*)^{n/2} + (\lambda_{n-j}^*)^{n/2}.$$

Or, comme $\lambda_k(\tilde{\Omega}_j) \leq \lambda_k^*$ et que $\lambda_k(\tilde{\Omega}_i) = \lambda_k^*$, alors l'indice i vérifie le minimum de $\lambda_k(\tilde{\Omega}_j)$ et, réciproquement, tout indice minimisant $\lambda_k(\tilde{\Omega}_j)$ fournit un domaine $\tilde{\Omega}_j$ optimal. \square

6.5.1 Présentation générale des domaines optimaux

Ci-dessous sont regroupés les domaines et valeurs propres obtenus numériquement à l'aide de l'algorithme d'optimisation de forme, résultats confortés par théorème 12 par exemple. Une illustration des fonctions propres associées ainsi que des commentaires concernant la résolution numérique des différentes valeurs propres considérées et les résultats obtenus seront donnés dans la section suivante.

L'ensemble des domaines présentés a été déterminé à partir de domaines initiaux consistant en un ou deux carrés isométriques (selon le nombre de composantes connexes), dont le bord est constitué de 40 nœuds. La plupart des calculs ont demandé entre 60 et 100 itérations de variation de bord, même si quelques valeurs propres en ont nécessité davantage. Pour chaque valeur, deux ou trois raffinements de bord ont été effectués à différents moments de l'algorithme (selon le nombre de composantes connexes), pour un nombre de nœuds du bord avoisinant les 300. De plus, dans le but de garantir une certaine qualité des valeurs propres correspondant aux domaines calculés, deux itérations de raffinement supplémentaires ont été effectuées une fois le domaine optimal déterminé : la première sur le domaine entier et la deuxième autour du bord uniquement (pour une résolution finale du bord avoisinant les 1000 nœuds).

A noter que l'algorithme d'optimisation de forme en lui-même a été effectué en utilisant le mass-lumping ; étant données les résolutions de triangulation considérées, les valeurs ainsi déterminées constituent de bonnes *estimations* des valeurs propres des domaines présentés. Toutefois, afin de calculer des *bornes supérieures* fiables aux valeurs propres des domaines optimaux obtenus (et donc aux domaines optimaux exacts), le mass-lumping n'a pas été employé lors des calculs finaux, effectués après raffinement des triangulations. Sauf indication contraire, les valeurs propres affichées ci-dessous sont donc **garanties supérieures** aux valeurs propres réelles.

Les résultats obtenus s'avèrent très satisfaisants, améliorant toutes les valeurs calculées par Oudet ([35]) à partir de la sixième et différant de moins d'un pourcent des valeurs obtenues par Antunes-Freitas ([5]) et des valeurs théoriques connues. On peut toutefois donner quelques pistes permettant d'augmenter très facilement l'efficacité de l'algorithme tel qu'utilisé ici. Premièrement, on remarque en observant les domaines de près un effet d'escalier sur leur bord, ce qui doit immanquablement perturber la minimisation des valeurs propres. Une procédure de lissage du bord constituerait donc un moyen facile de réduire les valeurs propres. Deuxièmement, l'utilisation de mass-lumping pendant l'optimisation de forme, bien pratique pour accélérer les nombreuses résolutions de systèmes linéaires au cours de l'algorithme, n'a pas pour seul effet négatif de modifier les valeurs propres calculées, mais également de perturber les fonctions propres associées et donc la déformation du bord du domaine. Il serait donc intéressant de rééditer ces expériences sans recourir au mass-lumping, quitte à repartir des domaines obtenus au préalable avec ce procédé. Finalement, l'algorithme d'optimisation est très sensible à bon nombre de paramètres et il est indiscutable qu'un réglage fin permettrait d'améliorer l'optimisation.

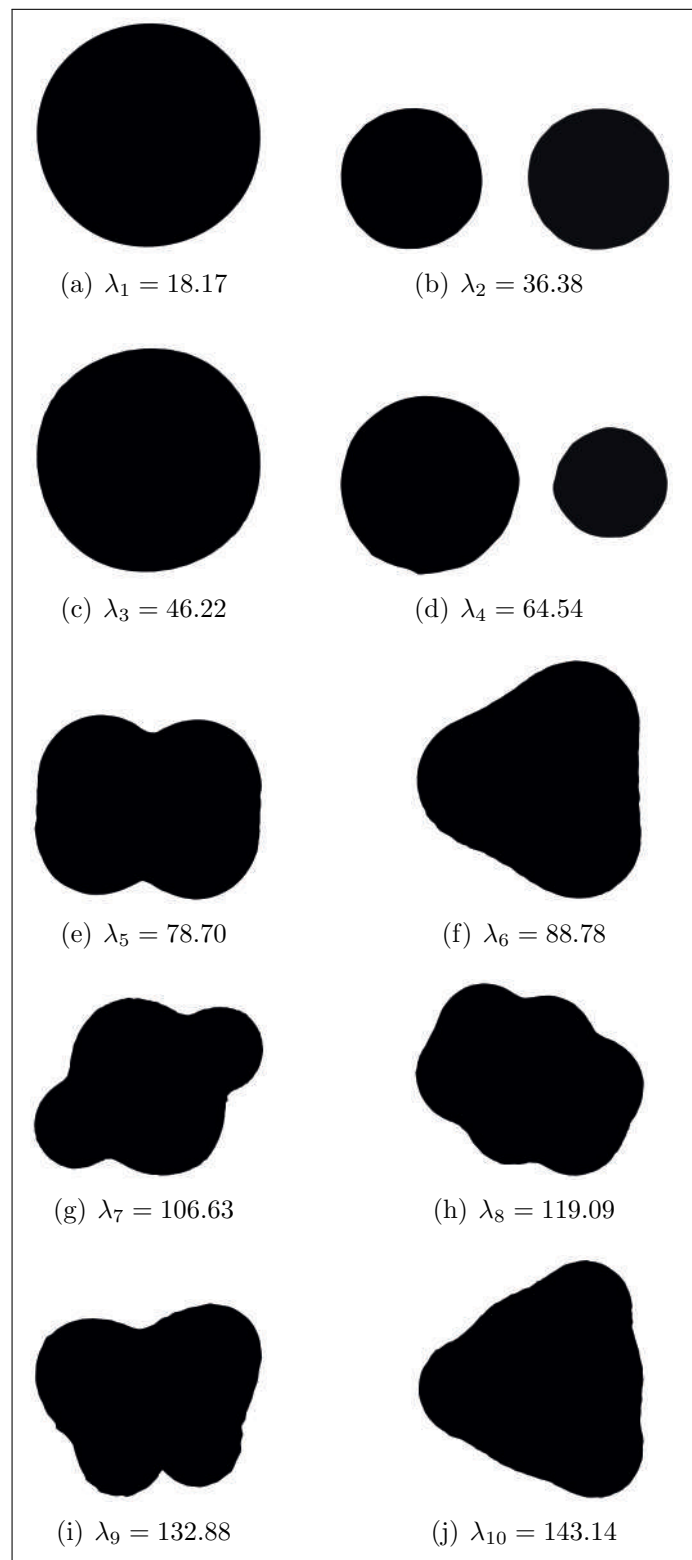


FIGURE 6.23 – Domaines optimaux déterminés numériquement et valeurs propres associées.

6.5.2 Présentation détaillée des domaines optimaux

Première valeur propre

Pour λ_1 , l'algorithme converge sans surprise vers le disque, dont la première valeur propre théorique vaut 18.17 (voir section 6.1.2) et qui constitue le domaine optimal pour la première valeur propre d'après le théorème 13 ci-dessous (dont la preuve est par exemple donnée dans [15]).

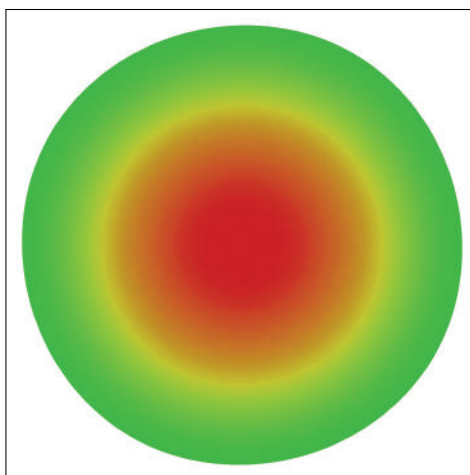


FIGURE 6.24 – Domaine optimal pour $\lambda_1 = 18.17$ (estimation à 18.17 avec mass-lumping).

Théorème 13 (*Faber-Krahn*)

Le domaine de \mathbb{R}^n de volume donné pour lequel la première valeur propre de Dirichlet est minimale est la boule.

Deuxième valeur propre

Pour λ_2 , l'algorithme converge, à partir d'un domaine initial constitué de deux composantes connexes (deux carrés isométriques en l'occurrence), vers la réunion disjointe de deux disques de même rayon, dont les deux premières valeurs propres théoriques valent 36.34 (conformément au théorème 12 et à la section 6.1.2) et qui constitue le domaine optimal pour la deuxième valeur propre d'après le théorème 14 ci-dessous.

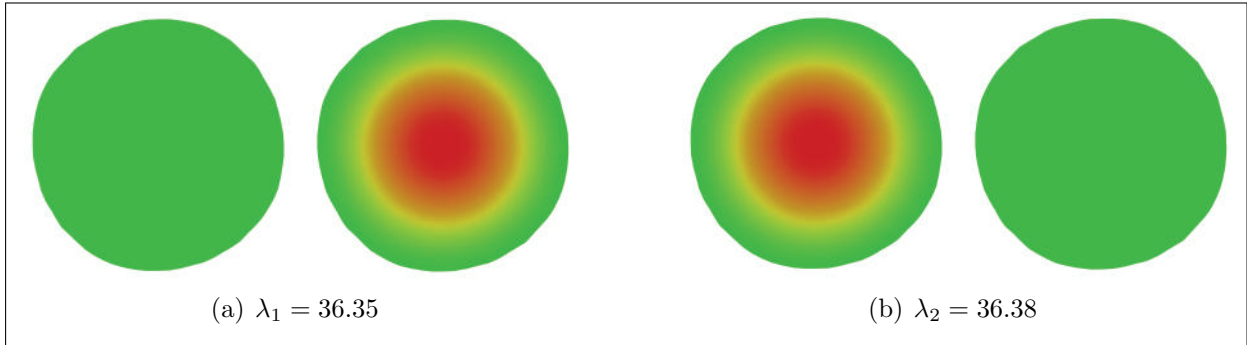


FIGURE 6.25 – Domaine optimal pour $\lambda_2 = 36.38$ (estimation à 36.36 avec mass-lumping).

Notons que la convergence à partir d'une seule composante connexe n'a malheureusement pas été observée lors des expérimentations. En effet, bien que l'algorithme semble se diriger vers une séparation du domaine en deux composantes connexes circulaires, les paramètres utilisés n'ont pas permis d'y parvenir.

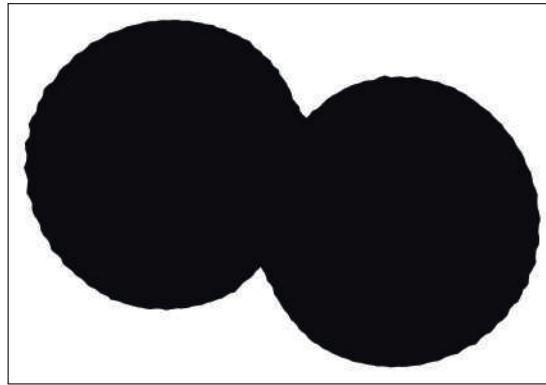


FIGURE 6.26 – Domaine obtenu à partir d'un carré, après 160 itérations ($\lambda_2 = 36.80$ avec mass-lumping).

Théorème 14 (*Krahn-Szegö*)

Le domaine de \mathbb{R}^n de volume donné pour lequel la deuxième valeur propre de Dirichlet est minimale est la réunion disjointe de deux boules de même rayon.

Preuve (tirée de [26])

Soit Ω un domaine de \mathbb{R}^n . Notons Ω_+ et Ω_- les domaines nodaux d'une deuxième fonction propre u_2 . Alors les restrictions de u_2 à Ω_+ et à Ω_- sont des fonctions propres de Dirichlet de leurs domaines de définition. De plus, comme elles ne changent pas de signe, il s'agit des premières fonctions propres :

$$\lambda_2(\Omega) = \lambda_1(\Omega_+) = \lambda_1(\Omega_-). \quad (6.1)$$

Notons Ω_+^* et Ω_-^* les boules de mêmes volumes que Ω_+ et Ω_- respectivement. D'après le théorème 13,

$$\lambda_1(\Omega_+^*) \leq \lambda_1(\Omega_+) \quad \text{et} \quad \lambda_1(\Omega_-^*) \leq \lambda_1(\Omega_-). \quad (6.2)$$

Considérons alors le domaine $\tilde{\Omega}$ défini comme la réunion disjointe de Ω_+^* et Ω_-^* :

$$\tilde{\Omega} = \Omega_+^* \amalg \Omega_-^* .$$

Ainsi, comme $\tilde{\Omega}$ est non connexe, $\lambda_2(\tilde{\Omega})$ vérifie

$$\lambda_2(\tilde{\Omega}) \leq \max\{\lambda_1(\Omega_+^*), \lambda_1(\Omega_-^*)\} \stackrel{(6.1)}{\leq} \max\{\lambda_1(\Omega_+), \lambda_1(\Omega_-)\} \stackrel{(6.2)}{=} \lambda_2(\Omega),$$

ce qui montre que le domaine optimal pour λ_2 est la réunion disjointe de deux boules. Mieux, ces boules ont même rayon ; sinon, on pourrait effectuer une réduction de la plus petite boule et un agrandissement de la plus grande de manière à conserver l'aire globale initiale et à diminuer λ_2 .

□

Troisième valeur propre

Pour λ_3 , l'algorithme converge vers le disque, dont la troisième valeur propre théorique vaut 46.12, ce qui va dans le même sens que la conjecture actuelle (voir [25]).

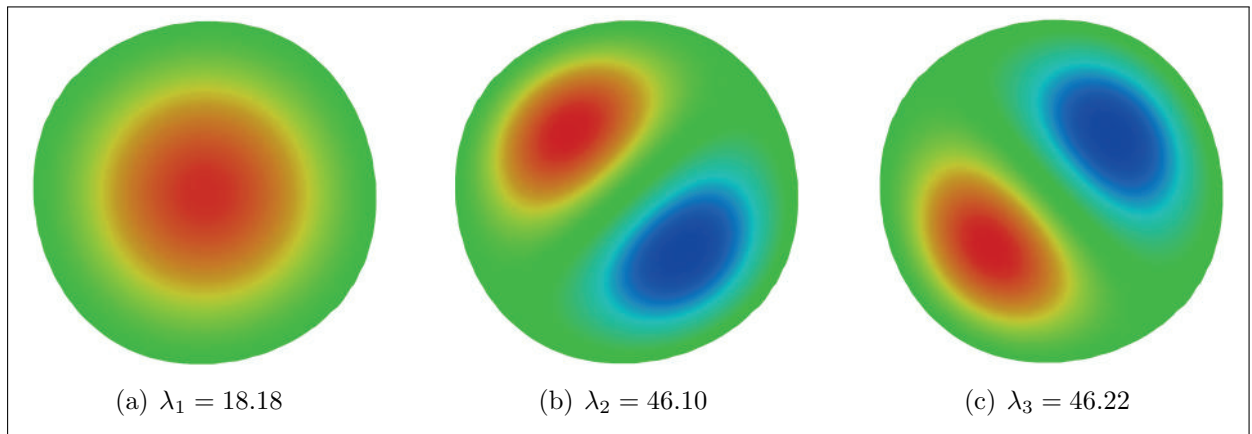


FIGURE 6.27 – Domaine optimal pour $\lambda_3 = 46.22$ (estimation à 46.19 avec mass-lumping).

Quatrième valeur propre

Pour λ_4 , l'algorithme semble converger, à partir d'un domaine initial constitué de deux composantes connexes (deux carrés isométriques en l'occurrence), vers la réunion disjointe de deux disques dont les rayons sont en rapport $(\lambda_1^*/\lambda_3^*)^{1/2} = 0.63$, conformément au

théorème 12 et à la conjecture sur λ_3 , pour une quatrième valeur propre théorique associée valant 64.29. La convergence est toutefois difficile et on peut regretter que le domaine illustré reste relativement éloigné de la réunion de disques conjecturée.

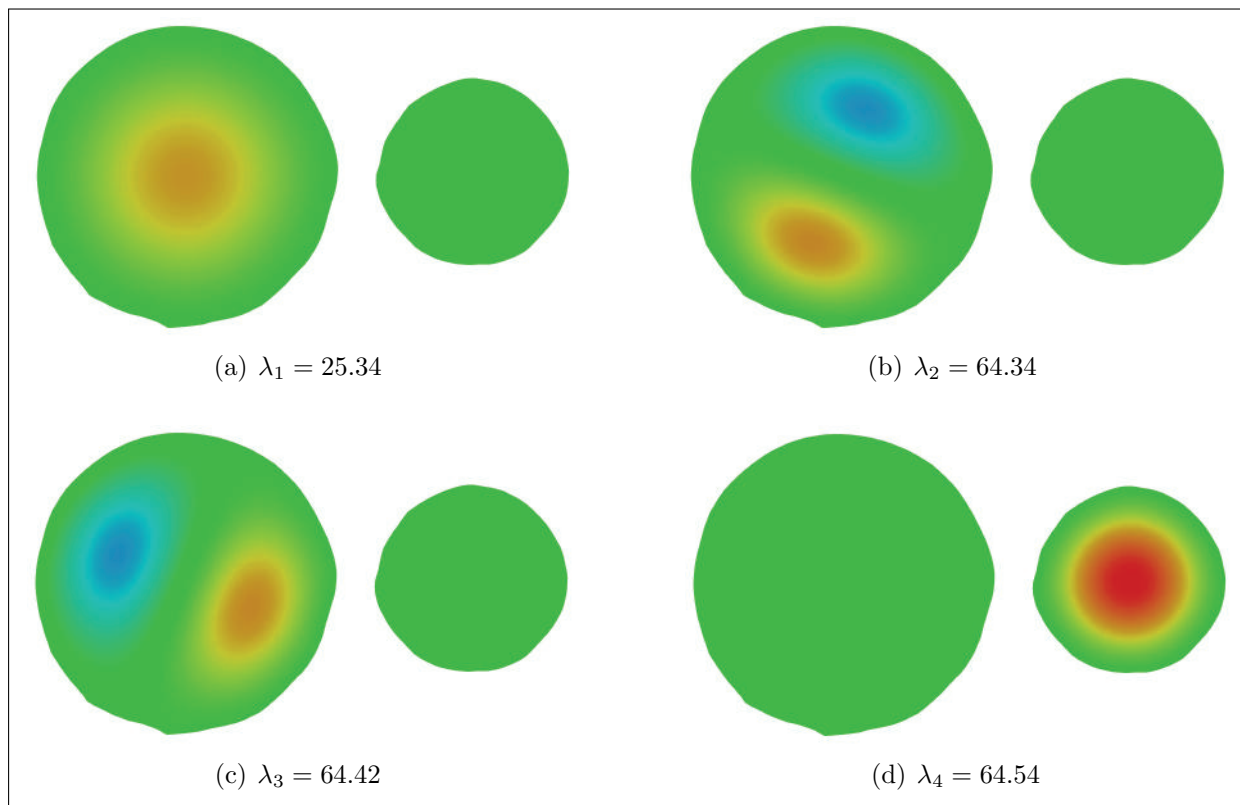


FIGURE 6.28 – Domaine optimal pour $\lambda_4 = 64.54$ (estimation à 64.41 avec mass-lumping).

Notons également que la convergence à partir d'une seule composante connexe (à savoir deux carrés isométriques) ne conduit qu'à un domaine connexe localement optimal, dont la quatrième valeur propre vaut 65.11 (calcul effectué avec mass-lumping).

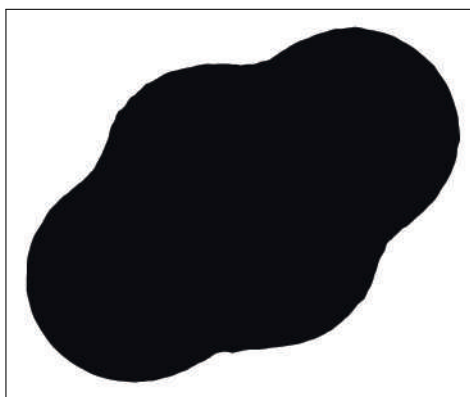


FIGURE 6.29 – Domaine localement optimal pour $\lambda_4 = 65.11$ (avec mass-lumping).

L'algorithme appliqué à la quatrième valeur propre s'avère donc doublement décevant, puisque peu satisfaisant à partir tant d'un domaine connexe (dont le résultat est un domaine localement optimal) que de deux composantes connexes (dont la convergence est poussive).

Cinquième valeur propre

Pour λ_5 , il est à noter que l'algorithme converge vers un domaine ne consistant pas en une réunion de disques, au contraire des quatre précédents domaines obtenus.

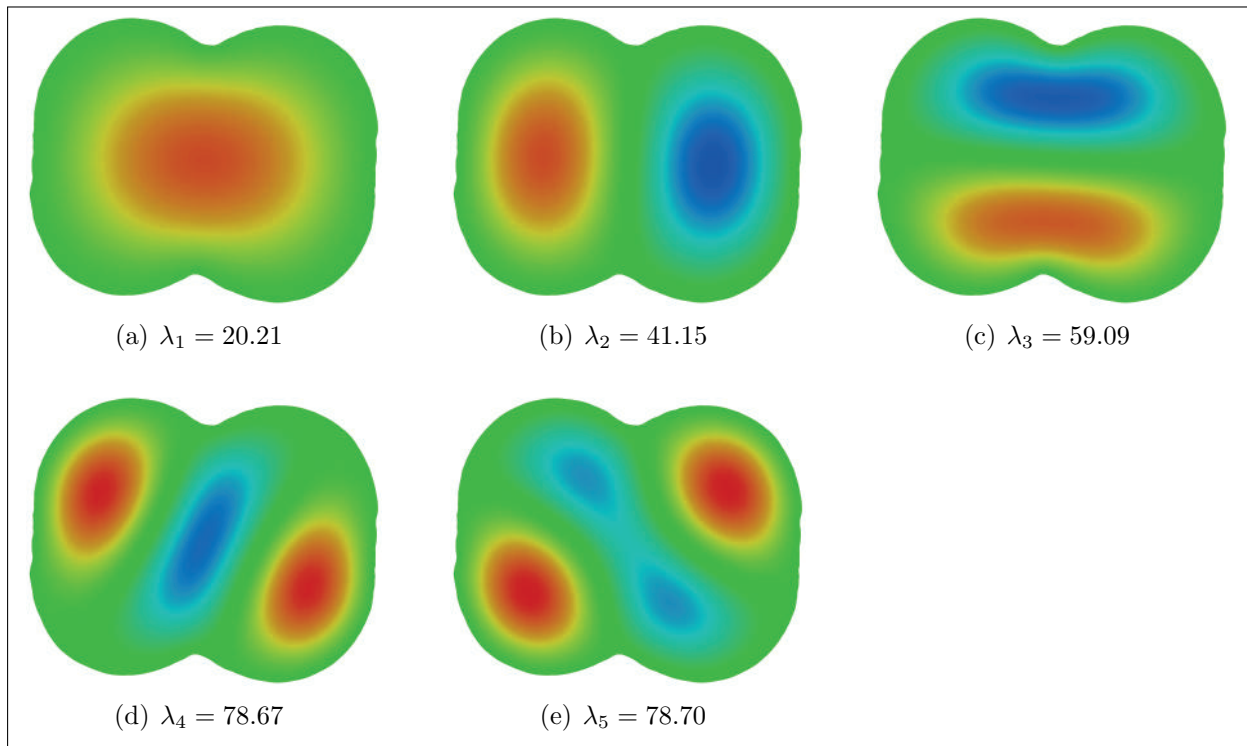
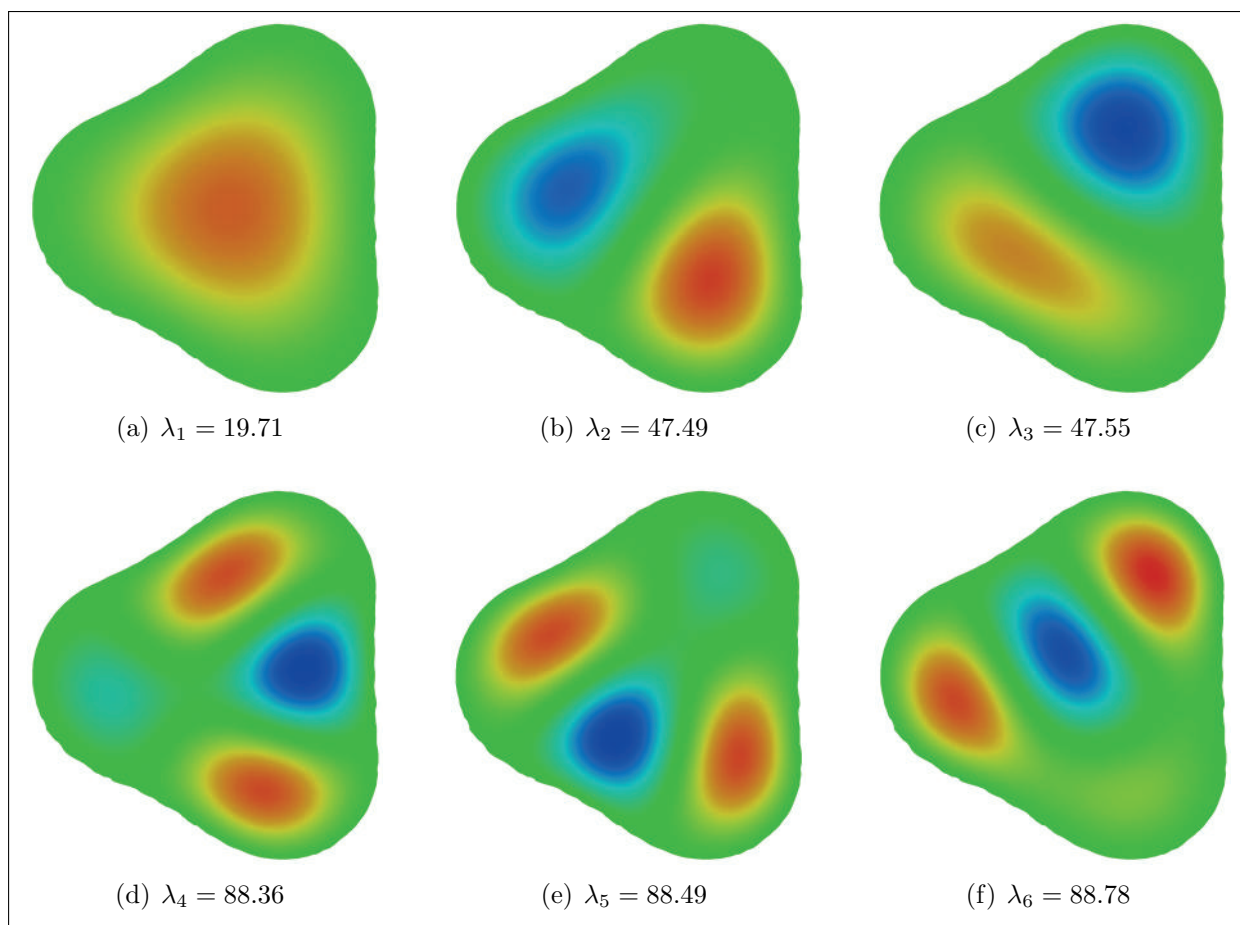


FIGURE 6.30 – Domaine optimal pour $\lambda_5 = 78.70$ (estimation à 78.63 avec mass-lumping).

Sixième valeur propre

FIGURE 6.31 – Domaine optimal pour $\lambda_6 = 88.78$ (estimation à 88.66 avec mass-lumping).

Septième valeur propre

Lors de sa découverte, le domaine obtenu pour λ_7 a constitué une surprise, puisqu'il est différent du domaine avancé par Oudet ([35]), seule référence à ce moment-là. En effet, le domaine illustré par ce dernier est constitué de deux composantes connexes et est associé à la valeur $\lambda_7 = 107.47$. Or, à partir d'un domaine non connexe (en l'occurrence deux carrés isométriques), mon algorithme conduit (non sans mal d'ailleurs) au domaine ci-dessous, bien différent de celui obtenu par Oudet et en désaccord avec le théorème 12 (l'une des composantes connexes ne correspondant à aucun domaine optimal précédent).

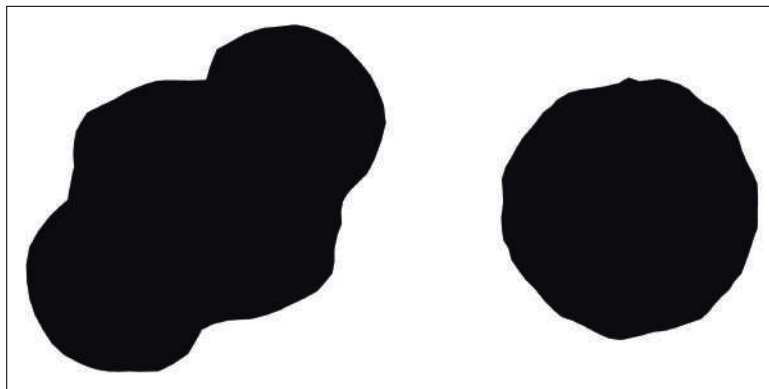


FIGURE 6.32 – Domaine obtenu à partir de deux carrés, après 60 itérations ($\lambda_7 = 111.92$ avec mass-lumping).

Il s'avère que ce domaine non connexe approche probablement un minimum local puisque, à partir d'un domaine connexe (carré), l'algorithme conduit au résultat ci-dessous, en accord avec le théorème 12 (puisque connexe et de septième valeur propre meilleure que le candidat donné par le théorème). De plus, comme la septième valeur propre du domaine obtenu est garantie inférieure à 106.63, on peut avancer que le domaine proposé par Oudet n'est pas optimal pour λ_7 .

Les récents calculs d'Antunes et Freitas ([5]) confirment la non-optimalité des domaines non-connexes et en particulier de celui avancé par Oudet. La valeur propre qu'ils ont obtenue s'avère d'ailleurs légèrement meilleurs que la mienne et leur domaine sensiblement différent (il possède en particulier moins de symétries).

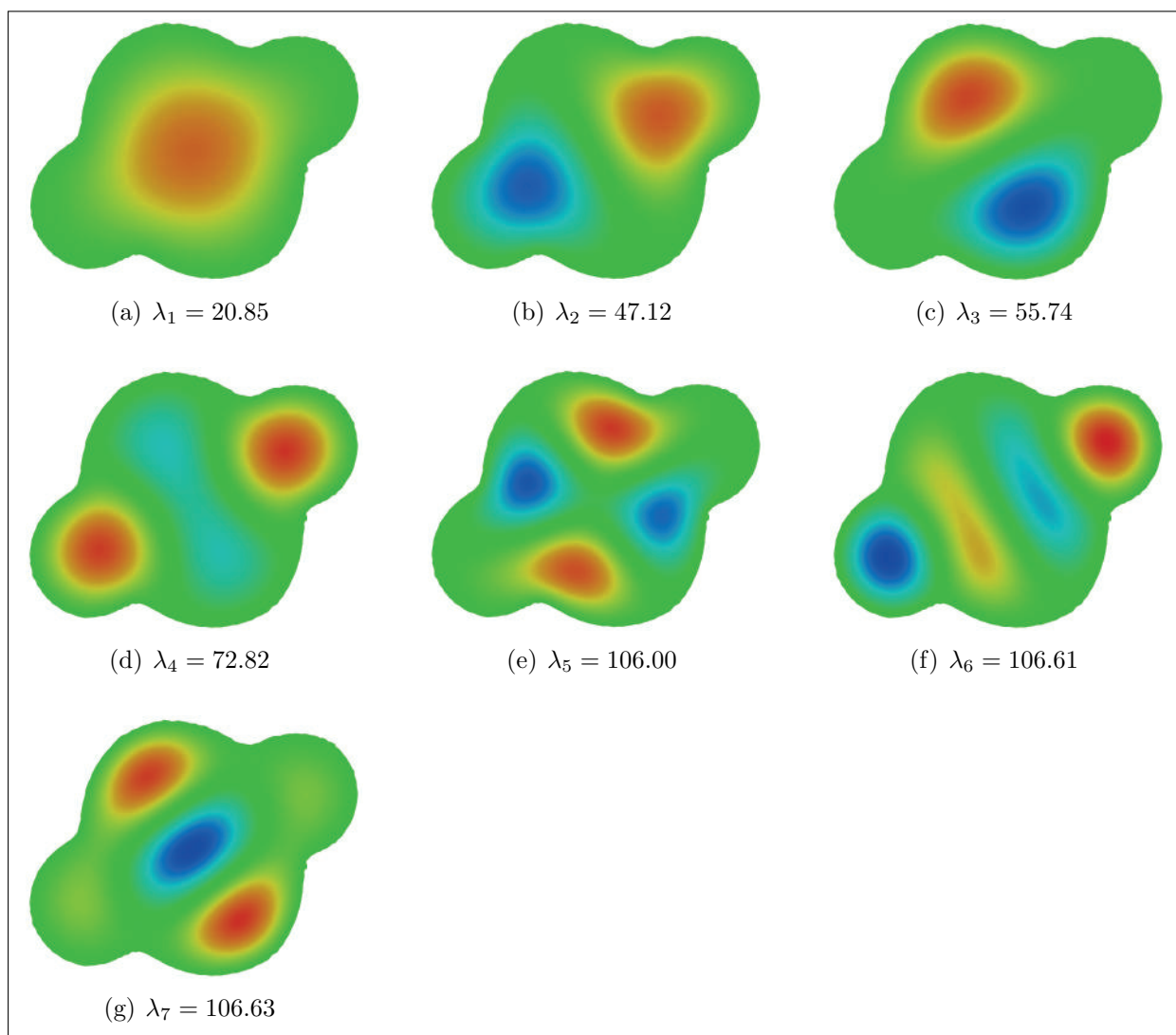


FIGURE 6.33 – Domaine optimal pour $\lambda_7 = 106.63$ (estimation à 106.51 avec mass-lumping).

Huitième valeur propre

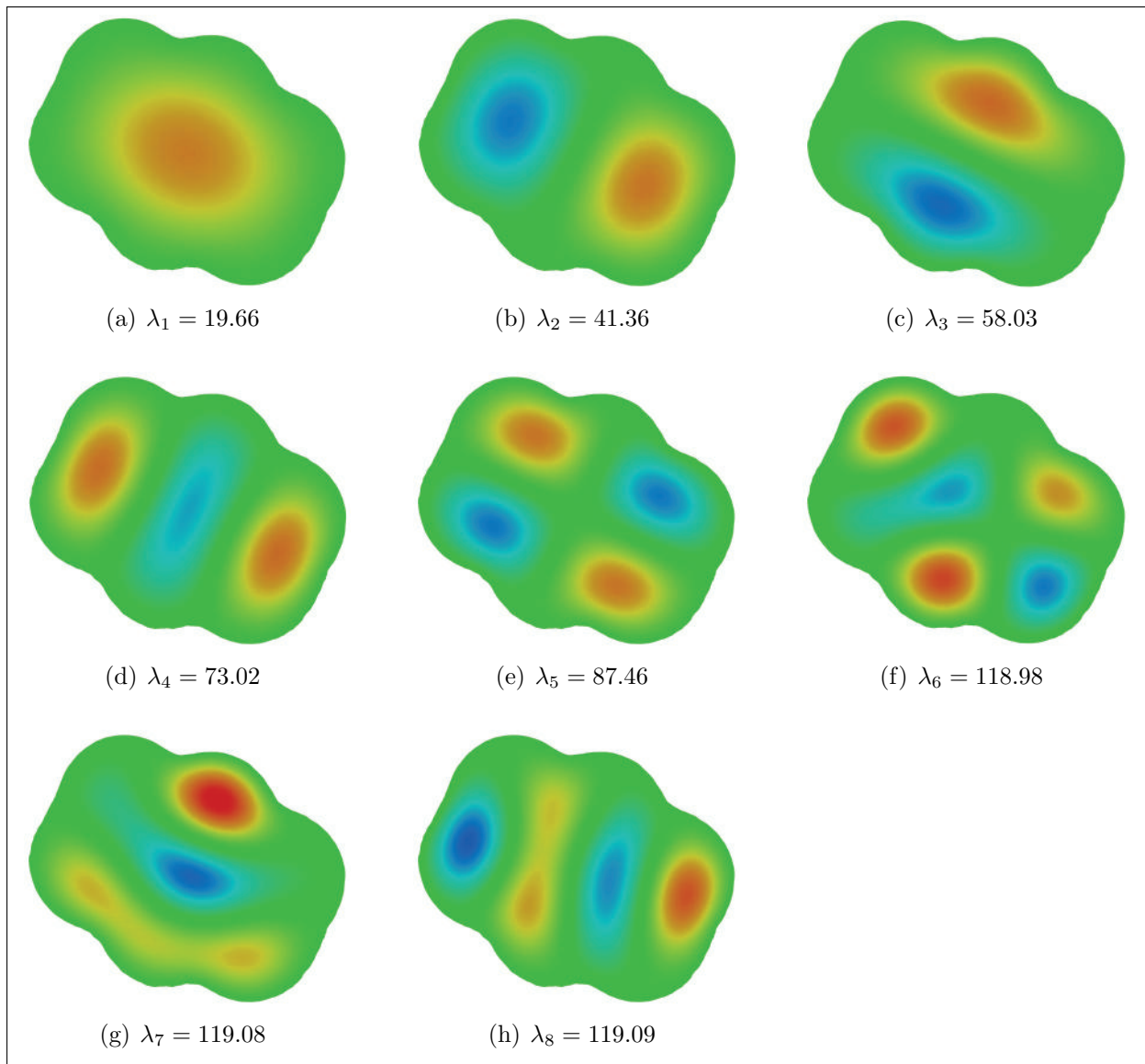


FIGURE 6.34 – Domaine optimal pour $\lambda_8 = 119.09$ (estimation à 118.89 avec mass-lumping).

Neuvième valeur propre

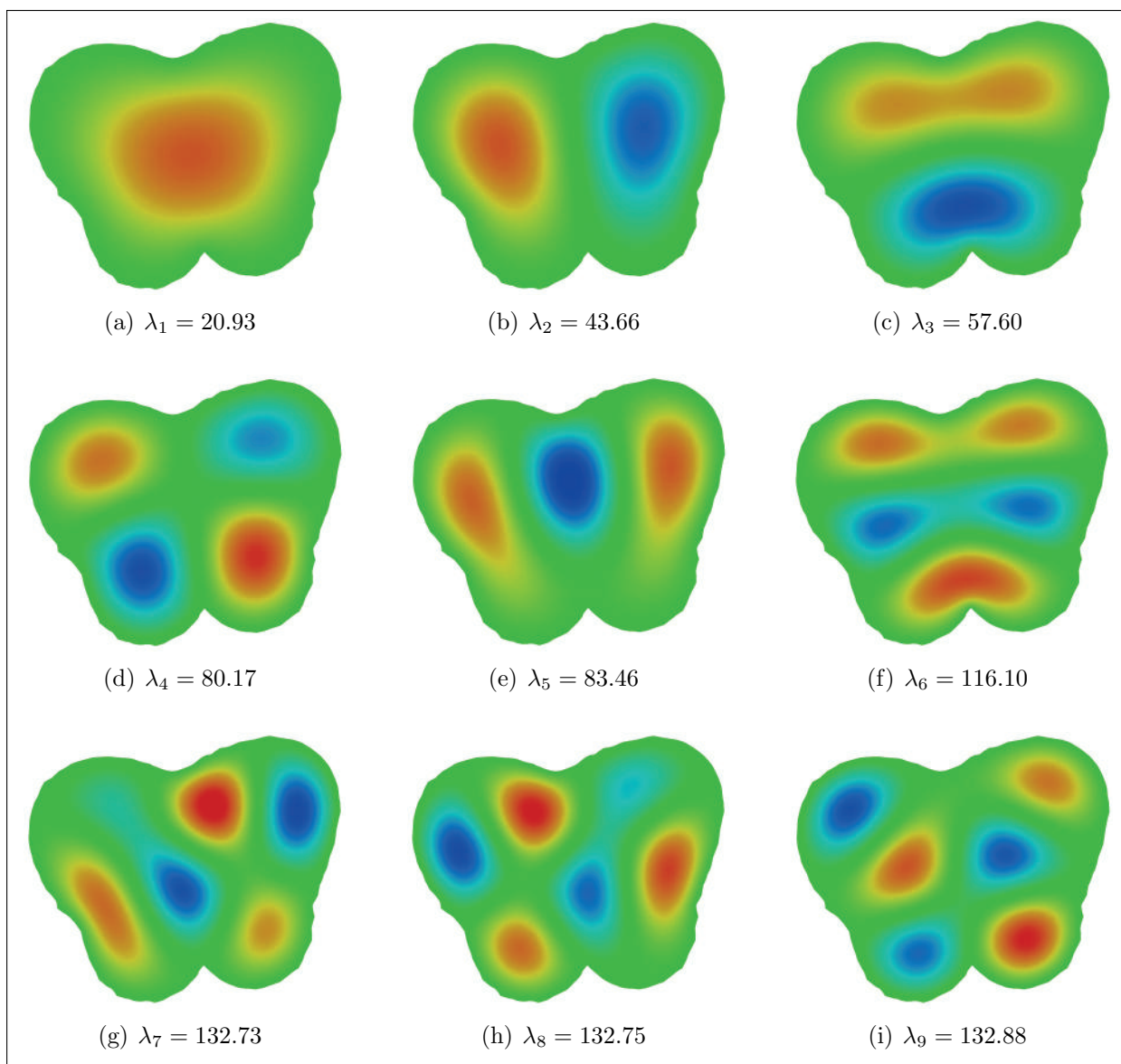


FIGURE 6.35 – Domaine optimal pour $\lambda_9 = 132.88$ (estimation à 132.63 avec mass-lumping).

Dixième valeur propre

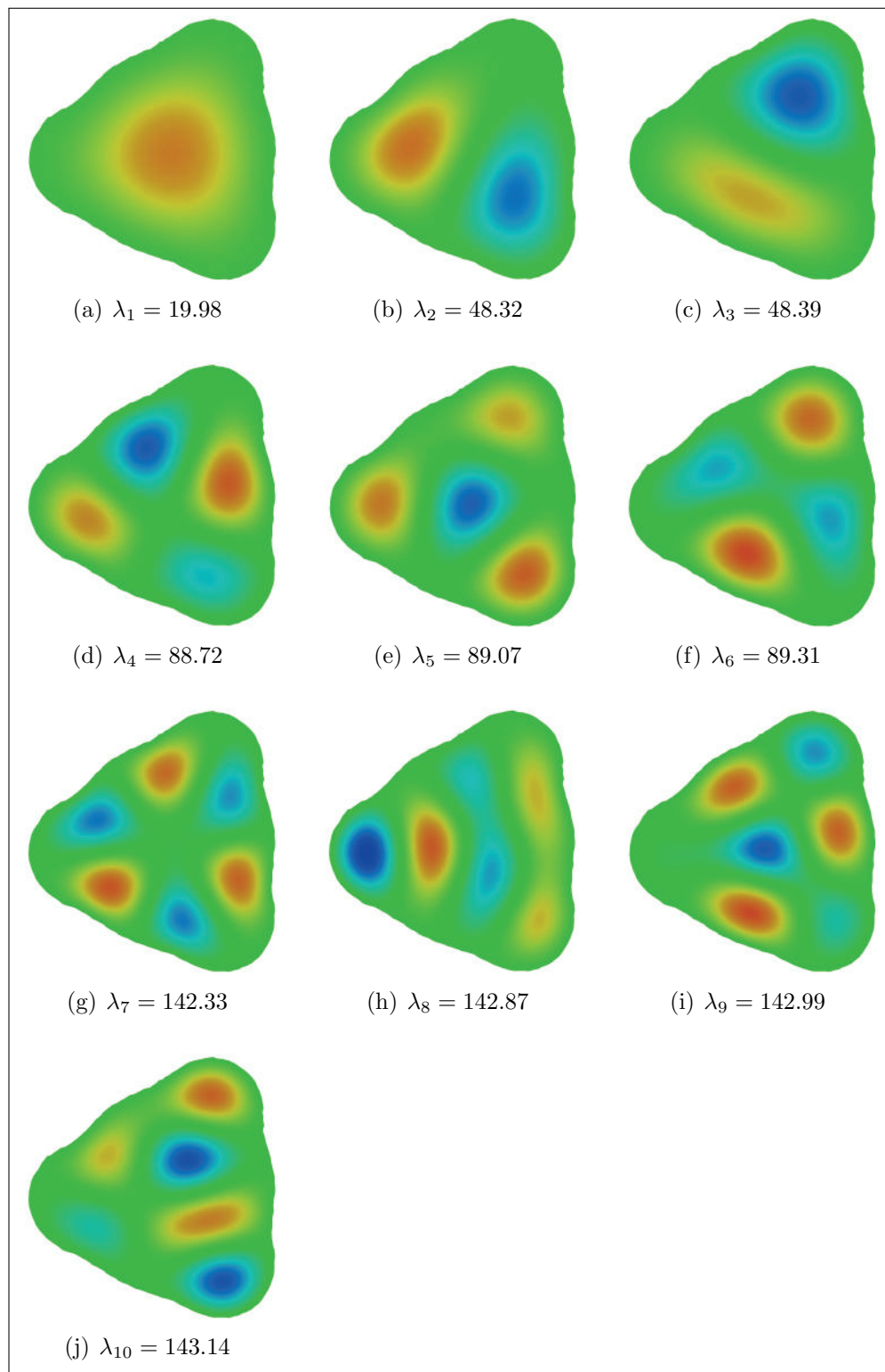


FIGURE 6.36 – Domaine optimal pour $\lambda_{10} = 143.14$ (estimation à 142.89 avec mass-lumping).

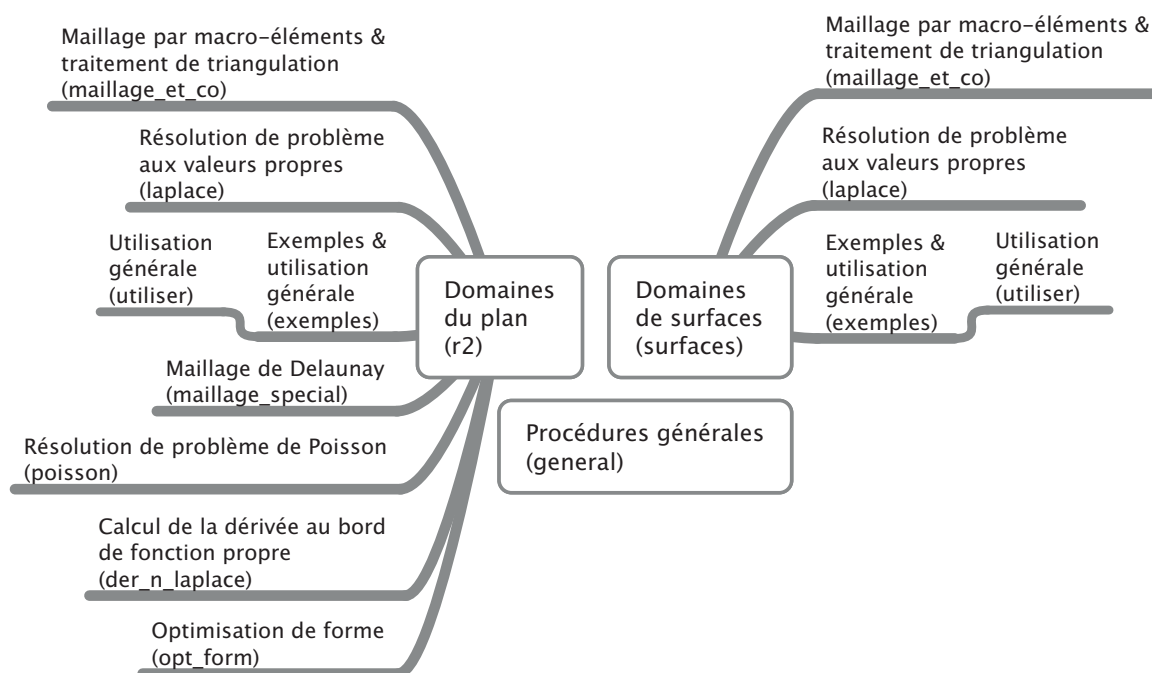
Chapitre 7

Manuel d'utilisation

Les programmes présentés ici ont été écrits pour ce travail en *Fortran 90*. Ils utilisent fréquemment la librairie *LAPACK* pour le calcul matriciel comme des produits et des résolutions d'équations (voir [3] ou [34]). La librairie *ARPACK* est également employée pour la résolution du problème matriciel aux valeurs propres (voir [32] ou [34]). Ces deux librairies doivent donc être préalablement installées pour permettre l'exécution des programmes y faisant appel. De manière générale, la compilation et l'exécution des programmes est simplifiée par la mise à disposition de fichiers *makefile* (notés *mk* puis le nom du programme associé), mais les chemins d'accès doivent être mis à jour pour correspondre à l'arborescence de l'ordinateur employé.

Concernant la visualisation de triangulations et de résultats, il est à noter que les procédures dites de *visualisation* se contentent de convertir les fichiers de données obtenus sous une forme reconnue par le logiciel *AVS/Express* (©*Advanced Visual Systems Inc.*). La visualisation à proprement parler se fait donc via ce logiciel (ou tout autre logiciel compatible), même s'il est tout à fait possible d'en employer un autre; dans ce cas, il s'agira éventuellement d'adapter les routines pour convertir convenablement les fichiers de données. Dans ce travail, le programme *Paraview* (©*Sandia Corporation, Kitware Inc.*) a finalement été préféré à *AVS/Express*. Il s'agit d'un logiciel open-source et multi-plateformes qui a l'avantage d'être particulièrement simple d'accès en plus d'être compatible avec les fichiers de données de type *AVS/Express*.

L'arborescence, schématisée ci-dessous, débute par trois dossiers principaux, nommés *general*, *r2* et *surfaces*. Ils contiennent respectivement des procédures générales, les procédures concernant les domaines du plan et celles concernant les domaines de surfaces.



Le dossier *general* contient quelques routines générales utilisées dans d'autres procédures (notamment les plus récentes) et permettent essentiellement la lecture et l'écriture de données de fichiers types.

Le dossier *r2* contient l'ensemble des programmes concernant les domaines de \mathbb{R}^2 :

- Des procédures de maillage (*maillage_et_co* pour le maillage par macro-éléments et les traitements de triangulation, *maillage_special* pour le maillage de Delaunay).
- Des procédures pour la résolution du problème aux valeurs propres (*laplace*).
- Une série d'exemples de domaines maillés et résolus (*exemples*). Ce dernier dossier contient en particulier, dans le sous-dossier *utiliser*, une routine permettant à l'utilisateur de mailler un domaine (par macro-éléments), de traiter la triangulation obtenue et de déterminer les solutions du problème aux valeurs propres associé sans avoir à transporter les fichiers de données d'un dossier à un autre.
- Une routine de résolution du problème de Poisson (*poisson*), ayant servi de modèle à la résolution du problème aux valeurs propres.
- Des procédures calculant la dérivée normale au bord du domaine d'une fonction propre de Dirichlet donnée (*der_n_laplace*).

- Des procédures d’optimisation de forme (*opt_form*).

Le dossier *surfaces* contient quant à lui les programmes concernant les domaines de surfaces, généralisant ceux spécifiques aux domaines de \mathbb{R}^2 :

- Le mailleur par macro-éléments et les traitements de triangulation (*maillage_et_co*).
- La résolution du problème aux valeurs propres (*laplace*).
- Une série d’exemples (*exemples*), complétée d’une procédure de maillage (par macro-éléments), de traitements de triangulation et de résolution du problème aux valeurs propres combinés (*utiliser*).

Il est à noter que ces procédures, spécifiques aux domaines de surfaces, permettent naturellement la prise en compte de domaines de \mathbb{R}^2 , modulo de légères modifications au niveau des fichiers de données (indication de la métrique triviale par exemple).

De nombreux exemples de fichiers de données (ainsi que quelques procédures de définition, par exemple de domaines ou de métriques) accompagnent les programmes, permettant à l’utilisateur de s’en servir comme modèles. De plus, chaque programme détaille lesquels sont des fichiers nécessaires à l’exécution (à fournir par l’utilisateur, tels les fichiers de définition de domaine) et lesquels sont créés par les procédures, comme fichiers de résultats finaux (comme celui de résultats du problème aux valeurs propres) ou intermédiaires (comme le fichier des nœuds du bord du domaine). À noter que les fichiers *.f90* ont été écrits à l’aide du logiciel *Xcode*, version 2.4.1 (©Apple Inc.). La mise en page est donc optimale lorsque les fichiers sont ouverts avec ce logiciel, alors qu’elle peut différer quelque peu sinon, notamment au niveau de l’alignement.

Chaque fichier de procédures contient de nombreuses explications concernant tant son utilisation que ses mécanismes. De manière générale, chaque fichier débute par un titre et une brève explication de ses fonctions. Les bibliothèques utilisées sont listées, de même que les procédures externes et internes au fichier. Les paramètres et les fichiers de données d’entrée et de sortie des procédures sont également expliqués en amont. Vient ensuite, pour chaque procédure, la déclaration des données, à commencer par les paramètres d’entrée et de sortie, suivie d’une brève explication de chacune d’elles. Vient ensuite la procédure en elle-même, précédant ses éventuelles sous-routines (lorsqu’elles sont spécifiques à la procédure en question).

Un paramètre en particulier est récurrent dans la plupart des procédures et détermine la quantité d’informations retournées à l’utilisateur à mesure de l’avancement des calculs : le paramètre *pdebug*. Selon le niveau d’informations choisi, le terminal affiche la procédure en cours d’exécution, le nom des fichiers créés et/ou la valeur de certains paramètres clefs

(comme le nombre de nœuds créés lors d'un maillage). Le paramètre *pdebug* est à modifier par l'utilisateur, si besoin, au début de chaque routine. Il accepte généralement les valeurs 0 (aucune information donnée), 1 (information partielle contenant le nom des fichiers créés), 2 (information complète, avec affichage des principaux résultats) et -9 (valeur spéciale, à utiliser essentiellement à des fins de debugging avancé, permettant l'affichage de nombreux résultats, telle la liste des nœuds créés). Les différentes valeurs acceptées sont indiquées au début de chaque routine. A noter qu'il est évidemment possible pour l'utilisateur de redéfinir les informations à afficher notamment lorsque *pdebug* vaut -9 .

7.1 Procédures générales

(.../general)

Ces routines utilitaires ont été créées dans le but de remplacer certaines procédures répétées fréquemment dans les autres programmes et effectuées sinon manuellement, telles la lecture et l'écriture de fichiers types (comme les fichiers de nœuds ou d'éléments). Cela permet ainsi de remplacer plusieurs lignes de code par un simple appel de procédure, avec un risque d'erreur réduit. Cela facilite de plus une éventuelle modification future de la structure des fichiers types, en garantissant la compatibilité de tous les programmes modulo une simple mise à jour de quelques procédures générales de lecture et d'écriture de fichiers types.

Le second objectif, lié à ce problème de compatibilité, est l'homogénéisation des fichiers types pour les domaines de \mathbb{R}^2 et de surfaces. En effet, les fichiers tels qu'utilisés ne sont pas compatibles selon qu'ils correspondent à un domaine de \mathbb{R}^2 ou à une surface. Par exemple, on ne peut pas lire un fichier de définition d'un domaine de \mathbb{R}^2 avec les procédures spécifiques aux surfaces. Il est également souhaitable d'autoriser les commentaires (notamment en début de fichier), ce qui n'est pas systématiquement le cas avec les structures originales des fichiers types utilisés.

Des procédures générales ont ainsi été créées et regroupées ici. De plus, afin de répondre au deuxième objectif fixé, une modification de la structure de certains types de fichier a été effectuée. Toutefois, ces procédures générales ayant été introduites tardivement dans le développement des programmes, elles ne sont utilisées que ponctuellement et la structure originale des fichiers a été conservée. Aussi, quelques procédures *provisoires* ont été créées afin de gérer les fichiers types d'origine. Dans une perspective de développement des programmes élaborés dans le cadre de ce travail, il paraît toutefois clair qu'une utilisation généralisée des nouvelles procédures et structures de fichier serait préférable.

La structure des fichiers des données est détaillée avant chaque procédure y faisant appel, accompagnée de remarques utiles sur les données associées et leur utilisation. La routine *test_general.f90* a pour but de tester les procédures générales, données dans le fichier *general.f90*. Les procédures générales concernent :

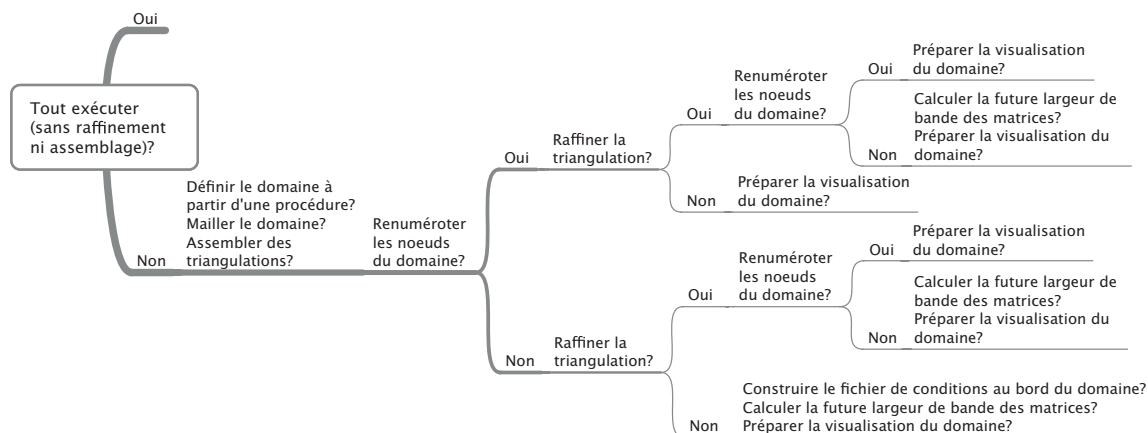
- Le test de paramètre de longueur de tableau (qui doit être positif, afin d'éviter des problèmes de débordement par exemple).
- L'effacement de fichier (éventuellement avec écriture d'une ligne de texte, pouvant par exemple tenir le rôle de fichier de données trivial avec un zéro pour unique contenu).
- La lecture des commentaires en début de fichier.
- La lecture, l'écriture et la copie des fichiers de domaine (indépendamment pour \mathbb{R}^2 et pour les surfaces), de nœuds (idem), d'éléments, de classes d'équivalence des nœuds, de conditions au bord, de nœuds sur face courbe, de bord, de domaine de raffinement, de paramètres de problème aux valeurs propres, de fichiers pour la visualisation et de résultats de problème aux valeurs propres (indépendamment pour \mathbb{R}^2 et pour les surfaces). Des procédures *provisoires* concernant les fichiers de domaine, de classes d'équivalence des nœuds et de résultats de problème aux valeurs propres sont également proposées.

7.2 Maillage par macro-éléments et traitements de triangulation

(.../r2/maillage_et_co et .../surfaces/maillage_et_co)

Les routines de maillage par macro-éléments, complétées de routines d'assemblage de triangulations, de renumérotation des nœuds, de raffinement de triangulation et de préparation à la visualisation, ont été rassemblées dans le fichier *compact.f90*. Le fichier *use_compact.f90* consiste en un programme d'utilisation des procédures de *compact.f90*, avec une interface utilisateur complète. Elle permet l'exécution des différentes étapes de maillage une à une ou en différents blocs, selon le choix de l'utilisateur. Cette décision est transmise au programme soit via les données prédéfinies de la procédure *optexec2* du fichier *optexec.f90* (si le paramètre *typeexec* de la procédure *optexec1* vaut 0), soit de manière interactive au fur et à mesure de l'avancée du programme (si *typeexec* vaut 1).

Les différentes possibilités offertes à l'utilisateur sont schématisées ci-dessous. Il s'agit essentiellement de choisir si un fichier de domaine doit être créé à partir de procédures de définition de domaine (*defdom1* et *defdom2* de *defdom.f90*), si le maillage à proprement parler, un assemblage de triangulations, une renumérotation, un raffinement et un calcul de la largeur de bande des futures matrices doivent être effectués et si un fichier de visualisation doit être créé.



Un fichier *domraff.f90* contenant la routine *domraff* définissant le domaine à raffiner doit également être joint, de même que, dans le cas d'un domaine de surface, un fichier *fctcolle.f90* contenant la routine *fctcolle* définissant les fonctions de recollement des différentes cartes. Finalement, pour visualiser la triangulation directement sur une surface de \mathbb{R}^3 (et non via des paramétrisations locales), un fichier *param.f90* contenant une routine *param* définissant l'image dans \mathbb{R}^3 des points des différentes paramétrisations doit être donné.

7.2.1 Définition du domaine

Un domaine à mailler par macro-éléments peut être donné de deux manières différentes : par des procédures (voir *defdom.f90*) ou par un fichier de données. La manière procédurale est avantageuse dans le cas de domaines symétriques ou, au contraire, relativement complexes. C'est notamment le cas lorsque certaines faces sont courbes, puisqu'il suffit de donner un algorithme de calcul des nœuds sur face courbe (alors que le fichier de données correspondant doit comprendre tous les nœuds sur face courbe).

La procédure *dom* permet le passage d'une définition procédurale du domaine au fichier de données correspondant (les procédures de maillage développées n'acceptant une définition du domaine que via un fichier de données). A noter que trois procédures de définition du domaine sont nécessaires pour la routine *dom* :

- La première, *defdom1*, détermine le nombre de macro-éléments du domaine (permettant ainsi au programme d'allouer la mémoire nécessaire aux autres données).
- La deuxième, *defdom2*, retourne les types de macro-éléments en présence, les coordonnées des sommets, les conditions au bord, le nombre de mailles et, dans le cas de macro-éléments à face courbe, le domaine de définition des courbes. Dans le cas de domaines de surfaces, la procédure retourne également les fonctions d'identification des faces des

macro-éléments, le nombre de cartes utilisées ainsi que, pour chaque macro-élément, la carte correspondante.

- Les courbes en elles-mêmes sont données par la dernière procédure de définition du domaine, nommée *fgamma* et opérant comme une fonction retournant les coordonnées d'un point d'une face courbe (à partir de la valeur de la variable associée).

7.2.2 Maillage par macro-éléments

La procédure de maillage à proprement parler, nommée *maill*, est notamment constituée de sous-procédures de lecture du fichier de domaine (*lec*), de création des nœuds (*noeud*), des éléments (*elem*) et des classes d'équivalence (*class*) lorsqu'il s'agit d'un domaine de surface. La procédure nécessite un paramètre de *résolution* du maillage *res* : deux points dont la distance est plus petite que *res* seront considérés égaux. Il s'agit donc de choisir ce paramètre raisonnablement pour ne pas risquer de créer de nœuds dédoublés, ni, à l'opposé, d'identifier des nœuds réellement distincts !

A noter que les routines *lec* et *noeud* pour les domaines du plan acceptent des pseudo macro-éléments de type *segments* (éventuellement courbes), afin de pouvoir être réutilisées par les procédures de maillage de Delaunay (qui ne nécessitent que la définition du bord du domaine). Remarquons pour finir que la procédure engendre un fichier de donnée contenant la liste des nœuds sur face courbe et les valeurs des variables associées. Bien que ces données ne soient pas nécessaires au maillage du domaine en soi, elles pourraient être utiles lors d'un éventuel futur raffinement de la triangulation.

7.2.3 Fichier de conditions au bord

La routine *bcond* se charge de lister dans un fichier les nœuds soumis à des conditions au bord de Dirichlet et la valeur de ces dernières. Ce fichier est dispensable pour le problème aux valeurs propres, puisque les conditions considérées sont homogènes. Le fichier de conditions au bord est en fait un vestige du problème ayant servi de modèle pour la résolution du problème aux valeurs propres, à savoir le problème de Poisson. Pour ce dernier, les conditions au bord étaient données par la fonction *ftbord* et lues par la procédure *bcond* pour la création d'un fichier de données contenant les nœuds du bord et les valeurs des conditions au bord associées.

Notons qu'une routine semblable à *bcond* se trouve dans la procédure de renumérotation *renum* (*cond*).

7.2.4 Assemblage de triangulations

La routine *assemblage* permet d'assembler les triangulations de deux parties d'intérieurs disjoints d'un domaine. Il peut en effet s'avérer utile d'assembler plusieurs triangulations

après les avoir traitées indépendamment les unes des autres, par exemple dans le cas de parties maillées plus finement que d'autres ou lorsqu'on souhaite que la numérotation des nœuds dépende de la partie du domaine dans lesquels ils se trouvent. Il peut également être avantageux de mailler une partie du domaine seulement, puis de définir le reste de la triangulation comme image de la première, par exemple par symétrie, avant d'assembler les différentes parties obtenues.

Préalablement à l'assemblage de deux triangulations, l'utilisateur doit s'assurer qu'elles soient compatibles, c'est-à-dire que la triangulation obtenue par identification des nœuds communs (i.e. ayant les mêmes coordonnées et situés dans la même carte) soit effectivement une triangulation, au sens de la section 2.3.3. L'assemblage de triangulations multiples se fait itérativement, en assemblant progressivement chaque sous-triangulations deux à deux.

A noter que les conditions au bord sont mises à jour par la procédure d'assemblage. En particulier, les conditions sur des parties se retrouvant à l'intérieur de la triangulation obtenue après assemblage sont ignorées.

7.2.5 Renumerotation

La procédure *renum* renumérote les nœuds dans le but de réduire la future largeur de bande des matrices. L'algorithme de renumérotation débute par le nœud donné par le paramètre *pprem* s'il est cohérent ; dans le cas contraire, l'algorithme démarre avec le premier nœud du bord s'il existe, le premier du domaine sinon. La numérotation la plus avantageuse est sauvée à la place des fichiers originaux et l'autre dans des fichiers de copie (si les fichiers proposés par l'utilisateur sont différents). Dans le cas où les largeurs de bande pour les deux numérotations sont égales, le paramètre *pbest* détermine laquelle des numérotations est retenue.

Parmi les sous-procédures de *renum*, notons *calcullda* qui permet le calcul des futures largeurs de bande des matrices et qui peut être utilisée indépendamment de la procédure de renumérotation. De plus, la routine *cond* est tirée de la procédure *bcond* d'écriture des nœuds du bord sous condition de Dirichlet.

7.2.6 Raffinement

La procédure de raffinement *raff* permet de raffiner la triangulation d'un domaine soit globalement, soit localement *autour* d'un domaine de raffinement. Ce sous-domaine peut être donné de plusieurs manières différentes, en fonction de la valeur du paramètre *modedom* : soit directement dans un fichier de données contenant les nœuds du domaine de raffinement, soit via une procédure déterminant ces nœuds. Dans le cas de raffinements successifs, on propose également à l'utilisateur de donner le domaine via une procédure lors de la première itération, puis d'utiliser un fichier de données mis à jour à chaque étape en fonction du raffinement et du domaine de raffinement précédent lors des itérations sui-

vantes. Cette dernière option a l'avantage d'éviter une réduction malheureuse de la rondeur des triangles lors de raffinements successifs ; en contrepartie, le domaine de raffinement obtenu peut différer légèrement du domaine théorique, qui sera reproduit plus fidèlement via une procédure de définition à chaque itérations.

La sélection des nœuds du domaine de raffinement est effectuée dans le fichier *domraff.f90*, via la procédure *domraff*. Selon la valeur du paramètre *modedom* et l'itération de raffinement en cours, cette procédure renvoie soit directement à la lecture d'un fichier de données contenant les nœuds du domaine de raffinement, soit à la procédure *dsdom* de *domraff.f90*, fournie par l'utilisateur et qui doit indiquer, pour tout nœud donné, s'il appartient ou non au domaine de raffinement.

Le paramètre *es* détermine si la nouvelle triangulation est sauvée dans les fichiers initiaux (la triangulation initiale étant alors sauvée dans des fichiers de copies) ou dans de nouveaux fichiers. La procédure *raff* laisse le choix à l'utilisateur, via le paramètre *courb*, de prendre en compte la courbure du domaine (i.e. les courbes éventuelles définissant les macro-éléments) afin d'épouser parfaitement le bord du domaine théorique initial. Dans le cas où l'utilisateur ne souhaite pas en tenir compte, deux options s'offrent encore à lui (toujours via *courb*) quant à la mise à jour du fichier contenant les nœuds sur face courbe : soit il peut *effacer* le fichier (i.e. remplacer son contenu par un 0), soit le programme met à jour le fichier en supposant que le raffinement n'a pas créé de nouveau nœud sur une face courbe. Cette dernière option ne peut donc être employée que dans des situations bien particulières, lorsque l'utilisateur est sûr qu'il ne va pas corrompre ces données. A noter que ce fichier n'est utilisé que pour le raffinement de triangulation (sachant toutefois que le maillage d'un domaine par la méthode de Delaunay peut nécessiter un raffinement).

Les différentes étapes du raffinement d'un domaine telles que présentées à la section 3.2.2 constituent des sous-procédures de *raff*. Il s'agit de la détermination des éléments à couper en 4 (*defel4*), des nœuds adhérant à ces éléments (*defnoadhd*) et des éléments à casser en 2 (*defel2*), puis de la découpe des éléments sélectionnés (*casel4* et *casel2*).

7.2.7 Fichier de visualisation

La routine *visu* compile les données de plusieurs fichiers concernant la triangulation et (en option) les résultats du problème aux valeurs propres en un fichier de données utilisable par le logiciel *AVS/Express* (ou compatibles, comme *Paraview*). Le nombre de fonctions propres à illustrer est indiqué par le paramètre *nbfcts*; les fonctions à illustrer le sont via *listfcts*, contenant les numéros desdites fonctions. Finalement, deux paramètres déterminent le mode de visualisation des domaines (un seul pour les domaines de \mathbb{R}^2). Le paramètre *modev2d* détermine le mode de visualisation désiré pour les domaines plans (et les cartes), i.e. l'espace dans lequel on les plonge, à savoir dans \mathbb{R}^2 ou dans \mathbb{R}^3 . Le paramètre *modev* détermine quant à lui ce qui doit être visualisé dans le cas de domaines de surfaces, à savoir le domaine de la surface lui-même ou les domaines correspondants dans l'atlas.

7.3 Maillage de Delaunay

(.../r2/maillage_special)

La procédure de maillage de Delaunay présentée ici concerne uniquement des domaines de \mathbb{R}^2 de conditions au bord homogènes (éventuellement mixtes). Bien qu'absent ici, un tel maillage pour des domaines de surface est toutefois essentiellement semblable, puisqu'il s'agit de mailler les sous-domaines correspondant aux différentes paramétrisations locales des coques partitionnant le domaine.

La procédure principale se trouve dans le fichier *use_maillspe.f90* et fait appel aux routines spécifiques au maillage de Delaunay du fichier *maillagespe.f90*, en plus de certaines procédures de maillage par macro-éléments et de préparation à la visualisation de *compact.f90* (*dom*, *lec*, *noeud*, *findnn*, *bcond*, *visu*). Les routines de *defdom.f90* (définition du domaine) et de *domraff.f90* (inutilisées mais indispensables à l'exécution de *compact.f90*), déjà utilisées pour le maillage par macro-éléments, sont également demandées par la routine principale.

La structure de la procédure de maillage de Delaunay suit essentiellement la chronologie de l'algorithme de maillage de Delaunay présentée à la section 3.3 et débute par la définition du domaine à mailler. Comme pour le maillage par macro-éléments, il est donné de manière procédurale via le fichier *defdom.f90*. Une définition via un fichier de donnée est également envisageable mais n'a été retenu qu'en tant que moyen de tester des domaines simples (bord homéomorphe à \mathbb{S}^1), la solution donnée par la routine *bordauto* (bord du domaine donné par la liste de ses nœuds successifs) se révélant peu satisfaisante en général.

Les procédures de définition du domaine à mailler de *defdom.f90* conservent la structure utilisée pour le maillage par macro-éléments, permettant la définition de macro-éléments (de types triangulaires et rectangulaires, éventuellement courbes) et de pseudo macro-éléments linéaires (éventuellement courbes). Ainsi, l'utilisateur a le choix d'utiliser des macro-éléments lorsqu'il souhaite définir lui-même la structure interne de la triangulation et des pseudo macro-éléments définissant le bord de son domaine lorsque la structure interne lui importe peu ou qu'il ne peut la donner. La routine *detmaill*, déterminant la méthode de maillage à employer, à savoir par macro-éléments ou par Delaunay, précède d'ailleurs le maillage à proprement parler. La méthode dépend en effet de la manière dont le domaine à mailler est donné : s'il est donné uniquement par des macro-éléments, il serait inutile de mailler le domaine avec une méthode de Delaunay, au contraire d'un domaine donné par les segments constituant son bord qui ne peut être traité qu'avec une telle méthode. A noter toutefois que, ici, la méthode de maillage est forcée pour adopter l'approche de Delaunay. Il va toutefois de soi qu'un développement naturel de ces procédures de maillage passerait par une prise en compte des deux possibilités de maillage de domaine, via une fusion des procédures de *compact.f90* et de *maillspe.f90*.

La routine *constrdata* est chargée de créer les fichiers de données à partir du fichier de définition du domaine. Il s'inspire en partie de la routine *maill* de *compact.f90* et fait d'ailleurs appel à la sous-routine *noeud*. Il emploie sa propre sous-routine *cbord* pour l'écriture du bord du domaine, avant d'effectuer un test de bonne définition de ce dernier via *testbord*. Cette procédure vérifie que toutes les composantes connexes du bord déterminent des courbes fermées.

Le maillage de Delaunay à proprement parler commence avec la routine *compdom*, chargée de déterminer les nœuds internes au domaine. Cela se fait en deux temps, comme décrit à la section 3.3, à savoir l'ajout des nœuds d'une petite grille recouvrant le domaine puis la sélection des nœuds se trouvant à l'intérieur du domaine (suffisamment loin des nœuds du bord du domaine). La partie cruciale de la routine est la détermination de la taille des mailles et la largeur de la frontière du domaine, c'est-à-dire la bande le long du bord du domaine à l'intérieur de laquelle aucun nouveau nœud ne sera inséré. J'ai choisi pour le premier paramètre le maximum des longueurs des segments du bord du domaine (d'où l'importance d'avoir des segments presque isométriques). Quant au second paramètre, il s'agit d'une fraction du premier ; j'ai par exemple utilisé le rapport 0.8.

La routine *boite* crée ensuite la boîte nécessaire au maillage de Delaunay ainsi que sa triangulation. Les coordonnées des nœuds définissant la boîte sont déterminés en fonction de la valeur maximale b des valeurs absolues des coordonnées des nœuds du domaine : il s'agit de $(-b, -b)$, $(b, -b)$, (b, b) et $(-b, b)$. Pour faciliter la suppression future de la boîte, le paramètre b est renvoyé en sortie de routine.

Vient ensuite l'insertion un à un des nœuds du domaine à la triangulation, effectuée grâce à la routine *maillboite*. Le maillage obtenu est appelé maillage-boîte : les nœuds du bord du domaine ainsi que les nœuds internes appartiennent à la triangulation, mais le bord du domaine n'est pas forcément respecté. De plus, quatre nœuds supplémentaires appartiennent à la triangulation : les sommets de la boîte.

Il s'agit à présent de tester le respect du bord du domaine et de modifier le maillage-boîte en conséquence, de supprimer les nœuds définissant la boîte ainsi que les éléments superflus. Ces actions sont effectuées successivement par les routines *correctbord*, *suppboite* et *suppept*. Il est à noter au sujet de la routine *correctbord* que des raffinements de triangulation peuvent être nécessaires pour corriger une triangulation ne respectant pas le bord du domaine (insertion de nouveaux nœuds sur des segments du bord). Ainsi, au même titre que lors d'un raffinement de triangulation classique tel qu'effectué par la routine *raff*, il s'agit pour l'utilisateur de décider de la prise en compte du paramètre *courb* pour que le raffinement épouse au mieux le bord théorique du domaine. Hors expérimentation, le paramètre *opt* doit être réglé sur la valeur 1 ; en sortie de routine, il indique le nombre d'itérations de correction du bord nécessaire. Finalement, le paramètre *eps*, représentant une valeur considérée nulle et donné comme paramètre d'entrée à la routine *correctbord* par *use_maillspe.f90*, doit être calibré soigneusement.

Dernière étape du maillage de Delaunay, l'écriture des conditions au bord sur les éléments ne se fait pas simultanément au maillage à proprement parler, mais en fin d'algorithme via la routine *bcondel* par comparaison avec les conditions au bord données initialement par le fichier de définition du domaine et reportées dans les fichiers de données concernant le bord du domaine.

7.4 Résolution du problème de Poisson

(.../r2/poisson)

Une routine de résolution du problème de Poisson est brièvement présentée ici, bien que ce problème n'ait pas bénéficié de description théorique dans ce travail. Sa modélisation est très proche de celle du problème aux valeurs propres (et d'ailleurs plus simple).

L'utilisateur de la routine du fichier *poisson.f90* est invité à fournir une triangulation du domaine, c'est-à-dire des fichiers de nœuds, d'éléments et de conditions au bord. De plus, la fonction *source* (du fichier *poisson.f90*) détermine la valeur du second membre de l'équation en un point donné; il incombe donc à l'utilisateur de définir cette fonction pour qu'elle corresponde au problème de Poisson considéré. Les résultats du programme sont sauvés dans le fichier *pot.dat*, qui liste les coordonnées des nœuds de la triangulation et les valeurs de la solution en chacun de ces points.

7.5 Résolution du problème aux valeurs propres

(.../r2/laplace et .../surfaces/laplace)

Les procédures de résolution du problème aux valeurs propres sont principalement rassemblées dans le fichier *laplace.f90*, constitué de la routine *laplace* et de ses sous-routines. Une procédure supplémentaire, nommée *t3shap*, est également requise et se trouve dans le fichier *fekform.f*. Ce fichier rassemble une collection de procédures retournant différentes données sur les fonctions formes (comme les valeurs de ces fonctions ou de leurs dérivées en un point) pour différents types d'éléments finis. Finalement, un fichier *metrique.f90* contenant la routine *metr* et retournant diverses données sur la métrique de la surface (inverse de la métrique et racine de son déterminant en un point) est nécessaire à l'exécution de *laplace*. À noter que, alors que cette routine est obligatoire pour les surfaces, elle est optionnelle pour les domaines de \mathbb{R}^2 (voir le paramètre *metrique* de la sous-routine *ematr*). Aussi, la procédure pour les domaines du plan permet en réalité de résoudre des problèmes aux valeurs propres sur des domaines de coque, c'est-à-dire sur des domaines de surface paramétrés via une seule carte.

Le fichier *use_laplace.f90* permet l'utilisation de *laplace*, par appel à la procédure après définition de ses paramètres d'entrée :

- Les noms des fichiers contenant la triangulation du domaine (i.e. les fichiers de nœuds, d'éléments et de conditions au bord, ainsi que ceux des classes d'équivalence dans le cas d'un domaine de surface).
- Le nom du fichier contenant les paramètres concernant le problème matriciel aux valeurs propres et l'utilisation de la librairie *ARPACK* (essentiellement le nombre de valeurs recherchées et le nombre maximal d'itérations).
- Le nom du fichier de sauvegarde des résultats de la procédure.
- Le paramètre *masslumping*, définissant si le procédé de mass-lumping sera utilisé pour la construction des matrices du problème.

La structure de la routine *laplace* est résumée ci-dessous ; elle sera détaillée plus bas.

1. Lecture et définition des paramètres, initialisations.
2. Détermination de la correspondance entre nœuds et équations (routine *veccli*).
3. Détermination des largeurs de bandes des matrices (routine *strmat*).
4. Construction des matrices (routine *rigidmass*).
5. Décomposition de Cholesky de la matrice de masse (routine *choles*).
6. Résolution du problème matriciel aux valeurs propres (routine *solve*), à savoir :
 - (a) Algorithme itératif de Lanczos (routine *resovvp*), constitué essentiellement du *reverse communication loop* (routine *dsband1*) et du calcul de la norme résiduelle pour le critère d'arrêt.
 - (b) Ecriture des solutions.
7. Normalisation des solutions.
8. Sauvegarde des résultats.

Après définition des paramètres liés à l'utilisation de la librairie *ARPACK* et lecture des données du problème (domaine et paramètres du problème aux valeurs propres), la routine *laplace* détermine, via la routine *veccli*, la correspondance entre nœuds du domaine et équations linéaires générées. Autrement dit, elle détermine les nœuds du domaine engendrant, via les fonctions formes correspondantes, des équations linéaires et les numérote. La routine *strmat* détermine ensuite la structure des futures matrices du problème en calculant leur largeur de bande, avant que les matrices soient construites par intégration numérique, via la routine *rigidmass*. Il est à noter à ce propos que la sous-routine *ematr* de construction des matrices élémentaires appelle la routine *t3shap* du fichier *fekform.f* ainsi que, dans le

cas d'un domaine de surface, la routine *metr* du fichier *metrique.f90*.

Le problème matriciel aux valeurs propres généralisé est ainsi entièrement modélisé. Toutefois, la routine *laplace* n'emploie pas directement la librairie *ARPACK* pour résoudre le problème généralisé, mais transite par un problème matriciel aux valeurs propre usuel, comme détaillé à la section 3.5, en utilisant notamment une décomposition de Cholesky de la matrice de masse (via la routine *choles*, qui consiste essentiellement en un appel de la routine *dpbtrf* de la librairie *LAPACK*). La routine *solve* délègue ensuite la résolution du problème aux valeurs propres usuel à proprement parler à la sous-routine *resopvp*, puis transforme successivement les solutions obtenues en solutions du problème généralisé (grâce à la routine *dtbtrs* de la librairie *LAPACK*), puis en solutions du problème initial, en tenant compte de la numérotation initiale de tous les nœuds du domaine.

La routine *resopvp* utilise *ARPACK* et sa structure est inspirée des exemples fournis par les concepteurs de la librairie. Il en est de même de la sous-routine *dsband1* (version modifiée de la routine *ARPACK* nommée *dsband*), chargée spécifiquement du *reverse communication loop* (pour plus de détails sur les routines *ARPACK*, voir [32]). Ces deux routines emploient en particulier la procédure *prodsym* de multiplication d'un vecteur par la matrice du problème. En plus d'utiliser *dsband1*, la routine *resopvp* calcule une norme résiduelle, qui est utilisée comme critère d'arrêt par *dsband1*.

Une fois la résolution terminée, *laplace* effectue encore une normalisation des solutions via *normsol* (norme L^2 ou H^1 sur le domaine) avant de sauvegarder les résultats complets de la procédure dans un fichier dédié. Notons finalement que la routine *normsol* appelle la routine *t3shap* du fichier *fekform.f* ainsi que, dans le cas d'un domaine de surface, la routine *metr* du fichier *metrique.f90*.

7.6 Procédures d'utilisation générale

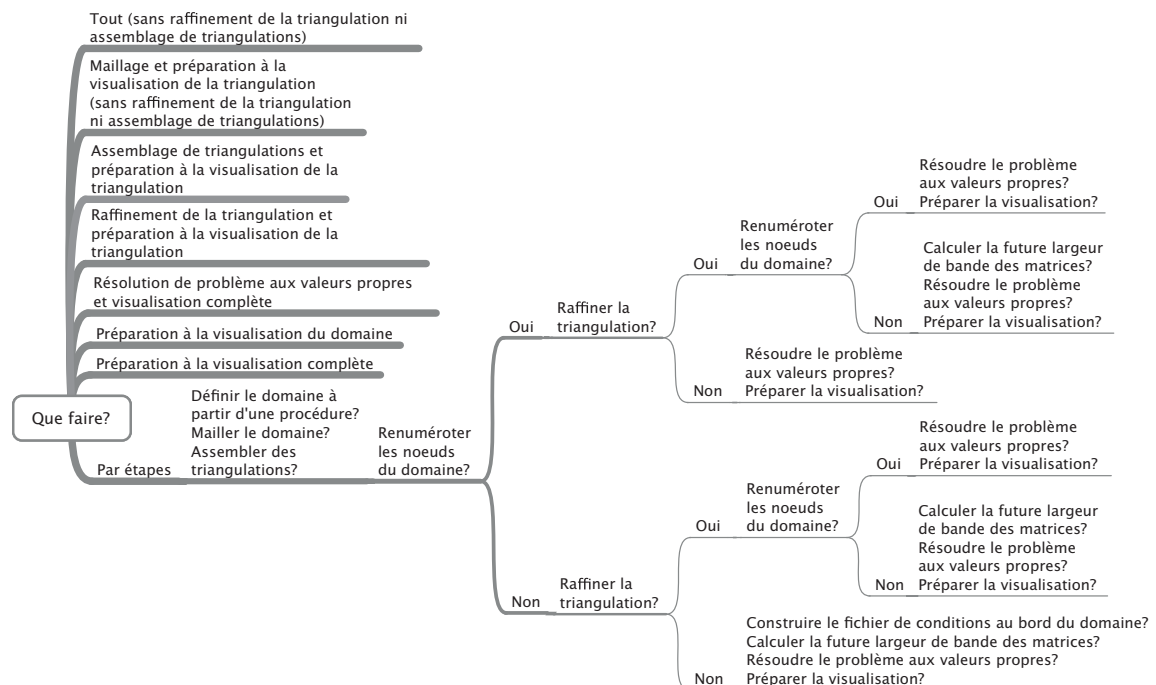
(.../r2/exemples/utiliser et .../surfaces/exemples/utiliser)

Une routine a été créée spécialement pour l'utilisation des procédures de maillage par macro-éléments (complétées des routines d'assemblage de triangulations, de renumérotation des nœuds, de raffinement de triangulation et de préparation à la visualisation) et/ou de résolution du problème aux valeurs propres, permettant à l'utilisateur de créer des exemples complets depuis un seul endroit et à partir des fichiers types de son choix. Il s'agit en quelque sorte d'une généralisation de la procédure de *use_compact.f90* d'utilisation de *compact.f90*, permettant également l'utilisation de *laplace.f90*.

Le fichier principal, intitulé *main.f90*, consiste en un programme d'utilisation des procédures de *compact.f90* et de *laplace.f90*, avec une interface utilisateur complète. Elle permet l'exécution des différentes étapes du traitement d'un exemple indépendamment les unes

des autres ou en différents blocs, selon le choix de l'utilisateur. En particulier, il est possible de mailler un domaine sans résoudre le problème aux valeurs propres associé ou, au contraire, d'effectuer la résolution à partir d'un domaine déjà maillé. Cette décision est transmise au programme soit via les données prédéfinies de la procédure *optexec2* du fichier *optexec.f90* (si le paramètre *typeexec* de la procédure *optexec1* vaut 0), soit de manière interactive au fur et à mesure de l'avancée du programme (si *typeexec* vaut 1).

Les différentes possibilités offertes à l'utilisateur sont schématisées ci-dessous. Il s'agit essentiellement de choisir si un fichier de domaine doit être créé à partir de procédures de définition de domaine (*defdom1* et *defdom2* de *defdom.f90*), si le maillage à proprement parler, un assemblage de triangulations, une renumérotation, un raffinement, le calcul des largeurs de bande des futures matrices et une résolution du problème aux valeurs propres doivent être effectués et si un fichier de visualisation doit être créé.



Outre un fichier contenant les paramètres concernant le problème matriciel aux valeurs propres et l'utilisation de la librairie *ARPACK*, un fichier *domraff.f90* contenant la routine *domraff* et définissant le domaine à raffiner doit également être joint. Il en est de même, dans le cas d'un domaine de surface, des fichiers *fctcolle.f90* et *metrique.f90*, contenant respectivement les routines *fctcolle* et *metr*, définissant les fonctions de recollement des différentes cartes, respectivement diverses données sur la métrique de la surface. Finalement, pour visualiser la triangulation directement sur une surface de \mathbb{R}^3 (et non via des paramétrisations locales), un fichier *param.f90* contenant une routine *param* définissant l'image dans \mathbb{R}^3 des points des différentes paramétrisations doit être fourni.

Pour plus de précisions concernant les routines de maillage par macro-éléments et de résolution du problème aux valeurs propres (notamment les bibliothèques et sous-routines nécessaires), voir les sections associées.

7.7 Calcul de la dérivée normale d'une fonction propre de Dirichlet

(.../r2/der_n_laplace)

Bien que le calcul de la dérivée normale d'une fonction propre de Dirichlet n'ait été utilisé que pour la résolution du problème d'optimisation des valeurs propres, l'algorithme bénéficie de procédures distinctes et les routines spécifiques sont rassemblées dans le fichier *der_n_laplace.f90*, à commencer par la routine principale *der_n_laplace*. Plusieurs sous-routines externes sont toutefois employées :

- Les procédures générales (*general.f90*) pour la lecture et l'écriture des fichiers de données.
- Les sous-routines *ematr* (construction des matrices élémentaires) et *choles* (décomposition de Cholesky) de *laplace.f90*.
- La routine *t3shap* de *fekform.f*.
- La bibliothèque *LAPACK*.

Le fichier *use_der_n_laplace.f90* permet quant à lui l'utilisation de *der_n_laplace*, par appel à la procédure après définition de ses paramètres d'entrée :

- Les noms des fichiers contenant la triangulation du domaine (i.e. les fichiers de nœuds et d'éléments).
- Le nom du fichier contenant les résultats du problème aux valeurs propres.
- Le nom du fichier dans lequel seront sauves les résultats de la procédure.
- Le numéro de la fonction propre dont on cherche à calculer la dérivée normale.
- Le paramètre *masslumping*, définissant si le procédé de mass-lumping sera utilisé pour la construction du second membre du problème.

Le fichier teste également l'écriture d'un fichier pour la visualisation de la dérivée normale (routine *visudn* de *der_n_laplace*), dont la fonction est essentiellement le debugging. Les structures de la procédure *der_n_laplace* et de ses sous-routines sont largement inspirées de *laplace* (résolution du problème aux valeurs propres) et de *poisson* (résolution du problème de Poisson).

7.8 Optimisation de forme

(.../r2/opt_form)

La procédure expérimentale d'optimisation de forme est pilotée par *use_opt_form.f90* et utilise certaines des routines de maillage de Delaunay (*maillspe.f90*), de maillage par macro-éléments (*compact.f90*), de résolution du problème aux valeurs propres (*laplace.f90* ainsi que *fekform.f*), de calcul de la dérivée normale de fonctions propres de Dirichlet (*der_n_laplace.f90*), ainsi que des procédures utilitaires générales (*general.f90*), notamment pour la lecture et l'écriture de fichiers types. En corollaire, les bibliothèques *LAPACK* et *ARPACK* sont également nécessaires. Les routines créées et utilisées spécifiquement pour l'optimisation de forme sont quant à elles rassemblées dans le fichier *optform.f90*.

La structure générale de la procédure est la suivante :

1. Initialisation : lecture des paramètres et des données de départ, construction et copies des fichiers nécessaires.

Début de la boucle principale.

2. Préparation au maillage : lecture du fichier d'avancement de l'algorithme, effacement des fichiers inutilisés, suppression des nœuds internes au domaine et uniformisation des conditions au bord (routine *usesuppnis*), raffinement du bord (via *raffbord*) si jugé nécessaire lors de la précédente itération (paramètre *nraff*), test et modification de la topologie du bord (*bdefbord*), test et modification de la résolution du bord (*resobord*).
3. Maillage de Delaunay.
4. Correction de la triangulation obtenue : traitement des éléments fixes (par découpe ou par suppression) en fonction du paramètre *elf* (*elfixnois*), renumérotation des nœuds.
5. Résolution du problème aux valeurs propres et préparation à la visualisation.
6. Détermination du nouveau domaine à l'aide de la fonction-coût et test de sortie de la boucle via le paramètre d'instruction *flag* (*detnewdom*).

Comme pour d'autres procédures, certains fichiers doivent être joints par l'utilisateur (même lorsqu'ils ne seront pas employés par l'algorithme). Il s'agit, concernant les fichiers contenant des routines, de *defdom.f90* pour la définition de domaine, de *domraff.f90* pour le raffinement, et de *metrique.f90* pour la prise en compte de la métrique.

Est également à joindre toute une série de fichiers de données qu'il serait fastidieux d'énumérer ici (d'autant qu'ils dépendent des options choisies par l'utilisateur) ; des explications complètes se trouvent au début du fichier *use_opt_form.f90*. Mentionnons toutefois les différentes familles de fichiers de données utilisés :

- Ceux du domaine en cours de traitement.
- Ceux du domaine précédent (notés ... *l*).
- Ceux du domaine initial (notés ... *0*).
- Ceux du domaine optimal retenu jusqu'alors (notés ... *b*).

Une dernière famille de fichiers peut encore être considérée. Il s'agit de fichiers qui ne seront pas modifiés par le programme durant toute son exécution (notés ... *s*). Le paramètre *dep* définit laquelle de ces familles de fichiers contient le domaine de départ de l'algorithme (qui peut également être donné par les routines *defdom1* et *defdom2* de *defdom.f90*).

Tous ces fichiers de données sont de types de ceux utilisés par les différentes procédures de maillage, résolution de problème aux valeurs propres et calcul de la dérivée normale de fonctions propres de Dirichlet, à l'exception des fichiers contenant des informations spécifiques à l'algorithme d'optimisation (comme *nomdp*). Les données y figurant sont :

- La valeur propre normalisée du domaine.
- Le maximum d'itérations permises et la tolérance accordée aux solutions lors de la résolution approchée du problème aux valeurs propres (pour *ARPACK*).
- Le facteur de déplacement des nœuds du bord du domaine employé (par rapport à la dérivée de la fonction-coût).
- La tolérance concernant l'algorithme d'optimisation.
- Le nombre d'itérations et le nombre d'itérations effectives (c'est-à-dire sans compter celles ayant mené à une reprise d'un domaine précédent et à un ajustement du paramètre de déplacement des nœuds du bord).

Détaillons encore le rôle de quelques routines importantes du fichier *opt_form.f90*. Le nombre de paramètres et d'options de l'algorithme étant particulièrement élevé, tous ne

seront pas expliqués ci-dessous. Seront notamment mis de côté les paramètres provenant directement des routines détaillées dans les sections précédentes (maillage, résolution de problème aux valeurs propres...). Le fichier *use_opt_form.f90* propose toutefois de nombreuses explications à propos de chaque paramètre et l'utilisateur s'y reportera si besoin.

La routine *bdefbord* permet de prendre en considération des domaines dont la topologie change d'une itération à une autre, par exemple par recoupement de deux parties du domaine. Pour ce faire, elle analyse la courbe obtenue comme image du bord du précédent domaine après déplacement des nœuds du bord; en l'état, cette courbe ne définit pas nécessairement un domaine. La routine *bdefbord* applique donc l'algorithme détaillé à la section 5.3, consistant essentiellement en la suppression des parties superflues de la courbe.

La routine *resobord* a pour but de garantir que, lors de chaque itération de l'algorithme, les segments du bord du domaine soient tous plus ou moins de même grandeur, afin par exemple que les valeurs de la fonction-coût ne soient pas trop influencées par la taille des éléments au bord du domaine. Il s'agit donc, dans un premier temps, de définir une résolution du bord du domaine (donnée par le paramètre *resit*). Elle est définie par la moyenne des longueurs des segments du bord de l'itération en cours. Ensuite, la routine compare chaque segment avec cette résolution : un segment trop long (par exemple plus grand que $\frac{3}{2} \cdot resit$) sera dédoublé; un segment trop court (par exemple plus court que $\frac{1}{3} \cdot resit$) sera supprimé (en remplaçant ses extrémités par leur point milieu). Toutefois, à moins que le segment soit petit au point que d'autres routines ne puissent distinguer ses deux extrémités (segment plus court que le paramètre de résolution global *res*), un segment trop court ne sera supprimé qu'à condition que les angles formés par le segment et ses voisins soient proches de l'angle plat (par exemple entre $3\pi/4$ et $5\pi/4$), afin de ne pas perturber la modification progressive de topologie d'un domaine au fil de l'algorithme.

La routine *elfixnois* permet de traiter les éléments parasites, c'est-à-dire ceux qui ne servent à rien ou, pire, qui empêchent l'algorithme d'optimisation de fonctionner correctement (comme des éléments dont les nœuds ne bougent pas, empêchant ainsi le domaine d'approcher un domaine optimal). Les éléments en question, dits éléments *fixes*, sont ceux dont les trois nœuds sont fixes. La routine propose les différentes options de traitement suivantes, déterminées par la valeur du paramètre *elf* :

- Suppression des éléments triviaux : ceux dont les trois arêtes sont fixes.
- Suppression des éléments *coins* : ceux dont deux arêtes sont fixes.
- Suppression des nœuds isolés (provenant de la suppression d'éléments) : ceux n'appartenant à aucun élément.
- Découpe des éléments fixes restants (via une arête libre) : ceux ayant trois nœuds fixes.

Pour finir, la routine *detnewdom* constitue le cœur de l'algorithme d'optimisation de forme. Ses fonctions, dictées par le paramètre *flag*, sont les suivantes :

- Nouvelle tentative de résolution du problème aux valeurs propre (via *laplace*) si la précédente n'a pas été menée à terme. Les paramètres du problème aux valeurs propres (nombre maximal d'itération de Lanczos, critère d'arrêt. . .) sont alors mis à jour.
- Reprise du domaine jusqu'alors optimal si le domaine actuel est moins bon (en fonction du paramètre *mode*).
- Calcul de la fonction-coût et de sa dérivée.
- Arrêt de l'algorithme si le critère d'arrêt est satisfait.
- Possibilité pour l'utilisateur de raffiner le bord du domaine (en fonction du paramètre *idebug*).
- Poursuite de l'algorithme par modification du (bord du) domaine.

Mentionnons encore deux paramètres particulièrement importants dans le déroulement de l'algorithme dans sa globalité, mais effectifs dans la routine *detnewdom* :

- Le paramètre *mode*, si non nul, donne la fréquence à laquelle le programme analyse le domaine obtenu en regard du domaine jusqu'alors optimal (et sauvé dans des fichiers dédiés) : si le domaine en cours de traitement est meilleur que le domaine jusqu'alors optimal, ce dernier est remplacé et l'algorithme suit son cours normalement ; dans le cas contraire, on recharge le domaine optimal et on ajuste le paramètre de déplacement des nœuds du bord (facteur de la fonction-coût en chaque nœud). On tente ainsi d'éviter de passer outre un domaine optimal à cause d'un déplacement trop brusque des nœuds du bord. Il ne faut toutefois pas hésiter à augmenter la valeur de *mode* pour limiter le risque de coincer le domaine au voisinage d'un optimal local, même si la vitesse de convergence peut alors être fortement réduite.
- Le paramètre *idebug* donne la fréquence à laquelle l'utilisateur décide de la direction à donner à l'algorithme, à savoir stopper, continuer sans effectuer de changement ou continuer en raffinant le bord du domaine. Cette dernière option est particulièrement utile lorsque le domaine ne bouge plus beaucoup mais que le critère d'arrêt n'est pas encore atteint (dérivée de la fonction-coût non négligeable).

7.9 Quelques remarques géométriques concernant la métrique

Quelques explications supplémentaires s'imposent probablement, pour les non initiés à la géométrie différentielle, au sujet de la construction des fichiers de données *param.f90* et surtout *metrique.f90*. Ces fichiers contiennent en effet la ou les fonctions paramétriques de la surface, respectivement des informations sur leurs métriques associées. Les quelques rappels élémentaires de géométrie ci-dessous expliquent brièvement ces concepts et détaillent les éléments à inclure dans les fichiers. Pour plus de détails, l'utilisateur se référera à n'importe quel livre de géométrie différentielle, mais par exemple [14].

Considérons une surface S et (f, U) une *paramétrisation locale* (ou *carte*) de S . Rappelons que cela signifie que $f : U \rightarrow S$ est \mathcal{C}^∞ , qu'il s'agit d'un homéomorphisme sur son image, que la dérivée $Df(x)$ de f en x est injective pour tout $x \in U$ et que U est un ouvert de \mathbb{R}^2 .

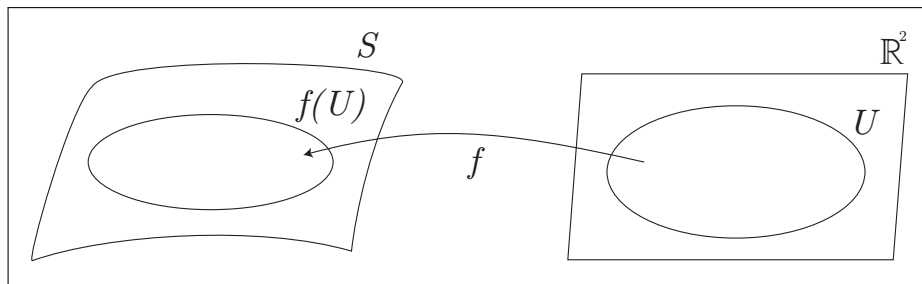


FIGURE 7.1 – Paramétrisation locale (U, f) d'une surface S .

Le fichier *param.f90* contient alors l'expression fonctionnelle de f ou, dans le cas où plusieurs cartes seraient nécessaires à la paramétrisation du domaine de S considéré, de toutes les fonctions paramétriques associées (on parle alors d'un *atlas* de S). Rappelons toutefois que ce fichier n'est nécessaire que pour visualiser le maillage ou les fonctions propres directement sur la surface et qu'il n'est pas utilisé à des fins calculatoires.

Si S est une surface de \mathbb{R}^n , alors, pour tout $p \in S$, le produit scalaire usuel de \mathbb{R}^n induit un produit scalaire $G(p) = \langle \cdot, \cdot \rangle_p$ sur le plan tangent $T_p S$ de S en p ; il s'agit de la *métrique* de S (souvent confondue avec la *première forme fondamentale* de S , qui est en fait la forme quadratique associée). Reprenons alors la carte (U, f) . On peut calculer l'expression matricielle G_f de G dans la base canonique de \mathbb{R}^2 via la carte (U, f) , à partir de la relation

$$G_f(x)(\eta_1, \eta_2) = \langle Df(x)\eta_1, Df(x)\eta_2 \rangle_{f(x)},$$

avec $x \in U$ et $\eta_1, \eta_2 \in \mathbb{R}^2$. Matriciellement, on a

$$\eta_1^t \cdot G_f(x) \cdot \eta_2 = \eta_1^t \cdot (Df(x))^t \cdot Df(x) \cdot \eta_2,$$

d'où, en notant $G_f(x) = \left(g_{ij}^f(x) \right)_{i,j=1,2}$,

$$g_{ij}^f(x) = e_i^t \cdot (Df(x))^t \cdot Df(x) \cdot e_j = \left(\frac{\partial f}{\partial x_i}(x) \right)^t \cdot \frac{\partial f}{\partial x_j}(x).$$

Si S est munie d'une métrique G non triviale (c'est-à-dire d'un produit scalaire non usuel), le même raisonnement conduit à

$$G_f(x)(\eta_1, \eta_2) = G(x)(Df(x)\eta_1, Df(x)\eta_2).$$

Dans ce cas, si G peut de plus être donné matriciellement, on obtient sous forme matricielle

$$g_{ij}^f(x) = \left(\frac{\partial f}{\partial x_i}(x) \right)^t \cdot G(f(x)) \cdot \frac{\partial f}{\partial x_j}(x).$$

Le fichier *metrique.f90* contient alors diverses informations sur la métrique G_f ou, dans le cas où plusieurs cartes seraient nécessaires à la paramétrisation du domaine de S considéré, sur les métriques de toutes les fonctions paramétriques associées. Précisément, l'utilisateur doit fournir, pour tout x , la valeur des coefficients $(g_{ij}^f(x))_{i,j=1,2}$ de la matrice inverse de $G_f(x)$, ainsi que la racine $\sqrt{\det G_f(x)}$ du déterminant de $G_f(x)$. Rappelons que ces données sont utilisées lors de la construction des matrices du système.

Remarquons pour finir que, à condition que cela ne prête pas à confusion, on note simplement G au lieu de G_f . Cette notation a d'ailleurs été adoptée dans ce travail.

Bibliographie

- [1] R.A. Adams, *Sobolev spaces*, Pure and applied mathematics, Academic Press, 1975.
- [2] L.V. Ahlfors, *Complex analysis : An introduction to the theory of analytic functions of one complex variable*, International series in pure and applied mathematics, McGraw-Hill, 1979.
- [3] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *Lapack user's guide*, 3 ed., SIAM, 1999.
- [4] C. Anné, *Spectre du laplacien et écrasement d'anses*, Annales scientifiques de l'E.N.S. de Paris **20** (1987), 271–280.
- [5] P.R.S. Antunes and P. Freitas, *Numerical optimization of low eigenvalues of the dirichlet and neumann laplacians*, Journal of optimization theory and applications **154** (2012), 235–257.
- [6] M.G. Armentano and R.G. Durán, *Mass-lumping or not mass-lumping for eigenvalue problems*, Numerical methods for partial differential equations **19** (2003), 653–664.
- [7] I. Babuška and J.E. Osborn, *Finite element-galerkin approximation of the eigenvalues and eigenvectors of selfadjoint problems*, Mathematics of computation **52** (1989), 275–297.
- [8] P. Bérard, *On ne peut pas entendre la forme d'un tambour : Introduction*, www-fourier.ujf-grenoble.fr/~pberard/D/tambours.pdf (consulté le 11.04.2012), 2001.
- [9] H. Brezis, *Analyse fonctionnelle : Théorie et applications*, Collection mathématiques appliquées pour la maîtrise, Masson, 1983.
- [10] D. Bucur and A. Henrot, *Minimization of the third eigenvalue of the dirichlet laplacian*, Proceedings of the royal society London **456** (2000), 985–996.
- [11] P. Buser, J. Conway, P. Doyle, and K.D. Semmler, *Some planar isospectral domains*, International mathematics research notices **9** (1994), 391–400.
- [12] G. Buttazzo and G. Dal Maso, *An existence result for a class of shape optimization problems*, Archive for rational mechanics and analysis **122** (1993), 183–195.
- [13] D. Calvetti, L. Reichel, and D.C. Sorensen, *An implicit restarted lanczos method for large symmetric eigenvalue problems*, Electronic transactions on numerical analysis **2** (1994), 1–21.

- [14] M.P. Do Carmo, *Differential geometry of curves and surfaces*, Prentice-Hall, 1976.
- [15] I. Chavel, *Eigenvalues in riemannian geometry*, Pure and applied mathematics, vol. 115, Academic Press, 1984.
- [16] P.G. Ciarlet, *The finite element method for elliptic problems*, 2 ed., Studies in mathematics and its applications, vol. 4, North-Holland Publishing Company, 1978.
- [17] R.L. Courant, *Variationnal methods for the solution of problems of equilibrium and vibration*, Bulletin of the american mathematical society **49** (1943), 1–23.
- [18] S.J. Cox and M. Ross, *Extremal eigenvalue problems for starlike planar domains*, Journal of differential equations **120** (1995), 174–197.
- [19] L.C. Evans, *Partial differential equations*, 3 ed., Graduate studies in mathematics, vol. 19, AMS, 2002.
- [20] G. Faber, *Beweiss, dass unter allen homogenen membranen von gleicher fläche und gleicher spannung die kreisförmige den tiefsten grundton gibt*, Sitzungsberichte der mathematisch-physikalischen Klasse der Bayerischen Akademie der Wissenschaften zu München Jahrgang (1923), 169–172.
- [21] B.G. Galerkin, *Rods and plates. series occurring in various questions concerning the elastic equilibrium of rods and plates*, Vestnik inzhenerov i tekhnikov **19** (1915), 897–908.
- [22] P.L. Georges, *Génération automatique de maillages : Applications aux méthodes d'éléments finis*, Collection recherche en mathématiques appliquées, vol. 16, Masson, 1991.
- [23] D. Gilbarg and N.S. Trudinger, *Elliptic partial differential equations of second order*, 2 ed., Classics in mathematics, Springer, 2001.
- [24] C. Gordon, D. Webb, and S. Wolpert, *Isospectral plane domains and surfaces via riemannian orbifolds*, Inventiones mathematicae **110** (1992), 1–22.
- [25] A. Henrot, *Minimization problems for eigenvalues of the laplacian*, Journal of evolution equations **3** (2003), 443–461.
- [26] ———, *Extremum problems for eigenvalues of elliptic operators*, Frontiers in mathematics, Birkhäuser, 2006.
- [27] A. Henrot and E. Oudet, *Le stade ne minimise pas λ_2 parmi les ouverts convexes du plan*, Comptes rendus de l'académie des sciences de Paris série 1 mathematics **332** (2001), 417–422.
- [28] A. Henrot and M. Pierre, *Variation et optimisation de formes : Une analyse géométrique*, Mathématiques et applications, vol. 48, Springer, 2005.
- [29] M. Kac, *Can one hear the shape of a drum ?*, American mathematics monthly **73** (1966), 1–23.
- [30] E. Krahn, *Über eine von rayleigh formulierte minimaleigenschaft des kreises*, Mathematische Annalen **94** (1925), 97–100.

- [31] ———, *Über minimaleigenschaften der kugel in drei und mehr dimensionen*, Acta et commentationes universitatis Tartuensis Dorpatensis **A9** (1926), 1–44.
- [32] R.B. Lehoucq, D.C. Sorensen, and C. Yang, *Arpack user's guide : Solutions of large-scale eigenvalue problems with implicitly restarted arnoldi methods*, SIAM, 1998.
- [33] M. Maintrot, *Finite element method on riemann surfaces and applications to the laplacian spectrum*, Ph.D. thesis, Ecole polytechnique fédérale de Lausanne, 2012.
- [34] Netlib, <http://www.netlib.org> (consulté le 11.04.2012).
- [35] E. Oudet, *Numerical minimization of eigenmodes of a membrane with respect to the domain*, ESAIM : COCV **10** (2004), 315–330.
- [36] R. Oudompheng, *Les harmoniques sphériques : Quand l'algèbre et la géométrie rejoignent la physique et les edp*, www.normalesup.org/~oudomphe/textes/20100108-laplacien.pdf (consulté le 11.04.2012), 2010.
- [37] C.C. Paige, *The computation of eigenvalues and eigenvectors of very large sparse matrices*, Ph.D. thesis, University of London, 1971.
- [38] B.N. Parlett, *The symmetric eigenvalue problem*, Classics in applied mathematics, vol. 20, SIAM, 1998.
- [39] G. Pòlya, *On the characteristic frequencies of a symmetric membrane*, Mathematische Zeitschrift **63** (1955), 331–337.
- [40] G. Pòlya and G. Szegő, *Isoperimetric inequalities in mathematical physics*, Annals of mathematics studies, vol. 27, Princeton University Press, 1951.
- [41] P.A. Raviart and J.M. Thomas, *Introduction à l'analyse numérique des équations aux dérivées partielles*, Collection mathématiques appliquées pour la maîtrise, Masson, 1983.
- [42] J.W.S. Rayleigh, *On the theory of resonance*, Philosophical transactions of the society of London **A161** (1870), 77–118.
- [43] ———, *The theory of sound*, 2 ed., vol. 1, Macmillan, 1894.
- [44] ———, *The theory of sound*, 2 ed., vol. 2, Macmillan, 1896.
- [45] W. Ritz, *über eine neue methode zur lösung gewisser variations probleme der mathematischen physik*, Zeitschrift für angewandte Mathematik und Mechanik **135** (1908), 1–61.
- [46] Y. Saad, *Numerical methods for large eigenvalue problems*, Series in algorithms and architectures for advanced scientific computing, Manchester University Press, 1992.
- [47] J. Sokolowski and J.P. Zolesio, *Introduction to shape optimization : Shape sensity analysis*, Springer series in computational mathematics, vol. 10, Springer, 1992.
- [48] G. Strang and G. Fix, *An analysis of the finite element method*, Series in automatic computation, Prentice-Hall, 1973.
- [49] A. Strohmaier and V. Uski, *An algorithm for the computation of eigenvalues, spectral zeta functions and zeta-determinants on hyperbolic surfaces*, (2011).

- [50] A.H. Stroud, *Approximate calculation of multiple integrals*, Prentice-Hall series in automatic computation, Prentice-Hall, 1971.
- [51] B.A. Troesch, *Elliptical membranes with smallest second eigenvalue*, Mathematics of computation **27** (1973), 767–772.
- [52] M.J. Turner, R.W. Clough, H.C. Martin, and L.C. Topp, *Stiffness and deflection analysis of complex structures*, Journal of the aeronautical sciences **23** (1956), 805–823.
- [53] S.A. Wolf and J.B. Keller, *Range of the first two eigenvalues of the laplacian*, Proceedings of the royal society London **447** (1994), 397–412.
- [54] Wolfram MathWorld, <http://www.mathworld.wolfram.com> (consulté le 11.04.2012).