

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABDELHAMID IBN BADIS DE MOSTAGANEM



FACULTE DES SCIENCES EXACTES ET SCIENCES DE LA NATURE ET
DE LA VIE
DEPARTEMENT DE MATHÉMATIQUES

MÉMOIRE

Présenté par
LADJAL Nouria

Pour Obtenir
LE DIPLÔME DE MAGISTER

Spécialité : **Mathématiques**

Option : Analyse des Systèmes, Contrôle et Optimisation Numériques

Intitulé :

Une Approche pour la Résolution des Problèmes aux Limites par les Ondelettes

Composition du jury de soutenance

Nom et prénom	Grade	Qualité	appartenance administrative
AMIR Abdessamad	Maître de conférences A	Président	UMAB
BOUAGADA Djillali	Maître de conférences A	Examineur	UMAB
DAHMANI Zoubir	Maître de conférences A	Examineur	UMAB
BAHRI Sidi Mohamed	Maître de conférences A	Encadreur	UMAB
BELHAMITI Omar	Maître de conférences B	Co-Encadreur	UMAB

Année Universitaire : 2011/2012

Une Approche pour la Résolution de Problèmes
Aux Limites Séparées par
les Ondelettes

LADJAL Nouria

Remerciements

Tout d'abord je remercie ALLAH le tout puissant de m'avoir donnée la santé, la volonté et le courage pour effectuer ce modeste travail.

Je remercie vivement mes deux directeurs : Dr. *BAHRI Sidi Mohamed*, et Dr. *BELHAMITI Omar* qui m'ont donné l'opportunité de travailler avec eux.

J'ai l'honneur de grouper tous les membres de l'équipe pédagogique de la post-graduation « *Analyse des Systèmes, Contrôle et Optimisation Numériques* » dans le jury de ma soutenance.

Je tiens à exprimer ma reconnaissance à Dr. *AMIR Abdessamad*, d'avoir présidé le jury de mon mémoire et Dr. *BOUAGADA Djilali* et Dr. *DAHMANI Zoubir* d'avoir accepté de faire partie du jury de ce mémoire.

En particulier, je tiens à remercier chaleureusement Dr. *BELHAMITI Omar* pour avoir su me donner confiance en moi, et pour ses conseils adéquats, et ses orientations notamment lors de la rédaction de ce manuscrit.

Un grand merci à tous les professeurs que j'ai rencontré durant tout mon cursus.

Je remercie également tout les collègues de la poste-graduation *BELGASSEM Rachid*, *ELOS-MANI Aissa* , et *MARIR Saliha* avec lesquels j'ai passé de bons moments.

De plus, ce mémoire n'aurait pas abouti sans le soutien de mes amis et amies un par un en particulier Mme *MARIR Saliha* qui ma toujours encouragé, et soutenue dans tous les moments, et Melle *HAMMOU Amouria*, Melle *KISSARLI Zineb* et Melle *BENSIKADOUR Djamia* pour ses remarques très utiles et pour ses conseils.

Ma reconnaissance va à Mr. *ZAMOURI Tahir* pour son encouragement, soutient, aide, et ses conseils. Merci !

Merci à Mme *MERIOUL T.* pour le temps qu'elle a consacré à la lecture de cet mémoire.

Enfin, un dernier hommage à celui qui m'a accompagné, et qui m'a porté à bout de bras lors de la rédaction de ce mémoire sans limitations.

Dédicaces

Je dédie ce modeste travail à tous les membres de ma famille plus particulièrement à mes adorables parents.

A mon cher Père pour son soutien effective, son encouragement, sa compréhension, sa présence et ses sacrifices. A ma très chère *Mère* pour sa patience, son courage, ses conseils très utiles surtout pour son amour ; que *DIEU* me les garde.

Ma reconnaissance va à tous mes frères *Mansour*, *Hadj* et *Charef*, à mes sœurs surtout *Fatiha*, pour son encouragement et pour son affectation et à ma chère sœur *Allou* et son mari *Nadjem*.

Merci à tous les enfants de ma famille *Mohamed*, *Abdelhamid*, *Ahlem* et surtout *Chourouk* et *CHAIB Djamel*.

A tous ceux qui m'ont soutenu surtout dans les moments les plus difficiles.

En particulier j'aimerais bien dédier ce mémoire à celle qui nous a quitté à jamais, à ma très chère Grand-mère.

Tu es et tu resteras toujours dans mon cœur. Je prie **le Tout-Puissant** de l'accueillir dans son vaste Paradis et de lui accorder sa Sainte Miséricorde.

NOTATIONS

\mathbb{N}	: Ensemble des entiers naturels.
\mathbb{Z}	: Ensemble des entiers relatifs.
\mathbb{R}	: Ensemble des nombres réels.
$\langle \cdot, \cdot \rangle$: Produit scalaire de L^2 .
$\ f\ $: Norme de f .
$E \oplus F$: Somme directe de deux espaces vectoriels.
$E \otimes F$: Produit tensoriel de deux espaces vectoriels.
Δ	: Opérateur de Laplace.
∇	: Opérateur de Gradient.
$\partial\Omega$: La frontière du domaine Ω .
$\Omega^{(s)}$: L'élément fini numéro s .
n_{x_e}	: Nombre de nœuds suivant l'axe des x
n_{y_e}	: Nombre de nœuds suivant l'axe des y
h_{x_e}	: Le pas de discrétisation par MEF suivant l'axe des x
h_{y_e}	: Le pas de discrétisation par MEF suivant l'axe des y
n_{x_c}	: Nombre de points de collocation intérieurs suivant l'axe des x
n_{y_c}	: Nombre de points de collocation intérieurs suivant l'axe des y
h_{x_c}	: Le pas de discrétisation par MCO suivant l'axe des x
h_{y_c}	: Le pas de discrétisation par MCO suivant l'axe des y

ABRÉVIATIONS UTILISÉS

MCO	: M éthode de C ollocation O rthogonale
MEF	: M éthode des E léments F inis
MCOEF	: M éthode de C ollocation O rthogonale sur E léments F inis
EDP	: É quation aux D érivées P artielles.
TOC	: T ransformée en O ndelettes C ontinue.
AMR	: A nalyse M ulti- R ésolution.

Table des matières

1	Introduction	7
2	Rappels des outils mathématiques utilisés	10
2.1	Méthode des Résidus Pondérés	11
2.1.1	Méthode de collocation par points	12
2.1.2	Méthode de collocation par sous domaines	13
2.1.3	Méthode de Galerkin	13
2.1.4	La méthode de collocation orthogonale	13
2.2	Quelque notions sur les Matrices	14
2.2.1	Espace de <i>Sobolev</i>	15
2.2.2	Formule de <i>Green</i>	15
2.3	Formulation variationnelle d'un problème de valeurs aux limites	16
2.3.1	Résolution d'un problème variationnelle	16
2.4	Ondelettes	18
2.4.1	Transformé en ondelette continu (TOC)	19
2.4.2	Transformé en ondelette discrète (TOD)	20
2.4.3	Analyse multi résolution (AMR)	20
2.4.4	Base d'ondelettes	21
2.4.5	Ondelettes bidimensionnelles	22
2.4.6	Analyse multi-résolution en 2 dimensions	22
3	La méthode de collocation orthogonale sur les éléments finis	25
3.1	Discrétisation du domaine Ω	25
3.1.1	Discrétisation par la méthode des éléments finis	26
3.1.2	Discrétisation par la méthode de collocation orthogonale	28
3.2	Déscription des équations par la méthode de collocation orthogonale (MCO)	29
3.2.1	Transformation de l'équation sur un élément de référence	29
3.2.2	Le calcul de la matrice locale	32
3.3	L'assemblage de matrices locales	33
3.3.1	Assemblage de l'élément 1	34
3.3.2	Assemblage de l'élément 2	35

3.3.3	Assemblage du dernier élément	36
3.4	Imposée les conditions aux limites	36
3.5	La résolution du système	39
3.6	Erreur d'approximation	39
3.7	L'utilisation des ondelettes pour la méthode de collocation orthogonale sur les éléments finis	41
4	Simulations et applications	43
5	Conclusion	47
6	ANNEXES	48
6.1	Polynôme de <i>Lagrange</i>	48
6.2	Base de Riesz	50
6.3	Transformé de Fourier	50
6.4	Méthode LU	50
	Annexe B	53
6.5	PROGRAMATION	53
	Bibliographie	68

| Introduction0

Le calcul scientifique est une discipline qui consiste à développer, analyser et appliquer des méthodes relevant de domaines mathématiques aussi variés que l'analyse, l'algèbre linéaire, la géométrie, la théorie de l'approximation et le calcul différentiel. En mathématiques appliquées, la résolution d'équations aux dérivées partielles occupe une place importante dans les mises de la recherche actuelle.

Les problèmes posés par la physique, la mécanique, la biologie, la chimie et les autres sciences de l'ingénieur sont modélisés mathématiquement par des équations différentielles ou par des systèmes d'équations différentielles sur un domaine géométrique, avec des conditions aux limites sur ses frontières. Généralement ces modèles sont représentés par l'équation suivante :

$$a(x, y)\Delta U(x, y) + b(x, y)\nabla U(x, y) + c(x, y)U(x, y) = f(x, y), \text{ sur } \Omega$$

où,

Ω est un ouvert de \mathbb{R}^2 , Δ est l'opérateur de *Laplace*, ∇ est l'opérateur *Gradient*, $U(x, y)$ est la fonction de $L^2(\Omega)$ qui représente la solution recherchée, $a(x, y)$, $c(x, y)$ sont deux fonctions continue sur Ω , $b(x, y)$ est un vecteur ligne, et le second membre de l'équation $f(x, y)$ est une fonction continue, avec les conditions aux limites définies sur la frontière du domaine Ω .

Dans les simulations numériques, les conditions aux limites imposées à la solution du problème sont souvent de type *Dirichlet*, *Neumann* ou mixtes de type *Robin*. Comme on veut générer des espaces vectoriels, on suppose alors que les conditions aux limites sont homogènes.

L'efficacité d'un algorithme est mesurée par la rapidité de sa convergence et sa précision.

Avec l'évolution scientifique et technologique, certains nouveaux problèmes difficiles à étudier sont apparus et dont la solution exacte est pénible à atteindre. Cependant, les méthodes numériques remplissent ce manque et donnent des solutions approchées de la solution exacte grâce à des *algorithmes* et des *logiciels* de calcul numérique.

Il est nécessaire de signaler que le choix de la méthode numérique dépend des particularités propres du système à étudier. Il n'existe en effet aucune " *boite noire* " qui puisse résoudre avec précision tous les types de problèmes.

L'objectif de ce mémoire est de développer une méthode bien adaptée à la résolution numérique des problèmes de valeurs aux limites, dite « *Méthode de Collocation Orthogonale sur les Eléments Finis* » dans le cas bidimensionnel. Cette méthode qui a été développée dans le cas mono dimensionnel par Dr. *BELHAMITI. O* [14].

La méthode de collocation orthogonale sur les éléments finis est l'arrangement de deux méthodes :

- la *méthode des éléments finis* qui consiste à discrétiser le domaine d'étude en un nombre fini d'éléments finis de forme géométrique simple.
- la *méthode de collocation orthogonale* appliquée sur chacun des éléments afin de le discrétiser en un nombre fini de points de collocation.

La mise en œuvre pratique est proposée dans le langage MATLAB qui présente l'avantage d'être d'une utilisation facile et de bénéficier d'une large propagation.

Le présent mémoire est organisé comme suit :

Le premier chapitre est introductif où on y trouvera des rappels des outils et des méthodes numériques utilisés au long de ce travail dont la méthode des résidus pondérés, la méthode de collocation orthogonale, et quelques propriétés concernant les matrices et enfin un brève rappel sur la théorie des ondelettes, l'analyse multi-résolution et les bases d'ondelettes en deux dimensions.

Le deuxième chapitre, représentant l'objet de ce mémoire, est conçu pour étudier la méthode de collocation orthogonale sur les éléments finis dans le cas bidimensionnel. On commence par la discrétisation du domaine en éléments puis on cherche à appliquer la MCO sur chacun des éléments finis du domaine d'étude afin d'obtenir les *matrices locales*, puis on assemble ces dernières en une seule matrice dite *matrice globale* et impose les conditions aux limites. Et enfin on applique la factorisation **LU** sur le problème obtenu pour trouver la solution qu'on cherche.

Le chapitre qui suit est plus pratique que théorique. Il constitue des applications numériques, les simulations et la programmation de la méthode étudiée implémentée en MATLAB. Chaque programme MATLAB est accompagné d'une brève description des paramètres d'entrées et de sorties.

Dans le quatrième chapitre nous avons donné une représentation formelle (dite non-standard) des opérateurs différentiels suivant une base d'ondelettes séparables bidimensionnelles. Pour les applications numériques, cette représentation suggère une approche pseudo différentielle que nous n'avons pas introduit ici et que nous avons laissé en perspectives de ce travail car elle nécessite beaucoup de calculs.

Préliminaires

Ce chapitre est conçu pour rappeler quelques notions élémentaires de mathématiques nécessaires pour la compréhension de ce mémoire.

Définition 2.0.1 *Un problème est dit stationnaire si le temps ne collabore pas comme une variable sinon c'est un problème d'évolution.*

Définition 2.0.2 (support d'une fonction) *Soit la fonction $\varphi : \Omega \rightarrow \mathbb{R}$. On définit le support de φ par*

$$\text{supp}(\varphi) = \overline{\{x \in R / \varphi(x) \neq 0\}}.$$

Définition 2.0.3 (Espace des fonctions régulières) *Soit Ω un ouvert de \mathbb{R}^n de bord $\partial\Omega$. On note par $C^\infty(\Omega)$ ou $D(\Omega)$ l'espace vectoriel des fonctions C^∞ sur Ω à support compact inclus dans Ω .*

2.1 Méthode des Résidus Pondérés

La méthode des résidus pondérés de **Grandell** est une méthode d'approximation qui consiste à remplacer un problème mathématique définie sur un milieu continu régi par des équations différentielles ou intégrales, par un problème discret équivalent régi par des équations matricielles que l'on sait résoudre numériquement.

Il est question de résoudre le problème aux limites en deux dimensions :

$$\begin{cases} (LU_{ex} + f)(x, y) = 0 & \text{sur } \Omega, \\ (CU)(x, y) = g(x, y) & \text{sur } \partial\Omega, \end{cases} \quad (2.1.1)$$

avec U_{ex} désigne la solution exacte du problème, $f \in L^k(\Omega)$ où k est le degré de dérivation le plus élevé de l'opérateur de dérivation L .

Le principe de cette méthode est la construction de la solution cherchée sous la forme d'une combinaison linéaire de fonctions polynômiales ou trigonométriques :

$$U(x, y) = \sum_{ij} (U_{ij}P_{ij})(x, y), \quad (2.1.2)$$

où U_{ij} sont des paramètres de l'approximation et P_{ij} sont les fonctions de base ou de forme.

Remarque 2.1.1 Si on remplace la solution trouvée dans le problème (2.1.1), on obtient un reste non nul appelé "résidu", on le note généralement par $\mathcal{R}(x, y)$, et on écrit

$$\mathcal{R}(x, y) = LU(x, y) + f(x, y) \neq 0. \quad (2.1.3)$$

Le but est de trouver une meilleure approximation i.e ; obtenir un résidu le plus petit possible. Donc on arrive à un problème de minimisation de résidus.

Pour minimiser le résidu, on cherche un ensemble de fonctions strictement positives ψ_{ij} dites *fonctions de pondérations* ou *fonctions tests (poids)* définies sur le domaine Ω telles qu'elles soient dérivables jusqu'à l'ordre du système, linéairement indépendantes, vérifient les conditions aux limites, et sont orthogonales avec le résidu i.e ;

$$\iint_{\Omega} \psi_{ij}(x, y) \mathcal{R}(x, y) dx dy = \iint_{\Omega} \psi_{ij}(x, y) [LU(x, y) + f(x, y)] dx dy = 0, \quad (2.1.4)$$

On conclut que, la méthode des résidus pondérés cherche à annuler le résidu en le pondérant sur le domaine Ω par un nombre fini de fonctions ψ_{ij} .

Le choix des fonctions tests ψ_{ij} dépend du type de la méthode utilisée : la méthode de collocation par points, de collocation par sous domaines, la méthode de *Galerkine*, et la méthode des moments etc.

2.1.1 Méthode de collocation par points

Dans cette méthode les fonctions de pondération sont représentées par les *impulsions de Dirac* en des points de collocations (x_i, y_j) telles que :

$$\psi_{ij}(x, y) = \begin{cases} 0 & \text{si } (x, y) \neq (x_i, y_j) \\ 1 & \text{si } (x, y) = (x_i, y_j) \end{cases}, \quad i = 1, \dots, n; \quad j = 1, \dots, .$$

Alors l'équation intégrale (2.1.4) devient

$$\iint_{\Omega} \psi_{ij}(x, y) \mathcal{R}(x, y) dx dy = 0 \Rightarrow \mathcal{R}(x_i, y_j) = 0,$$

ainsi on obtient un système d'équations algébriques dont les inconnus sont les U_{ij} . Alors l'intérêt du choix des fonctions de *Dirac* est pour éviter le calcul de l'intégrale sur le domaine.

2.1.2 Méthode de collocation par sous domaines

La méthode de collocation par sous domaines est conçu par *Biezeno et Koch* en **1923**. Dans cette méthode le domaine Ω est considéré comme une réunion fini de sous domaines finis Ω_k , de telle sorte que ces derniers forment une partition du domaine Ω . Les fonctions poids sont choisies comme étant les *fonctions Indicatrices* :

$$\psi_{ij}(x, y) = \begin{cases} 0 & \text{si } (x, y) \notin \Omega_k, \\ 1 & \text{si } (x, y) \in \Omega_k, \end{cases}$$

avec $\Omega_k = X_i \times Y_j = [x_i, x_{i+1}] \times [y_j, y_{j+1}]$, $k = 1, \dots, n_{sd}$ (n_{sd} étant le nombre de sous domaines).

2.1.3 Méthode de Galerkinge

Cette méthode introduite par le mathématicien et ingénieur *Boris Grigoryevich Galerkinge* en **1915**. Dans cette méthode les fonctions de pondérations sont les fonctions de base P_{ij} de la solution approximative :

$$\psi_{ij} = P_{ij}, \quad i = 1, \dots, n_x ; \quad j = 1, \dots, n_y$$

où n_x, n_y représente le nombre des points de collocation sur l'axe des x et l'axe des y respectivement.

Remarque 2.1.2 *En 1978 Villadsen et Michelsen ont comparé les solutions des différentes méthodes des résidus pondérés, pour des problèmes de valeurs aux limites et ont conclu que la méthode de Galerkinge est la plus compliquée à mettre en œuvre mais elle fournit des résultats bien précis. Par contre, la méthode de collocation par points est simple à réaliser mais elle fournit une mauvaise précision, car les solutions sont très sensibles à la fois du choix des points de collocations, et du nombre de paramètres de l'approximation. Dans ce dernier cas, Villadsen a suggéré de choisir les racines du polynôme de Jacobi comme des points de collocation. Dans la méthode de collocation par point, les résultats obtenus sont aussi bons que ceux de Galerkinge, d'où la naissance d'une nouvelle méthode dite méthode de collocation orthogonale.*

2.1.4 La méthode de collocation orthogonale

La méthode de collocation orthogonale est la méthode de collocation par points dont les points de collocation sont pris comme les racines d'un polynôme de *Jacobi*.

Les polynômes de *Jacobi* de degré n sont des polynômes orthogonaux notés par $J_n^{(\alpha, \beta)}$ et sont définis sur l'intervalle de d'orthogonalité $[0, 1]$ par la combinaison linéaire suivante :

$$J_n^{(\alpha, \beta)}(x) = \sum_{i=0}^n (-1)^{n-i} \gamma_{n,i} x^i, \quad (2.1.5)$$

où, α, β : paramètres caractérisants les polynômes de tels sorte que $\alpha > -1, \beta > -1$.

$\gamma_{n,i}$: des coefficients constants tels que :

$$\begin{aligned} \gamma_{n,0} &= 1, \\ \gamma_{n,i} &= \frac{n-i+1}{i} \frac{n+i+\alpha+\beta}{i+\beta} \gamma_{n,i-1}, \end{aligned}$$

avec

$$\gamma_{n,i} = \frac{n!}{i!(n-i)!} \frac{\Gamma(n+i+\alpha+\beta+1)}{\Gamma(n+\alpha+\beta+1)} \frac{\Gamma(\beta+1)}{\Gamma(i+\beta+1)},$$

où la fonction $\Gamma(\cdot)$ est la fonction *Gamma* (voir Annexe).

Les polynômes de *Jacobi* sont orthogonaux alors pour tout $x \in [0, 1]$ on a

$$\langle J_n^{(\alpha, \beta)}, J_m^{(\alpha, \beta)} \rangle = \int_0^1 (1-x)^\alpha x^\alpha J_n^{(\alpha, \beta)}(x) J_m^{(\alpha, \beta)}(x) dx = \begin{cases} 0 & \text{si } n \neq m \\ \neq 0 & \text{sinon} \end{cases},$$

avec $m = 1, \dots, n ; \alpha, \beta > -1$.

Remarque 2.1.3 Les points de collocation de Jacobi sont répartis avec une symétrie parfaite soit par rapport à la racine au milieu si le degré du polynôme de Jacobi est impair soit par rapport à l'intervalle au milieu si le degré est pair avec une accumulation sur les deux bords.

2.2 Quelques notions sur les Matrices

a- Produit de deux matrices

Le produit d'une matrice $A = (a_{ij})_{\substack{1 \leq i \leq m \\ 1 \leq j \leq p}}$ de taille (m, p) par une matrice $B = (b_{ij})_{\substack{1 \leq i \leq p \\ 1 \leq j \leq n}}$ de taille (p, n) est la matrice $C = (c_{ij})_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}}$ de taille (m, n) , dont les coefficients sont donnés par

$$c_{ij} = \sum_{k=1}^p a_{ik} \times b_{kj} \quad , \quad i = 1, \dots, m, j = 1, \dots, n.$$

Le produit matriciel est associatif et distributif par rapport à la somme matricielle, mais il n'est pas commutatif en général.

b- Matrices bandes

On dit qu'une matrice $A = (a_{ij})_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}}$ de taille $m \times n$ est une matrice bande si elle n'admet d'éléments non nuls que sur un "certain nombre" de diagonales autour de la diagonale principale. Plus précisément, on dit que A est une matrice bande- p inférieure si $a_{ij} = 0$ quand $i > j + p$ et bande- q supérieure si $a_{ij} = 0$ quand $j > i + q$.

On appelle simplement matrice bande- p une matrice qui est bande- p inférieure et supérieure.

Les matrices diagonales sont des matrices bandes pour lesquelles $p = q = 0$.

Les matrices triangulaires correspondent à $p = m - 1, q = 0$ (triangulaires inférieures), ou $p = 0, q = n - 1$ (triangulaires supérieures).

2.3 Espace de Sobolev

Soit Ω un ouvert de \mathbb{R}^n .

Définition 2.3.1 (Espace $L^2(\Omega)$) L'espace $L^2(\Omega)$ est un espace des fonctions définies sur Ω et de carré intégrables

On définit l'espace de Sobolev d'ordre 1 noté $H^1(\Omega)$ par :

$$H^1(\Omega) = \left\{ U \in L^2(\Omega), \frac{\partial U}{\partial x_i} \in L^2(\Omega), \forall i \right\},$$

L'espace

$$H^2(\Omega) = \left\{ U \in L^2(\Omega), \frac{\partial U}{\partial x_i} \in L^2(\Omega) \text{ et } \frac{\partial^2 U}{\partial x_i \partial x_j} \in L^2(\Omega), \forall i, j \right\}$$

est l'espace de Sobolev d'ordre 2.

On note :

$$H_0^1(\Omega) = \{ U \in H^1(\Omega), U = 0 \text{ sur } \partial\Omega \}.$$

2.4 Formule de Green

Lemme 2.4.1 Si $U \in H^2(\Omega)$ et $v \in H^1(\Omega)$:

$$-\int_{\Omega} \Delta U(x, y)v(x, y) dx dy = \int_{\Omega} \nabla U(x, y)\nabla v(x, y) dx dy + \int_{\partial\Omega} \frac{\partial U}{\partial N}(x, y)v(x, y) d\sigma$$

avec $\frac{\partial U}{\partial N} = \nabla U \cdot N$, où N est un vecteur normal extérieur à $\partial\Omega$, $\frac{\partial U}{\partial N}$ est appelée la dérivée normale de U .

2.5 Formulation variationnelle d'un problème de valeurs aux limites

En mathématique, la formulation variationnelle d'un problème de valeurs aux limites correspond à une formulation faible de ce problème. Autrement dit la formulation variationnelle c'est une autre manière d'énoncer un problème aux limites. L'intérêt de cette approche est de pouvoir disposer de concepts et de propriétés de l'analyse fonctionnelle, en particulier ceux de l'espace de *Hilbert*¹ et de *Sobolev*.

Considérons Ω un ouvert borné de \mathbb{R}^n de bord $\partial\Omega$ supposé lipschitzien, alors trouvons une fonction U définie sur Ω solution du problème suivant :

$$\begin{cases} -\Delta U(x) + c(x)U(x) = f(x), & \forall x \in \Omega \\ U(x) = 0, & \forall x \in \partial\Omega \end{cases} \quad (2.5.1)$$

où, $c \in L^\infty(\Omega)$, $f \in L^2(\Omega)$. En outre on suppose que

$$\exists c_0 > 0 \text{ telle que } \forall x \in \Omega, c(x) \geq c_0 > 0.$$

On multiplie les deux membres de la première équation du (2.3.1) par une fonction régulière dite fonction test v et en intégrant sur le domaine Ω on trouve :

$$-\int_{\Omega} \Delta U(x)v(x) dx + \int_{\Omega} c(x)U(x)v(x) dx = \int_{\Omega} f(x)v(x) dx.$$

Maintenant, en utilisant la formule de *Green* on obtient :

$$\int_{\Omega} \nabla U(x)\nabla v(x) dx + \int_{\partial\Omega} \frac{\partial U(x)}{\partial N}v(x) d\tau + \int_{\Omega} c(x)U(x)v(x) dx = \int_{\Omega} f(x)v(x) dx,$$

le terme de bord du domaine va disparaître car on va prendre v nulle sur $\partial\Omega$ comme la solution U .

D'où le problème est réécrit de la manière suivante :

¹un espace de *Hilbert* est un espace préhilbertien complet.

Trouver une solution $U \in H_0^1(\Omega)$ qui satisfait

$$\int_{\Omega} \nabla U(x) \nabla v(x) dx + \int_{\Omega} c(x) U(x) v(x) dx = \int_{\Omega} f(x) v(x) dx, \quad \forall v \in H_0^1(\Omega). \quad (2.5.2)$$

L'équation (2.3.2) est appelée la formulation variationnelle ou la formulation faible du problème (2.3.1).

2.5.1 Résolution d'un problème variationnelle

Avant de passer à l'étape de résolution, la question qui se pose :

est ce que la solution de la formulation variationnelle existe ? Et si elle existe est elle unique ?

Définition 2.5.1 (forme bilinéaire) On dit que a est une forme bilinéaire sur $V \times V$ si :

1. a est défini de $V \times V$ dans \mathbb{R} ,
2. a est linéaire par rapport à chaque argument.

Définition 2.5.2 (forme bilinéaire continu) Une forme bilinéaire a est dite **continue** sur $V \times V$ s'il existe une constante positive M telle que :

$$|a(u, v)| \leq M \|u\| \|v\|, \quad \forall (u, v) \in V \times V.$$

Définition 2.5.3 (forme bilinéaire coercive) Une forme bilinéaire a est dite **coercive** sur $V \times V$ s'il existe un réel $\alpha > 0$ tel que pour tout $u \in V$,

$$a(u, u) \geq \alpha \|u\|^2.$$

Remarque 2.5.1 Une forme coercive est aussi dite V -elliptique.

Théorème 2.5.1 (Lax-Milgram) Soit V un espace de Hilbert sur \mathbb{R} et a une forme bilinéaire continue et V -elliptique. On considère L une forme linéaire sur V .

Alors il existe un unique $u \in V$ tel que

$$a(u, v) = L(v), \quad \forall v \in V.$$

De plus, on a l'estimation

$$\|u\|_V \leq \frac{1}{\alpha} \|L\|_{V'}$$

où V' est l'ensemble des formes linéaires sur V .

Le théorème de *Lax-Milgram* est l'un des fondements de la méthode des éléments finis. Ce théorème permet d'assurer de l'existence et de l'unicité de solution de la formulation variationnelle et donc du problème considéré.

En effet, posons

$$\begin{aligned}
 a(U, v) &= \int_{\Omega} \nabla U(x) \nabla v(x) dx + \int_{\Omega} c(x) U(x) v(x) dx \\
 L(v) &= \int_{\Omega} f(x) v(x) dx
 \end{aligned}$$

on a,

$H_0^1(\Omega)$: un espace de *Hilbert*,

$a(U, v)$: une forme bilinéaire continue (linéarité et la continuité de l'intégrale) et elle est coercive (c est positive),

$L(v)$: une forme linéaire continue sur $H_0^1(\Omega)$.

Sous ses hypothèses et d'après le théorème de *Lax-Milgram* on conclut que le problème (2.3.2) admet une unique solution $U \in H_0^1(\Omega)$ de plus $\Delta U \in L^2(\Omega)$.

Est ce que la solution du problème variationnel (2.3.2) est solution du problème (2.3.1) ?

Théorème 2.5.2 U est solution de (2.3.1) si et seulement si U est solution de (2.3.2).

Preuve. Si U est solution de (2.3.1) alors U est solution de (2.3.2) (évidente).

Si U est solution de (2.3.2) alors U est solution de (2.3.1)

En effet, supposons que U est solution de (2.3.2) et choisissons une fonction $v \in D(\Omega)$ (car $D(\Omega) \subset H_0^1(\Omega)$)

alors

$$\int_{\Omega} \nabla U(x) \nabla v(x) dx + \int_{\Omega} c(x) U(x) v(x) dx = \int_{\Omega} f(x) v(x) dx, \quad \forall v \in D(\Omega).$$

En appliquant la formule de *Green* inverse, on obtient :

$$- \int_{\Omega} \Delta U(x) v(x) dx + \int_{\Omega} c(x) U(x) v(x) dx = \int_{\Omega} f(x) v(x) dx, \quad \forall v \in D(\Omega).$$

Lemme 2.5.1 Soit $f \in L^2(\Omega)$, on a

$$\int_{\Omega} f(x) \varphi(x) dx = 0, \text{ pour tout } \varphi \in D(\Omega) \Rightarrow f = 0 \text{ presque partout dans } \Omega.$$

D'après ce lemme, on a

$$-\Delta U(x) + c(x) U(x) = f(x) \text{ pp dans } \Omega.$$

D'autre part on a $U \in H_0^1(\Omega)$ donc $U = 0$ sur la frontière de Ω .

Ainsi U est solution de (2.3.1). □

2.6 Ondelettes

La terminologie ondelette a été utilisée par *Morlet* en **1975**, bien que la première base d'ondelettes connue date de **1910**, créée par *Haar*. Il est intéressant de remarquer que les premiers travaux sur les ondelettes ont été réalisés dans le formalisme de la physique théorique par *Morlet* et *Grossman*, ou par des personnes issues de cette discipline, comme *I. Daubechies*. La transformée en ondelettes permet d'exprimer, sans perte les informations d'un signal dans une base dans laquelle on obtient la contribution locale de chaque fréquence au signal. *Mallat* et *Meyer* créèrent un environnement, appelé analyse multi-résolution (AMR), permettant une construction systématique de bases d'ondelettes, associées à une transformée discrète rapide.

Dans cette partie nous présentons le principe de l'analyse multirésolution, et on rappelle brièvement par les principales propriétés des ondelettes et les bases d'ondelettes dans le cas monodimensionnel et bidimensionnels.

Définition 2.6.1 Une ondelette est une petite onde qui a un début et une fin, c'est la signification physique. Mathématiquement, une ondelette est une fonction oscillante de valeurs réelles, de carré intégrable et de moyenne nulle i.e. ,

$$\int_{\mathbb{R}} \psi(t) dt = 0$$

et de norme 1

$$\|\psi\|_{L^2(\mathbb{R})}^2 = \left(\int_{\mathbb{R}} |\psi(t)|^2 dt \right)^{\frac{1}{2}} = 1.$$

ψ est aussi dite ondelette mère.

Définition 2.6.2 Les ondelettes filles sont obtenues par translation et dilatation / contraction de l'ondelette mère ψ et ont la forme :

$$\psi_{s,u}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right),$$

où

$u \in \mathbb{R}$: est la translation temporelle (position).

$s \in \mathbb{R}^*$: est l'échelle de dilatation ($s > 1$) / contraction ($s < 1$).

L'ondelette $\psi_{u,s}$ est normalisée de norme :

$$\|\psi_{s,u}\|_{L^2(\mathbb{R})}^2 = \|\psi\|_{L^2(\mathbb{R})}^2 = 1.$$

Définition 2.6.3 (Moments nuls d'une ondelette) On dit que l'ondelette ψ est à m moments nuls si

$$\int_{-\infty}^{+\infty} t^k \psi(t) dt = 0, \quad 0 \leq k < m,$$

cela revient à dire que ψ est orthogonale à tous les polynômes de degré inférieur à m .

Le nombre de moments nuls joue un rôle très important dans l'approximation et la compression des ondelettes : si l'ondelette a N moments nuls, alors ψ est soutenu sur $[-N + 1, N]$. En outre, la régularité de cette dernière fonction augmente asymptotiquement linéairement avec N

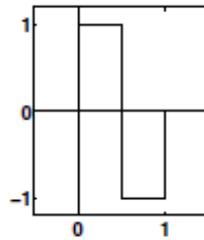
$$\psi \in C^{\mu N}$$

pour N suffisamment grand, $\mu = 0.2$.

Exemple 2.6.1 Ondelette de Haar

Nous définissons sur $\Omega = L^2([0, 1])$ la fonction suivante :

$$\psi(t) = \begin{cases} 1 & \text{si } 0 \leq t \leq \frac{1}{2} \\ -1 & \text{si } \frac{1}{2} \leq t < 1 \\ 0 & \text{sinon} \end{cases}$$



Ondelette de Haar

Exemple 2.6.2 Ondelette de Morlet

Elle est définie comme suit :

$$\psi(t) = \sqrt{2} \exp\left(\frac{-t^2}{\alpha^2}\right) \left[\exp(-\pi t) - \exp\left(-\frac{\pi^2 \alpha^2}{4}\right) \right].$$

2.6.1 Transformée en ondelette continue (TOC)

La transformée en ondelette continue d'une fonction $f \in L^2(\mathbb{R})$ en temps u et échelle s est définie par le produit scalaire avec l'ondelette $\psi_{u,s}$

$$\begin{aligned} Wf(u, s) &= \langle f, \psi_{u,s} \rangle \\ &= \int_{\mathbb{R}} f(t) \frac{1}{\sqrt{s}} \overline{\psi\left(\frac{t-u}{s}\right)} dt, \end{aligned}$$

où W désigne l'initial de wavelet (ondelette en anglais), $\overline{\psi\left(\frac{t-u}{s}\right)}$ est le complexe conjugué de $\psi\left(\frac{t-u}{s}\right)$.

La transformée en ondelette continue est une analyse temps échelle, elle permet d'avoir des transformations d'une fonction au voisinage du point u et à l'échelle s .

La TOC peut se réécrire comme un produit de convolution (voir Annexes) :

$$Wf(u, s) = \left(f * \check{\psi}_s \right) (u),$$

avec,

$$\check{\psi}_s(t) = \frac{1}{\sqrt{s}} \overline{\psi\left(\frac{-t}{s}\right)}.$$

2.6.2 Construction de la fonction originale

Pour pouvoir reconstruire la fonction originale f , l'ondelette ψ doit nécessairement vérifier la condition d'admissibilité suivante :

$$C_\psi = \int_0^\infty \left| \hat{\psi}(\xi) \right|^2 \frac{d\xi}{\xi} < +\infty, \quad (2.6.1)$$

où $\hat{\psi}(\xi)$ est la transformée de *Fourier* de ψ (voir Annexe). Cette reconstruction s'effectue par :

$$f(x) = \frac{1}{C_\psi} \int \int Wf(u, s) \psi_{s,u}(x) \frac{duds}{u^2}.$$

Remarque 2.6.1 *Si l'ondelette ψ vérifie la condition (2.4.1) elle est dite admissible.*

2.6.3 Transformée en ondelette discrète (TOD)

Notons par

$$s = s_0^m \quad \text{et} \quad u = nu_0 s_0^m \quad \text{avec} \quad s_0, u_0 \in \mathbb{Z}.$$

On définit la transformée en ondelette discrète par :

$$Wf(m, n) = s_0^{-\frac{m}{2}} \int_{\mathbb{R}} f(t) \psi(s_0^{-m}t - nu_0) dt, \quad (s_0, u_0) \in (\mathbb{R}_+^*)^2, \quad (m, n) \in \mathbb{Z}^2,$$

avec s_0^m l'échelle et $nu_0 s_0^m$ représente la translation.

On parlera de transformée en ondelettes dyadique dans le cas où $s_0 = 2$ et $u_0 = 1$

$$Wf(m, n) = 2^{-\frac{m}{2}} \int_{\mathbb{R}} f(t) \psi(2^{-m}t - n) dt,$$

le signal décomposé $f(x)$ est reconstruit par la formule suivante :

$$f(x) = \sum_j \sum_n Wf(j, n) \psi_{m,n}(x).$$

2.6.4 Base d'ondelettes

Définition 2.6.4 *Une base d'ondelettes est une base préhilbertienne de $L^2(\mathbb{R})$ de la forme $\psi_{j,k}$ obtenue par translation et dilatation dyadique de l'ondelette mère ψ :*

$$\psi_{j,k}(t) = 2^{-j/2} \psi(2^{-j}t - k), \quad j, k \in \mathbb{Z}.$$

Exemple 2.6.3 La base d'ondelettes de Haar est définie par :

$$\psi_{j,k}(t) = \begin{cases} 1 & \text{si } \frac{k}{2^j} \leq t < \frac{k}{2^j} + \frac{1}{2^{j+1}}, \\ -1 & \text{si } \frac{k}{2^j} + \frac{1}{2^{j+1}} \leq t < \frac{k+1}{2^j}, \\ 0 & \text{sinon.} \end{cases}$$

Elle est orthogonale.

Remarque 2.6.2 Ingrid Daubechies a cherchée à construire des bases d'ondelettes orthogonales à supports compacts et à m moments nuls.

Définition 2.6.5 Une base d'ondelettes $\psi_{m,k}(t)$ est dite orthonormale si

$$\begin{aligned} \langle \psi_{j,k}, \psi_{i,l} \rangle &= \int_{\mathbb{R}} \psi_{j,k}(x) \psi_{i,l}(x) dx \\ &= \delta(j-i) \delta(k-l), \end{aligned}$$

où δ est la matrice de Kronecker

$$\delta(j-i) = \begin{cases} 1 & \text{si } j = i \\ 0 & \text{si } j \neq i \end{cases}.$$

Définition 2.6.6 Une ondelette ψ est dite orthogonale si la famille $\{\psi_{j,k}, (j,k) \in \mathbb{Z}^2\}$ forme une base orthogonale de $L^2(\mathbb{R})$.

La construction des bases d'ondelettes repose sur la notion de l'analyse multi-résolution.

2.6.5 Analyse multi-résolution (AMR)

Les AMR (analyses multi-résolution) d'ondelettes, sont des espaces d'approximation générés par des bases de fonctions introduites par **Stéphan Mallat** en **1986**. Cette thématique permet de calculer des approximations successives d'une fonction à des résolutions spatiales de plus en plus grossières. L'approximation multi-résolution calcule l'approximation d'une fonction f à diverses résolutions par sa projections orthogonales sur une famille d'espaces $(V_j)_{j \in \mathbb{Z}}$.

Définition 2.6.7 Une analyse multi-résolution (AMR) de $L^2(\mathbb{R})$ est une suite de sous-espaces vectoriels (fermés) $(V_j)_{j \in \mathbb{Z}}$ de $L^2(\mathbb{R})$ vérifiant les conditions suivantes :

1. $(V_j)_{j \in \mathbb{Z}}$ constitue une suite d'espaces emboîtés : $\forall j \in \mathbb{Z}, V_j \subset V_{j+1}$.
2. $\lim_{j \rightarrow +\infty} V_j = \{0\}$, $\lim_{j \rightarrow -\infty} V_j$ est dense dans $L^2(\mathbb{R})$.
3. Une fonction $f(x)$ appartient à V_j si et seulement si sa dilatée $f(2x)$ appartient à V_{j+1} .
4. Il existe $\phi(x) \in L^2(\mathbb{R})$, appelée fonction d'échelle ou l'ondelette père telle que

$$\{\phi(x-k), k \in \mathbb{Z}\}$$

forme une base orthonormale de V_0 .

Remarque 2.6.3 - La première condition signifie que V_j est un sous espace de V_{j+1} et donc la projection d'une fonction sur V_j constitue une moins bonne approximation que sa projection sur V_{j+1} . Cette inclusion montre que l'approximation sur V_{j+1} contient toute l'information nécessaire pour calculer l'approximation sur V_j .

- Les secondes montre que quand $j \rightarrow +\infty$ on perd toutes les informations sur la fonction, alors que quand $j \rightarrow -\infty$ l'approximation est asymptotiquement égale au fonction d'origine
- De troisième condition, on peut passer de V_j à V_{j+1} en multipliant par le facteur 2 d'échelle.

Les espaces V_j sont appelés les espaces d'approximations et le complémentaire orthogonale de V_j dans V_{j+1} noté par W_j est appelé espace de détails et on écrit :

$$V_{j+1} = V_j \oplus W_j$$

où le symbole \oplus désigne la somme directe de sous espaces vectoriels.

Ainsi la projection orthogonale d'une fonction f sur V_{j+1} peut se décomposer comme la somme de sa projection orthogonale sur V_j et W_j . les familles $\left\{ \phi_{j,n}(t) = 2^{-\frac{j}{2}} \phi(2^{-j}t - n), n \in \mathbb{Z} \right\}$ et $\left\{ \psi_{j,n}(t) = 2^{-\frac{j}{2}} \psi(2^{-j}t - n), n \in \mathbb{Z} \right\}$ étant les bases orthonormales des espaces V_j et W_j respectivement.

La projection de f sur V_{j+1} est donnée par

$$P_{V_{j+1}}f(t) = P_{V_j}f(t) + P_{W_j}f(t).$$

où

$$\begin{aligned} P_{V_j}f(t) &= \sum_n a_j[n] 2^{-\frac{j}{2}} \phi(2^{-j}t - n), \\ P_{W_j}f(t) &= \sum_n c_j[n] 2^{-\frac{j}{2}} \psi(2^{-j}t - n), \end{aligned}$$

et

$$a_j[n] = \left\langle f(t), 2^{-\frac{j}{2}} \phi(2^{-j}t - n) \right\rangle, \quad c_j[n] = \left\langle f(t), 2^{-\frac{j}{2}} \psi(2^{-j}t - n) \right\rangle$$

les coefficients $a_j[n]$ et $c_j[n]$ sont appelés coefficients d'approximation et coefficients d'ondelettes ou de détails du fonction au niveau de résolution j .

2.6.6 Bases d'ondelettes bidimensionnelles

Analyse multi-résolution bidimensionnelles

Considérons $(V_j)_{j \in \mathbb{Z}}$ une analyse multi-résolution de $L^2(\mathbb{R})$, on définit l'analyse multi-résolution séparable de $L^2(\mathbb{R}^2)$ par $\mathbf{V}_j = V_j^2 = V_j \otimes V_j$. Si $\phi(x)$ est la fonction d'échelle associée à l'AMR $(V_j)_{j \in \mathbb{Z}}$ alors on peut donner une base orthonormale de $(\mathbf{V}_j)_{j \in \mathbb{Z}}$ par

$$\left\{ \phi_{n,m}(x, y) = 2^{-j} \phi(2^{-j}x - n) \phi(2^{-j}y - m), (n, m) \in \mathbb{Z}^2 \right\}.$$

Remarque 2.6.4 L'AMR dans le cas bidimensionnel a les mêmes propriétés que dans le cas monodimensionnel.

Suposons l'espace de détails \mathbf{W}_j est le complémentaire orthogonale de \mathbf{V}_{j+1} dans \mathbf{V}_j . Dans le cas $2D$, on a besoin de trois espaces de détails \mathbf{W}_j^h , \mathbf{W}_j^d , \mathbf{W}_j^v , de tels sorte que

$$\mathbf{V}_{j+1} = \mathbf{V}_j \oplus \mathbf{W}_j^h \oplus \mathbf{W}_j^d \oplus \mathbf{W}_j^v,$$

où \mathbf{W}_j^h est l'espace des détails horizontaux, \mathbf{W}_j^d est l'espace des détails diagonaux et \mathbf{W}_j^v est l'espace des détails verticaux.

Si $\psi(t)$ est l'ondelette mère associée à l'AMR $(V_j)_{j \in \mathbb{Z}}$ on pose les trois fonctions :

$$\begin{aligned}\psi^h(x, y) &= \phi(x)\psi(y), \\ \psi^d(x, y) &= \psi(x)\psi(y), \\ \psi^v(x, y) &= \psi(x)\phi(y),\end{aligned}$$

par translation et dilatation de ces bases, on obtient des bases orthonormales de 3 espaces de détails

$$\begin{aligned}\{2^{-j}\psi^h(2^{-j}x - n, 2^{-j}y - m), (n, m) \in \mathbb{Z}^2\} &\text{ base orthonormale de } \mathbf{W}_j^h, \\ \{2^{-j}\psi^v(2^{-j}x - n, 2^{-j}y - m), (n, m) \in \mathbb{Z}^2\} &\text{ base orthonormale de } \mathbf{W}_j^v, \\ \{2^{-j}\psi^d(2^{-j}x - n, 2^{-j}y - m), (n, m) \in \mathbb{Z}^2\} &\text{ base orthonormale de } \mathbf{W}_j^d,\end{aligned}$$

ces ondelettes extraient les détails d'une fonction à différentes échelles et orientations.

Exemple 2.6.4 *Considérons la base de Haar en 2D. La fonction d'échelle est la fonction caractéristique du carré unité :*

$$\Phi(x, y) = \chi_{[0,1]^2}(x, y).$$

Les images suivantes devraient suffire à comprendre la nature des ondelettes de Haar en 2-D et la fonction d'échelle :

1	1	-1	1	1	-1	-1	-1
$\Phi(x, y)$	$\psi^d(x, y)$	$\psi^h(x, y)$	$\psi^v(x, y)$				

Définition 2.6.8 *Une base d'ondelettes séparables de $L^2(\mathbb{R}^2)$ peut être associée à toute base d'ondelettes orthonormées de $L^2(\mathbb{R})$*

$$\psi_{k_1, k_2}^{j_1, j_2}(x, y) = \{\psi_{j_1, k_1}(x) \psi_{j_2, k_2}(y)\}_{j_1, j_2, k_1, k_2 \in \mathbb{Z}^4}. \quad (2.6.2)$$

Les fonctions $\psi_{j_1, k_1}(x) \psi_{j_2, k_2}(y)$ mélangent informations à deux échelles différentes 2^{j_1} , 2^{j_2} le long de x et y , ce qui souvent nous voulons éviter.

La méthode de collocation orthogonale sur les éléments finis

Ce chapitre traite la *méthode des éléments finis* qui est une technique d'approximation numérique de problèmes de valeurs aux limites.

Cette méthode est très utile dans le cas où la forme géométrique du domaine d'étude est **irrégulière**. Le principe de cette méthode est de discrétiser le problème en découpant le domaine considéré en un nombre fini de sous domaines de formes géométriques simples (segment, carré, rectangle, triangle ...) que l'on appelle éléments finis. Sur chacun de ces éléments, on applique l'une des méthodes des résidus pondérées présentées dans le chapitre précédent (section 1) afin d'obtenir des matrices élémentaires. Ensuite, on procède à un assemblage en une matrice globale.

Considérant le problème de valeurs aux limites suivant :

$$a(x, y)\Delta U(x, y) + b(x, y)\nabla U(x, y) + c(x, y)U(x, y) = f(x, y), \text{ sur } \Omega =]a, b[\times]c, d[, \quad (3.0.1)$$

avec les conditions aux limites :

$$\begin{cases} U(a, y) = g_1(y), & U(b, y) = g_2(y) \\ U(x, c) = g_3(x), & U(x, d) = g_4(x), \end{cases} \quad (3.0.2)$$

où, Ω est un ouvert de \mathbb{R}^2 ; a, c, f sont dans $C(\overline{\Omega}, \mathbb{R})$, b est dans $C(\overline{\Omega}, \mathbb{R}^2)$, g_1 et g_2 sont dans $C([c, d], \mathbb{R})$ et g_3 et g_4 sont dans $C([a, b], \mathbb{R})$.

3.1 Discrétisation du domaine Ω

3.1.1 Discrétisation par éléments finis

L'idée de base de la discrétisation par éléments finis est de découper le domaine d'étude Ω en un nombre fini de sous domaines $\Omega^{(i)}$ de dimensions finies. Ces derniers sont appelés des *éléments finis* de telle sorte qu'ils forment un recouvrement admissible du domaine Ω i. e. ;

$$\bigcup_i \Omega^{(i)} = \Omega,$$

et

$$\forall i, j \quad \Omega^{(i)} \cap \Omega^{(j)} = \begin{cases} \text{soit un point} \\ \text{soit tout un coté} \\ \emptyset. \end{cases}$$

Remarque 3.1.1 *L'avantage de la discrétisation par éléments finis est qu'on peut localiser un nœud, un élément sans aucun problème.*

Pour discrétiser $\bar{\Omega} = [a, b] \times [c, d]$ on procède de la façon suivante :

a- Discrétisation suivant l'axe des x

En choisissant le nombre des nœuds n_{x_e} , on peut calculer le pas de discrétisation h_{x_e} suivant l'axe des x par la formule suivante :

$$h_{x_e} = (b - a) / (n_{x_e} - 1),$$

et la valeur de l'abscisses $x_e^{(i)}$ est définie par :

$$x_e^{(i)} = a + (i - 1) h_{x_e}, \quad i = 1, \dots, n_{x_e},$$

comme il est montré dans la figure suivante

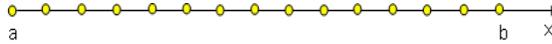


Figure 1 : Discretisation suivant l'axe des x

b- Discrétisation suivant l'axe des y

En choisissant le nombre des nœuds n_{y_e} , on peut calculer le pas de discrétisation h_{y_e} suivant l'axe des y par la formule suivante :

$$h_{y_e} = (d - c) / (n_{y_e} - 1),$$

et la valeur de l'abscisses $y_e^{(j)}$ est définie par :

$$y_e^{(j)} = c + (j - 1) h_{y_e}, \quad j = 1, \dots, n_{y_e},$$

comme il est montré dans la figure suivante

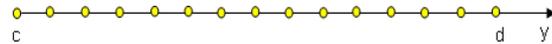


Figure 2 : Discretisation suivant l'axe des y

En arrangeant les deux discrétisations, on obtient un domaine maillé, où les points $(x_e^{(i)}, y_e^{(j)})$ obtenus sont appelés des **nœuds**, on les rassemble dans la matrice X_e suivante :

$$X_e = \begin{pmatrix} x_e^{(1)} & x_e^{(2)} & \dots & x_e^{(n_{x_e})} & y_e^{(1)} & \dots & y_e^{(n_{y_e})} \\ y_e^{(1)} & y_e^{(1)} & \dots & y_e^{(1)} & x_e^{(2)} & \dots & x_e^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ y_e^{(1)} & y_e^{(2)} & \dots & y_e^{(2)} & x_e^{(n_{x_e})} & \dots & x_e^{(n_{x_e})} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ y_e^{(1)} & y_e^{(n_{y_e})} & \dots & y_e^{(n_{y_e})} & x_e^{(1)} & \dots & x_e^{(1)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ y_e^{(1)} & y_e^{(n_{y_e})} & \dots & y_e^{(n_{y_e})} & x_e^{(n_{x_e})} & \dots & x_e^{(n_{x_e})} \end{pmatrix}, \quad (3.1.1)$$

de telle manière que chacun de ses colonnes représente les coordonnées d'un nœud

$$x_e^{(1)} = a, y_e^{(1)} = c, x_e^{(n_{x_e})} = b, y_e^{(n_{x_e})} = d.$$

Le choix de sens de lecture des nœuds est important pour les numériciens, le choix est dans le sens **lexicographique** i.e; de gauche à droite et de bas en haut.

L'algorithme suivant donne la numérotation des nœuds et ses coordonnées

```

pour j de 1 jusqu'à  $n_{y_e}$ 
  pour i de 1 jusqu'à  $n_{x_e}$ 
    nœud  $\leftarrow$  nœud+1
     $X_{e1, \text{nœud}} \leftarrow x_i$ 
     $X_{e2, \text{nœud}} \leftarrow y_i$ 
  fin pour
fin pour

```

On cherche maintenant à relier les nœuds pour obtenir des **éléments rectangulaire**. Tout élément rectangulaire est composé de quatre nœuds, voir la figure (??)

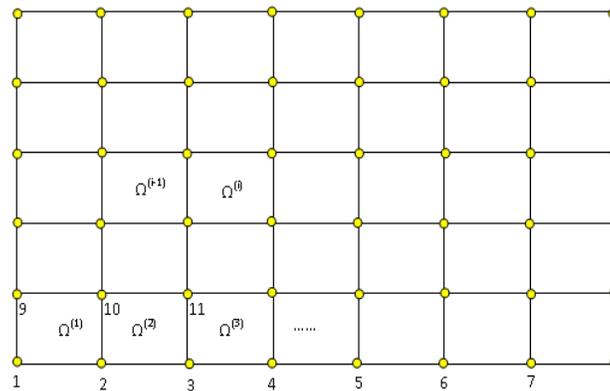


Figure 3 : Discrétisation en EF

Ainsi, on obtient $(n_{x_e} - 1)(n_{y_e} - 1)$ éléments rectangulaires.

L'algorithme suivant nous donne le numéro de chaque élément fini et le numéro des nœuds qui le composent :

```

elt  $\leftarrow$  0
pour j de 1 à  $n_{y_e}-1$ 
  pour i de 1 à  $n_{x_e}-1$ 
    elt  $\leftarrow$  elt+1
    elemfini $_{1, \text{elt}} \leftarrow (j-1) \times n_{x_e} + i$ 
    elemfini $_{2, \text{elt}} \leftarrow (j-1) \times n_{x_e} + i+1$ 
    elemfini $_{3, \text{elt}} \leftarrow (j-1) \times n_{x_e} + i+1+n_{x_e}$ 
    elemfini $_{4, \text{elt}} \leftarrow (j-1) \times n_{x_e} + i+n_{x_e}$ 
  fin pour
fin pour

```

Remarque 3.1.2

1. Le choix du pas constant dans la discrétisation est le plus facile à mettre en œuvre mais si la solution présente une instabilité dans certaine région du domaine, on raffine le maillage dans ses régions.
2. Le sens de numérotation des sommets dans un rectangle est toujours inverse à celui des aiguilles d'une montre.

Maintenant on va appliquer la méthode de collocation orthogonale sur chaque élément fini.

3.1.2 La méthode de collocation orthogonale

On s'intéresse à l'application de la méthode de collocation orthogonale sur chaque élément fini.

La méthode de collocation orthogonale consiste aux étapes suivantes :

1. Discrétisation d'un élément fini $\Omega^{(i)} = [x_e^{(i)}, x_e^{(i+1)}] \times [y_e^{(i)}, y_e^{(i+1)}]$.

Dans cette étape on choisit quelques points de collocation dans l'élément qu'on va obtenir par la suite une approximation de la solution en ces points.

Pour trouver les points de collocation, on procède de la même manière que pour la discrétisation par élément finis. Étant donné $(n_{x_c} + 2)$, (resp. $(n_{y_c} + 2)$) points de collocation qui sont représentés par les racines du polynômes de *Jacobi* que l'on note par x_c (resp. y_c) suivant l'axe des x (resp. l'axe des y), on calcule les pas de discrétisation :

$$\begin{aligned} h_{x_c}^{(k)} &= (x_c^{(k+1)} - x_c^{(k)}), \quad k = 1, \dots, n_{x_c} + 1, \\ h_{y_c}^{(l)} &= (y_c^{(l+1)} - y_c^{(l)}), \quad l = 1, \dots, n_{y_c} + 1. \end{aligned}$$

L'algorithme ci-dessus nous montre la numérotation des points de collocation et ses coordonnées, où la matrice X_c est conçue de telle sorte que chacune de ses colonnes représente les coordonnées d'un point de collocation

```

| k = 0;
| pour j de 1 à nyc + 2
|   pour i de 1 à nxc + 2 - (j - 1)
|     k → k + 1;
|     Xc1,k → xci
|     Xc2,k → ycj
|   fin pour
| fin pour

```

Par conséquent,

$$X_c = \begin{pmatrix} x_c^{(1)} & x_c^{(2)} & \dots & x_c^{(n_{x_c}+2)} & x_c^{(1)} & \dots & x_c^{(n_{x_c}+2)} & \dots & x_c^{(1)} & \dots & x_c^{(n_{x_c}+2)} \\ y_c^{(1)} & y_c^{(1)} & \dots & y_c^{(1)} & y_c^{(2)} & \dots & y_c^{(2)} & \dots & y_c^{(n_{y_c}+2)} & \dots & y_c^{(n_{y_c}+2)} \end{pmatrix} \quad (3.1.2)$$

avec, $x_c^{(1)} = x_e^{(i)}$, $y_c^{(n_{y_c}+2)} = x_e^{(i+1)}$, $y_c^{(1)} = y_e^{(i+1)}$, $y_c^{(n_{y_c}+2)} = y_e^{(i+1)}$.

Voir la figure (??)

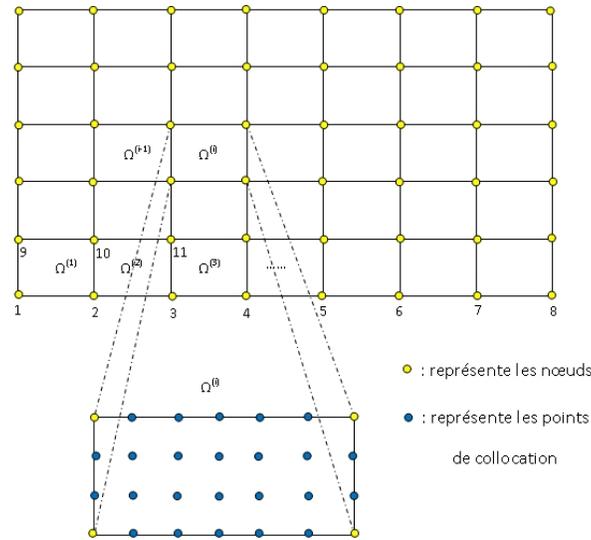


Figure 4 : Discrétisation du domaine $[0, 1] \times [0, 1]$ par MCOEF

1. 2. Discrétisation de équation (3.0.1)

Dans cette étape, on va écrire cette équation en chaque point de X_c de la discrétisation.

Choissant une approximation lagrangienne en $2D$ de la solution du problème (3.0.1)-(3.0.2) sur un élément $\Omega^{(s)} = [x_e^{(s)}, x_e^{(s+1)}] \times [y_e^{(s)}, y_e^{(s+1)}]$ de la forme suivante :

$$U^{(s)}(x, y) = \sum_{i=1}^{n_{x_c}+2} \sum_{j=1}^{n_{y_c}+2} U_{i,j}^{(s)} L_{ij}(x, y); \quad (3.1.3)$$

où, $L_{ij}(x, y) = l_i(x)l_j(y)$ est le polynôme de *Lagrange* voir ([10]) et $U_{i,j}^{(s)}$ sont les paramètres d'approximation.

Après la substitution de (3.2.1) dans l'équation (3.0.1), on obtient un résidus non nul $\mathcal{R}(x, y)$:

$$\mathcal{R}(x, y) = a(x, y)\Delta U^{(s)}(x, y) + b(x, y)\nabla U^{(s)}(x, y) + c(x, y)U^{(s)}(x, y) - f^{(s)}(x, y) \neq 0. \quad (3.1.4)$$

Remarque 3.1.3 Pour des raisons techniques, on est amené à transformer le développement de calcul sur un élément de référence.

En effet, tout intervalle (a, b) de \mathbb{R} peut être transformé en un intervalle de référence $(0, 1)$ grâce à une transformation linéaire :

$$\begin{aligned} Tr : (a, b) &\rightarrow (0, 1) \\ x &\mapsto \frac{x-a}{b-a}. \end{aligned} \quad (3.1.5)$$

De même pour le cas de deux dimensions, chaque pavé $[a, b] \times [c, d]$ peut être transformé en un élément de référence $[0, 1] \times [0, 1]$.

Pour la transformation d'un élément fini $\Omega^{(s)}$ en un élément de référence $[0, 1] \times [0, 1]$ on note :

$$(\mathbf{x}, \mathbf{y}) = \left(\frac{x - x_e^{(s)}}{h_{x_e}}, \frac{y - y_e^{(s)}}{h_{y_e}} \right),$$

avec $h_{x_e} = x_e^{(s+1)} - x_e^{(s)}$, $h_{y_e} = y_e^{(s+1)} - y_e^{(s)}$.

Donc

$$\Delta U^{(s)}(x, y) = \frac{1}{h_{x_e}^2} \frac{\partial^2 U^{(s)}}{\partial \mathbf{x}^2}(\mathbf{x}, \mathbf{y}) + \frac{1}{h_{y_e}^2} \frac{\partial^2 U^{(s)}}{\partial \mathbf{y}^2}(\mathbf{x}, \mathbf{y}) \quad (3.1.6)$$

$$\nabla U^{(s)}(x, y) = \left(\frac{1}{h_{x_e}} \frac{\partial U^{(s)}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{y}), \frac{1}{h_{y_e}} \frac{\partial U^{(s)}}{\partial \mathbf{y}}(\mathbf{x}, \mathbf{y}) \right). \quad (3.1.7)$$

Le résidu transformé sur l'élément de référence devient :

$$\begin{aligned} & \mathcal{R}(\mathbf{x}, \mathbf{y}) \\ = & a(\mathbf{x}, \mathbf{y}) \frac{1}{h_{x_e}^2} \frac{\partial^2 U^{(s)}}{\partial \mathbf{x}^2}(\mathbf{x}, \mathbf{y}) + \frac{1}{h_{y_e}^2} \frac{\partial^2 U^{(s)}}{\partial \mathbf{y}^2}(\mathbf{x}, \mathbf{y}) + b(\mathbf{x}, \mathbf{y}) \left(\frac{1}{h_{x_e}} \frac{\partial U^{(s)}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{y}), \frac{1}{h_{y_e}} \frac{\partial U^{(s)}}{\partial \mathbf{y}}(\mathbf{x}, \mathbf{y}) \right)^T \\ & + c(\mathbf{x}, \mathbf{y}) U^{(s)}(\mathbf{x}, \mathbf{y}) - f^{(s)}(\mathbf{x}, \mathbf{y}) \\ = & a(\mathbf{x}, \mathbf{y}) \frac{1}{h_{x_e}^2} \frac{\partial^2 U^{(s)}}{\partial \mathbf{x}^2}(\mathbf{x}, \mathbf{y}) + \frac{1}{h_{y_e}^2} \frac{\partial^2 U^{(s)}}{\partial \mathbf{y}^2}(\mathbf{x}, \mathbf{y}) + \frac{1}{h_{x_e}} b1(\mathbf{x}, \mathbf{y}) \frac{\partial U^{(s)}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{y}) + \\ & \frac{1}{h_{y_e}} b2(\mathbf{x}, \mathbf{y}) \frac{\partial U^{(s)}}{\partial \mathbf{y}}(\mathbf{x}, \mathbf{y}) + c(\mathbf{x}, \mathbf{y}) U^{(s)}(\mathbf{x}, \mathbf{y}) - f^{(s)}(\mathbf{x}, \mathbf{y}), \end{aligned}$$

avec

$$(b1(\mathbf{x}, \mathbf{y}), b2(\mathbf{x}, \mathbf{y})) = b(\mathbf{x}, \mathbf{y}).$$

Pour des raisons de simplification, on prend $n_{x_c} = n_{y_c} = n_c$.

Le principe de la méthode consiste à minimiser le résidu $\mathcal{R}(\mathbf{x}, \mathbf{y})$ au sens des distributions en le pondérant par les fonctions

$$\psi_{kl}(\mathbf{x}, \mathbf{y}) = \begin{cases} 0 & \text{si } (\mathbf{x}, \mathbf{y}) \neq (\mathbf{x}_k, \mathbf{y}_l) \\ 1 & \text{si } (\mathbf{x}, \mathbf{y}) = (\mathbf{x}_k, \mathbf{y}_l) \end{cases}.$$

On a donc

$$\iint_{[0,1] \times [0,1]} \mathcal{R}(\mathbf{x}, \mathbf{y}) \psi_{kl}(\mathbf{x}, \mathbf{y}) dx dy = 0,$$

ce qui implique

$$\mathcal{R}(\mathbf{x}_k, \mathbf{y}_l) = 0, \quad (3.1.8)$$

avec $k = 1, \dots, n_c + 2$; $l = 1, \dots, n_c + 2$.

Dans toute la suite on utilise la notation suivante :

$$h(\mathbf{x}_k, \mathbf{y}_l) = h_{k,l}.$$

Alors, l'équation (3.2.5) devient pour tout $k, l = 1, \dots, n_c + 2$:

$$\begin{aligned} \mathcal{R}(\mathbf{x}_k, \mathbf{y}_l) &= a_{k,l} \left(\sum_{i=1}^{n_c+2} \sum_{j=1}^{n_c+2} \frac{1}{h_{x_e}^2} U_{i,j}^{(s)} B_{x_{k,i}}^{(s)} M_{y_{l,j}}^{(s)} + \sum_{i=1}^{n_c+2} \sum_{j=1}^{n_c+2} \frac{1}{h_{y_e}^2} U_{i,j}^{(s)} M_{x_{k,i}}^{(s)} B_{y_{l,j}}^{(s)} \right) \quad (3.1.9) \\ &+ b_{1k,l} \sum_{i=1}^{n_c+2} \sum_{j=1}^{n_c+2} \frac{1}{h_{x_e}} U_{i,j}^{(s)} A_{x_{k,i}}^{(s)} M_{y_{l,j}}^{(s)} + b_{2kl} \sum_{i=1}^{n_c+2} \sum_{j=1}^{n_c+2} \frac{1}{h_{y_e}} U_{i,j}^{(s)} M_{x_{k,i}}^{(s)} A_{y_{l,j}}^{(s)} \\ &+ c_{k,l} \sum_{i=1}^{n_c+2} \sum_{j=1}^{n_c+2} U_{i,j}^{(s)} M_{x_{k,i}}^{(s)} M_{y_{l,j}}^{(s)} - f_{k,l}^{(s)} \\ &= 0, \end{aligned}$$

avec

$$\begin{aligned} A_{x_{j,i}}^{(s)} &= \frac{dl_i}{d\mathbf{x}}(\mathbf{x}_j), & A_{y_{j,i}}^{(s)} &= \frac{dl_i}{d\mathbf{y}}(\mathbf{y}_j), & M_{x_{k,i}}^{(s)} &= l_i(\mathbf{x}_k), \\ B_{x_{j,i}}^{(s)} &= \frac{d^2 l_i}{d\mathbf{x}^2}(\mathbf{x}_j), & B_{y_{j,i}}^{(s)} &= \frac{d^2 l_i}{d\mathbf{y}^2}(\mathbf{y}_j), & M_{y_{l,j}}^{(s)} &= l_j(\mathbf{y}_l). \end{aligned}$$

Pour les expressions de $A_{x_{k,i}}^{(s)}$, $A_{y_{l,j}}^{(s)}$, $B_{x_{k,i}}^{(s)}$, $B_{y_{l,j}}^{(s)}$, $M_{x_{k,i}}^{(s)}$ et $M_{y_{l,j}}^{(s)}$ voir annexe. Par conséquent l'équation (??) devient :

$$\begin{aligned} &\sum_{i=1}^{n_c+2} \left[a_{k,l} \left(\frac{1}{h_{x_e}^2} B_{x_{k,i}}^{(s)} M_{y_{l,1}}^{(s)} + \frac{1}{h_{y_e}^2} M_{x_{k,i}}^{(s)} B_{y_{l,1}}^{(s)} \right) \right] U_{i,1}^{(s)} \\ &+ \sum_{i=1}^{n_c+2} \left[\frac{b_{1k,l}}{h_{x_e}} A_{x_{k,i}}^{(s)} M_{y_{l,1}}^{(s)} + \frac{b_{2k,l}}{h_{y_e}} M_{x_{k,i}}^{(s)} A_{y_{l,1}}^{(s)} + c_{k,l} \left(M_{x_{k,i}}^{(s)} M_{y_{l,1}}^{(s)} \right) \right] U_{i,1}^{(s)} \\ &+ \sum_{i=1}^{n_c+2} \left[a_{k,l} \left(\frac{1}{h_{x_e}^2} B_{x_{k,i}}^{(s)} M_{y_{l,2}}^{(s)} + \frac{1}{h_{y_e}^2} M_{x_{k,i}}^{(s)} B_{y_{l,2}}^{(s)} \right) \right] U_{i,2}^{(s)} \\ &+ \sum_{i=1}^{n_c+2} \left[\frac{b_{1k,l}}{h_{x_e}} A_{x_{k,i}}^{(s)} M_{y_{l,2}}^{(s)} + \frac{b_{2k,l}}{h_{y_e}} M_{x_{k,i}}^{(s)} A_{y_{l,2}}^{(s)} + c_{k,l} \left(M_{x_{k,i}}^{(s)} M_{y_{l,2}}^{(s)} \right) \right] U_{i,2}^{(s)} \\ &\vdots \\ &+ \sum_{i=1}^{n_c+2} \left[a_{k,l} \left(\frac{1}{h_{x_e}^2} B_{x_{k,i}}^{(s)} M_{y_{l,n_c+2}}^{(s)} + \frac{1}{h_{y_e}^2} M_{x_{k,i}}^{(s)} B_{y_{l,n_c+2}}^{(s)} \right) \right] U_{i,n_c+2}^{(s)} \\ &+ \sum_{i=1}^{n_c+2} \left[\frac{b_{1k,l}}{h_{x_e}} A_{x_{k,i}}^{(s)} M_{y_{l,n_c+2}}^{(s)} + \frac{b_{2k,l}}{h_{y_e}} M_{x_{k,i}}^{(s)} A_{y_{l,n_c+2}}^{(s)} + c_{k,l} M_{x_{k,i}}^{(s)} M_{y_{l,n_c+2}}^{(s)} \right] U_{i,n_c+2}^{(s)} \\ &= f_{k,l}^{(s)}, \end{aligned}$$

avec, $l, k = 1, \dots, n_c + 2$.

Donc on peut le réécrire sous la forme compacte suivante

$$\sum_{i,j=1}^{n_c+2} N_{i,j}^{(s)} U_{ij}^{(s)} = f_{k,l}, \quad (3.1.10)$$

avec

$$N_{i,j}^{(s)} = \left[a_{k,l} \left(\frac{1}{h_{x_e}^2} B_{x_{k,i}}^{(s)} M_{y_{l,j}}^{(s)} + \frac{1}{h_{y_e}^2} M_{x_{k,i}}^{(s)} B_{y_{l,j}}^{(s)} \right) + \frac{b1_{k,l}}{h_{x_e}} A_{x_{k,i}}^{(s)} M_{y_{l,j}}^{(s)} + \frac{b2_{k,l}}{h_{y_e}} M_{x_{k,i}}^{(s)} A_{y_{l,j}}^{(s)} + c_{k,l} M_{x_{k,i}}^{(s)} M_{y_{l,j}}^{(s)} \right].$$

Ainsi l'équation (??) devient

$$\begin{aligned} & N_{1,1}^{(s)} U_{1,1} + N_{2,1}^{(s)} U_{2,1} + N_{3,1}^{(s)} U_{3,1} + \dots + N_{n_c+2,1}^{(s)} U_{n_c+2,1} \\ & + N_{1,2}^{(s)} U_{1,2} + N_{2,2}^{(s)} U_{2,2} + N_{3,2}^{(s)} U_{3,2} + \dots + N_{n_c+2,2}^{(s)} U_{n_c+2,2} \\ & + \dots \\ & + N_{1,n_c+2}^{(s)} U_{1,n_c+2} + N_{2,n_c+2}^{(s)} U_{2,n_c+2} + N_{3,n_c+2}^{(s)} U_{3,n_c+2} + \dots + N_{n_c+2,n_c+2}^{(s)} U_{n_c+2,n_c+2} \\ & = f_{k,l}^{(s)}. \end{aligned}$$

Par conséquent, on obtient un système de $(n_c + 2)^2$ équations linéaires associées à $(n_c + 2)^2$ inconnues dans chacun des $(n_{x_e} - 1)(n_{y_e} - 1)$ éléments finis, qu'on écrit directement sous forme matricielle :

$$AL^{(s)}U^{(s)} = BL^{(s)}.$$

3.1.3 Le calcul de la matrice locale

Après avoir discrétisé l'équation sur un élément de référence, on recherche maintenant à évaluer l'écriture matricielle du système, on définit la matrice locale $AL^{(s)}$, et le second membre local $BL^{(s)}$ dans l'élément fini $\Omega^{(s)} = [0, 1] \times [0, 1]$ par :

$$AL_{(l-1)(n_c+2)+k, (j-1)(n_c+2)+i}^{(s)} = \left[\begin{array}{c} a_{k,l} \left(\frac{1}{h_{x_e}^2} B_{x_{k,i}}^{(s)} M_{y_{l,j}}^{(s)} + \frac{1}{h_{y_e}^2} M_{x_{k,i}}^{(s)} B_{y_{l,j}}^{(s)} \right) + \frac{1}{h_{x_e}} b1_{k,l} A_{x_{k,i}}^{(s)} M_{y_{l,j}}^{(s)} \\ + \frac{1}{h_{y_e}} b2_{k,l} M_{x_{k,i}}^{(s)} A_{y_{l,j}}^{(s)} + c_{k,l} M_{x_{k,i}}^{(s)} M_{y_{l,j}}^{(s)}, \end{array} \right]^T,$$

avec $k, l = 1, \dots, n_c + 2$;

$$BL_{(l-1)(n_c+2)+k,l}^{(s)} = f_{k,l},$$

et l'inconnu

$$U^{(s)} = \begin{pmatrix} U_{1,1}^{(s)} \\ \vdots \\ U_{n_c+2,1}^{(s)} \\ U_{1,2}^{(s)} \\ \vdots \\ U_{n_c+2,2}^{(s)} \\ \vdots \\ U_{1,n_c+2}^{(s)} \\ \vdots \\ U_{n_c+2,n_c+2}^{(s)} \end{pmatrix}.$$

La matrice obtenue $AL^{(s)}$ est de dimension $(n_c + 2) \times (n_c + 2)$. Elle est appelée matrice locale et le vecteur $BL^{(s)}$ est appelé second membre locale dans l'élément numéro s .

3.2 L'assemblage de matrices locales

Dans cette étape on veut assembler toutes les matrices locales obtenues sur chaque élément, en une matrice globale.

Cette procédure se réalise de la façon suivante :

```

pour m de 1 jusqu'à elt
  pour i de 1 jusqu'à nel
    pour j de 1 jusqu'à nel
      AG(element(i, m), element(j, m)) → AG(element(i, m), element(j, m)) + AL(i, j)
    fin pour
      BG(element(i, m)) → BG(element(i, m)) + BL(i)
    fin pour
  fin pour
fin pour

```

où elt désigne le nombre total des éléments finis, nel est le nombre de points de collocation dans chaque élément fini, AG est la matrice globale et BG est le second membre globale.

3.3 Les conditions aux limites

Une fois la matrice et le second membre locaux obtenus, on a besoin d'imposer les conditions aux limites.

L'algorithme suivant nous permet d'imposer les conditions aux limites :

```

1. Initialisation ind et val_impose deux vecteurs de dimension  $p$  à zéro
2. conditions aux limites en  $x = a$  et  $y$  quelconque
 $k \rightarrow 0$ 
pour  $i$  de 1 pas de  $(n_{x_e} - 1)(n_{x_c} + 1) + 1$  jusqu'à  $p$ 
     $k \rightarrow k + 1$ 
    ind( $i$ )  $\rightarrow 1$ 
    val_impose( $i$ )  $\rightarrow u_d(k)$ 
fin pour
3. pour  $i$  de 1 jusqu'à  $p$ 
    si ind( $i$ )=1
        pour  $j$  de 1 jusqu'à  $p$ 
            si  $i \neq j$ 
                 $BG(j) \rightarrow BG(j) - AG(j, i)val\_impose(i)$ 
                 $AG(i, j) \rightarrow 0$ 
            fin si
        fin pour
        pour  $k$  de 1 jusqu'à  $p$ 
             $AG(i, k) \rightarrow 0$ 
        fin pour
         $AG(i, i) \rightarrow 0$ 
         $BG(i) \rightarrow val\_impose(i)$ 
    fin si
fin pour

```

où

p : nombre de points de collocation et de nœuds dans le domaine d'étude,

n_{x_c} : nombre de points de collocation intérieur suivant l'axe des x ,

n_{y_c} : nombre de points de collocation intérieur suivant l'axe des y ,

ind : un indicateur pour détecter le noeud ou on doit imposer une valeur,

val_impose : un vecteur pour la valeur à imposé,

AG : matrice globale,

BG : second membre globale.

3.4 La résolution du système

Après avoir imposé les conditions aux limites, on obtient le système linéaire suivant :

$$(AG)U = BG. \quad (3.4.1)$$

Pour résoudre ce système on utilise la méthode **LU** (voir Annexe).

Une méthode de représentation des opérateurs différentiels en terme d'ondelettes

Considérons une analyse multi-résolution en $2D$

$$\dots \subset \mathbf{V}_j = V_j \otimes V_j \subset \mathbf{V}_{j+1} \subset \dots, j \in \mathbb{Z},$$

avec les bases associées au espaces \mathbf{V}_j , \mathbf{W}_j^h , \mathbf{W}_j^v et \mathbf{W}_j^d sont respectivement

$$\begin{aligned} \Phi_{n_1, n_2}^j(x, y) &= 2^{-j} \phi(2^{-j}x - n_1) \phi(2^{-j}y - n_2), \\ \Psi_{n_1, n_2}^{j, h}(x, y) &= 2^{-j} \phi(2^{-j}x - n_1) \psi(2^{-j}y - n_2), \\ \Psi_{n_1, n_2}^{j, v}(x, y) &= 2^{-j} \psi(2^{-j}x - n_1) \phi(2^{-j}y - n_2), \\ \Psi_{n_1, n_2}^{j, d}(x, y) &= 2^{-j} \psi(2^{-j}x - n_1) \psi(2^{-j}y - n_2), \end{aligned}$$

où $\phi(x)$ et $\psi(x)$ sont la fonction échelle et l'ondelette mère associées à l'AMR $(V_j)_{j \in \mathbb{Z}}$.

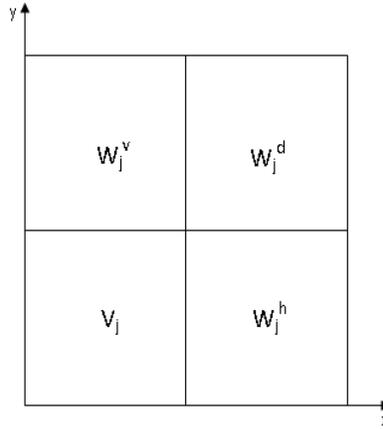
L'analyse multi-résolution d'une fonction f de $L^2(\mathbb{R}^2)$ consiste à calculer les projections orthogonales sur les espaces \mathbf{V}_j , \mathbf{W}_j^h , \mathbf{W}_j^v et \mathbf{W}_j^d comme suit :

$$\begin{aligned} P_{\mathbf{V}_j} f(x, y) &= \sum_{n_1=-\infty}^{+\infty} \sum_{n_2=-\infty}^{+\infty} s^j [n_1, n_2] \Phi_{n_1, n_2}^j(x, y), \\ P_{\mathbf{W}_j^h} f(x, y) &= \sum_{n_1=-\infty}^{+\infty} \sum_{n_2=-\infty}^{+\infty} c_j^h [n_1, n_2] \Psi_{n_1, n_2}^{j, h}(x, y), \\ P_{\mathbf{W}_j^v} f(x, y) &= \sum_{n_1=-\infty}^{+\infty} \sum_{n_2=-\infty}^{+\infty} c_j^v [n_1, n_2] \Psi_{n_1, n_2}^{j, v}(x, y), \\ P_{\mathbf{W}_j^d} f(x, y) &= \sum_{n_1=-\infty}^{+\infty} \sum_{n_2=-\infty}^{+\infty} c_j^d [n_1, n_2] \Psi_{n_1, n_2}^{j, d}(x, y), \end{aligned}$$

avec

$$\begin{aligned} s^j [n_1, n_2] &= \langle f(x, y), \Phi_{n_1, n_2}^j(x, y) \rangle, \\ c_j^h [n_1, n_2] &= \langle f(x, y), \Psi_{n_1, n_2}^{j, h}(x, y) \rangle, \\ c_j^v [n_1, n_2] &= \langle f(x, y), \Psi_{n_1, n_2}^{j, v}(x, y) \rangle, \\ c_j^d [n_1, n_2] &= \langle f(x, y), \Psi_{n_1, n_2}^{j, d}(x, y) \rangle. \end{aligned}$$

Dans la figure ci-dessous les zones \mathbf{V}_j , \mathbf{W}_j^h , \mathbf{W}_j^v et \mathbf{W}_j^d correspondent respectivement aux coefficients d'approximations, de détails horizontaux, verticaux et diagonaux au niveau de résolution j



En choisissant \mathbf{V}_n et \mathbf{V}_{n-J} comme étant le plus fin et le plus grossier sous espace, respectivement, et utilisant $\mathbf{V}_j = \mathbf{V}_{j-1} \oplus \mathbf{W}_{j-1}$, on obtient :

$$\mathbf{V}_n = \mathbf{V}_{n-J} \oplus \mathbf{W}_{n-J} \oplus \mathbf{W}_{n-J+1} \oplus \dots \oplus \mathbf{W}_{n-1}.$$

Alors l'approximation de f sur \mathbf{V}_n peut être étendue comme suit :

$$P_{\mathbf{V}_n} f(x, y) = P_{\mathbf{V}_{n-J}} f(x, y) + \sum_{j=n-J}^{n-1} Q_j f(x, y),$$

avec

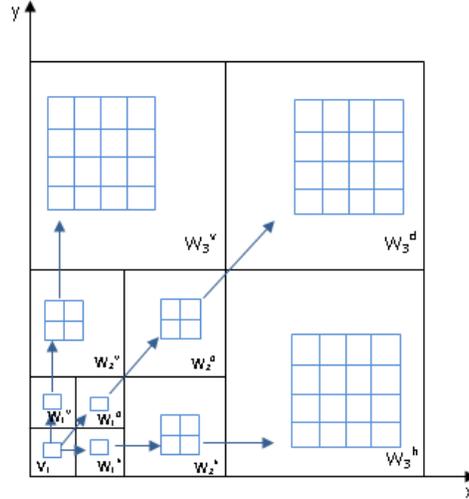
$$Q_j f(x, y) = Q_j^h f(x, y) + Q_j^v f(x, y) + Q_j^d f(x, y),$$

où

$$Q_j^h = P_{\mathbf{W}_j^h}, \quad Q_j^v = P_{\mathbf{W}_j^v} f(x, y), \quad Q_j^d = P_{\mathbf{W}_j^d} f(x, y).$$

Ici, $P_{\mathbf{V}_{n-J}} f$ est la plus grossière approximation sur \mathbf{V}_{n-J} et les projections Q_j sur \mathbf{W}_j donnent les détails $Q_j f$ sur \mathbf{W}_j construits par l'approximation la plus fine, $P_{\mathbf{V}_n} f$, de f sur \mathbf{V}_n .

La décomposition est du type de celle présentée dans la figure



Une dcomposition sur 3 niveaux de resolution

4.1 La représentation non-standard des opérateurs différentiels

Soit T un opérateur donné. On approxime T par $T_n = P_{V_n} T P_{V_n}$ et avec V_n et V_{n-J} ci-dessus, on étend T_n par :

$$T_n = P_{V_n} T P_{V_n} = \sum_{j=0}^{J-1} [P_{V_{n-j}} T P_{V_{n-j}} - P_{V_{n-j-1}} T P_{V_{n-j-1}}] + P_{V_{n-J}} T P_{V_{n-J}},$$

et on utilise

$$P_{V_{j+1}} = P_{V_j} + Q_j,$$

pour obtenir la représentation non-standard de T ,

$$T_n = \sum_{j=n-J}^{n-1} [A_j + B_j + C_j] + T_{n-J},$$

où $T_{n-J} = P_{V_{n-J}} T P_{V_{n-J}} : V_{n-J} \rightarrow V_{n-J}$ et

$$A_j = \sum_{\lambda, \lambda' = h, v, d} A_j^{\lambda, \lambda'} = \sum_{\lambda, \lambda' = h, v, d} Q_j^\lambda T Q_j^{\lambda'} : W_j^{\lambda'} \rightarrow W_j^\lambda,$$

$$B_j = \sum_{\lambda = h, v, d} B_j^\lambda = \sum_{\lambda = h, v, d} Q_j^\lambda T P_{V_j} : V_j \rightarrow W_j,$$

$$C_j = \sum_{\lambda=h,v,d} C_j^\lambda = \sum_{\lambda=h,v,d} P_{\mathbf{V}_j} T Q_j^\lambda : \mathbf{W}_j \rightarrow \mathbf{V}_j.$$

La représentation NS découple les échelles par l'action sur \mathbf{V}_j et \mathbf{W}_j à chaque niveau. Alors pour $f \in L^2(\mathbb{R}^2)$, on a l'approximation :

$$Tf \approx \sum_{j=n-J}^{n-1} \left[\sum_{\lambda,\lambda'=h,v,d} A_j^{\lambda,\lambda'} f + \sum_{\lambda=h,v,d} (B_j^\lambda + C_j^\lambda) f \right] + T_{n-J} f.$$

Les étapes dans l'application d'un opérateur T sur des fonctions périodiques $f \in L^2([0, 1]^2)$ de période 1 sont les suivantes :

- Approximer f dans \mathbf{V}_n par $P_{\mathbf{V}_n} f$ pour avoir la matrice s^n de même rang ($2^n \times 2^n$).
- Décomposer s^n sur les espaces d'AMR en utilisant la transformée en ondelette rapide en $2D$ pour obtenir c_j^h, c_j^v, c_j^d et s^j pour $j = n-1, n-2, \dots, n-J$.
- Pour $n-J \leq j \leq n-1$, construire les matrices $A_j^{\lambda,\lambda'}, B_j^{\lambda,\lambda'}, C_j^{\lambda,\lambda'}$ représentant les opérateurs $A_j^{\lambda,\lambda'}, B_j^\lambda, C_j^\lambda$, respectivement.
- L'approximation $Tf \approx T_n f$ est donnée par :

$$T_n f = \sum_{j=n-J}^{n-1} \left[\sum_{\lambda=h,v,d} \left(\sum_{n_3, n_4=0}^{2^j-1} \tilde{c}_{n_3, n_4}^{j, \lambda} \Psi_{n_3, n_4}^{j, \lambda} \right) + \sum_{n_3, n_4=0}^{2^j-1} \tilde{s}^j \Phi_{n_3, n_4}^{j, \lambda} \right],$$

où,

$$\begin{aligned} \tilde{c}^{j,h} &= \sum_{\lambda=h,v,d} A_j^{j,h,\lambda} \odot c_j^\lambda + B_j^{j,h} \odot s^j, \\ \tilde{c}^{j,v} &= \sum_{\lambda=h,v,d} A_j^{j,v,\lambda} \odot c_j^\lambda + B_j^{j,v} \odot s^j, \\ \tilde{c}^{j,d} &= \sum_{\lambda=h,v,d} A_j^{j,d,\lambda} \odot c_j^\lambda + B_j^{j,d} \odot s^j, \\ \tilde{s}^j &= \sum_{\lambda=h,v,d} C_j^{j,\lambda} \odot c_j^\lambda, \text{ pour } n-J+1 \leq j \leq n-1, \\ \tilde{s}^{n-j} &= \sum_{\lambda=h,v,d} C^{n-j,\lambda} \odot c_{n-j}^\lambda + T^{n-J} \odot s^{n-J}. \end{aligned}$$

L'opération \odot est définie comme suit :

Définition 4.1.1 Si P est une matrice bloc ($k \times k$) avec des matrices blocs ($k \times k$) notée $P^{i,j}$ et D est une matrice ($k \times k$), alors

$$(P \odot D)_{i,j} = \sum_{m,n} P_{m,n}^{i,j} D_{m,n}.$$

La fonction $T_n f$ peut être reconstruite dans l'espace \mathbf{V}_n comme suit :

$$T_n f = \sum_{n_1=0}^{2^n-1} \sum_{n_2=0}^{2^n-1} s_{n_1, n_2}^n \Phi_{n_1, n_2}^n.$$

4.2 Construction de la matrice correspondante à la représentation NS

L'action de l'opérateur T_n sur la fonction $f \in L^2([0, 1]^2)$,

$$\begin{aligned} T_n f & : = P_{V_n} T P_{V_n} f = P_{V_n} T \left(\sum_{n_1, n_2} \langle f, \Phi_{n_1, n_2}^n \rangle \Phi_{n_1, n_2}^n \right), \\ & = \sum_{n_3, n_4} \underbrace{\left(\sum_{n_1, n_2} \underbrace{\langle T \Phi_{n_1, n_2}^n, \Phi_{n_3, n_4}^n \rangle}_{T_{n_1, n_2}^{n, n_3, n_4}} \underbrace{\langle f, \Phi_{n_1, n_2}^n \rangle}_{s_{n_1, n_2}^n} \right)}_{\tilde{s}_{n_3, n_4}^n} \Phi_{n_3, n_4}^n, \end{aligned}$$

est représentée par la structure à 4 dimensions :

$$T_{n_1, n_2}^{n, n_3, n_4} = \langle T \Phi_{n_1, n_2}^n, \Phi_{n_3, n_4}^n \rangle.$$

Pour chaque n_3, n_4 , la matrice de dimension $(2^n \times 2^n)$

$$T_{n_1, n_2}^{n, n_3, n_4} = \langle T \Phi_{n_1, n_2}^n, \Phi_{n_3, n_4}^n \rangle, \quad 0 \leq n_1, n_2 \leq 2^n - 1$$

est le bloc dans la représentation de la structure T^n de $T_n : V_n \rightarrow V_n$,

$$T^n = \begin{bmatrix} T^{n, 0, 0} & T^{n, 0, 1} & \dots & T^{n, 0, 2^n - 1} \\ T^{n, 1, 0} & T^{n, 1, 1} & \dots & T^{n, 1, 2^n - 1} \\ \vdots & \vdots & \dots & \vdots \\ T^{n, 2^n - 1, 0} & T^{n, 2^n - 1, 1} & \dots & T^{n, 2^n - 1, 2^n - 1} \end{bmatrix}.$$

4.3 L'opérateur $T = \partial_x$

L'opérateur $T = \partial_x$ a la représentation suivante :

$$T_{n_1, n_2}^{n, n_3, n_4} = \langle \partial_x (\Phi_{n_1, n_2}^n), \Phi_{n_3, n_4}^n \rangle = 2^n r_{n_3 - n_1} \delta_{n_2, n_4},$$

où,

$$r_l = \int_{-\infty}^{+\infty} \Phi(x-l) \bar{\Phi}(x) dx, \quad l \in \mathbb{Z}.$$

avec $\bar{\Phi}$ représente le conjugué de Φ .

Ainsi, T^n est complètement déterminé par r_l .

Avec un changement mineur dans la notation, la matrice bloc $T_x^n (= T^n$ pour $T = \partial_x$) devient :

$$T_x^n = \begin{bmatrix} X^{1,1} & X^{1,2} & \dots & X^{1,2^n-1} & X^{1,2^n} \\ X^{2,1} & X^{2,2} & \dots & X^{2,2^n-1} & X^{2,2^n} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ X^{2^n-1,1} & X^{2^n-1,2} & \dots & X^{2^n-1,2^n-1} & X^{2^n-1,2^n} \\ X^{2^n,1} & X^{2^n,2} & \dots & X^{2^n,2^n-1} & X^{2^n,2^n} \end{bmatrix},$$

avec,

$$X_{n_1, n_2}^{n_3, n_4} = T_{n_1, n_2}^{n_3, n_4} = 2^n r_{n_3-n_1} \delta_{n_2, n_4}.$$

Chaque X^{n_3, n_4} contient seulement les colonnes non nulles, la $n_4^{ième}$ colonne. La colonne non nulle de $X^{1,1}$ est donnée par :

$$c = 2^n [0, r_{-1}, r_{-2}, \dots, r_{-(L-2)}, 0, \dots, 0, r_{L-2}, \dots, r_1]^T.$$

Les matrices X^{n_3, n_4} sont obtenues par le biais des "*Forward shift Rows Wrap aRound*" et "*Forward shift Columns Wrap aRound*" qui servent à permuter les lignes et les colonnes comme suit :

$$X^{n_3, n_4+1} = FSRWR(X^{n_3, n_4}), \quad X^{n_3+1, n_4} = FSCWR(X^{n_3, n_4}),$$

où , pour $A = (\vec{c}_1, \vec{c}_2, \dots, \vec{c}_{n-1}, \vec{c}_n)$,

$$FSRWR(A) = [\vec{c}_n, \vec{c}_1, \vec{c}_2, \dots, \vec{c}_{n-1}],$$

et pour,

$$A = \begin{bmatrix} r'_1 \\ \vec{r}_2 \\ \vdots \\ \vec{r}_n \end{bmatrix}, \quad FSCWR(A) = \begin{bmatrix} r'_n \\ \vec{r}_1 \\ \vdots \\ \vec{r}_{n-1} \end{bmatrix}.$$

On note que seulement la matrice $X^{1,1}$, est calculée et stockée comme les autres blocs $X^{i,j}$ peut être dynamiquement produits.

Application numérique

Ce chapitre est consacré à l'application numérique pour illustrer la méthode étudiée dans ce mémoire et pour montrer l'efficacité et la stabilité de cette méthode. On a utilisé le logiciel MATLAB pour construire nos programmes.

On veut résoudre le problème de valeurs aux limites suivant :

$$\begin{cases} -\frac{1}{2}\Delta U(x, y) + (x, y) \nabla U(x, y) + xyU(x, y) = xy \exp(x^2 + y^2) \text{ sur }]0, 1[\times]0, 1[\\ U(0, y) = \exp(y^2), \quad U(1, y) = \exp(1 + y^2), \\ U(x, 0) = \exp(x^2), \quad U(x, 1) = \exp(x^2 + 1), \end{cases} \quad (5.0.1)$$

qui admet pour solution analytique :

$$U(x, y) = \exp(x^2 + y^2). \quad (5.0.2)$$

Pour traiter ce problème par la méthode de collocation orthogonale sur les éléments finis, nous discrétisons le domaine d'étude en choisissant n_{x_e} noeuds, n_{x_c} points de collocation intérieurs suivant l'axe des x et n_{y_e} noeuds, n_{y_c} points de collocation intérieurs suivant l'axe

des y ; comme défini sur le graphe suivant :

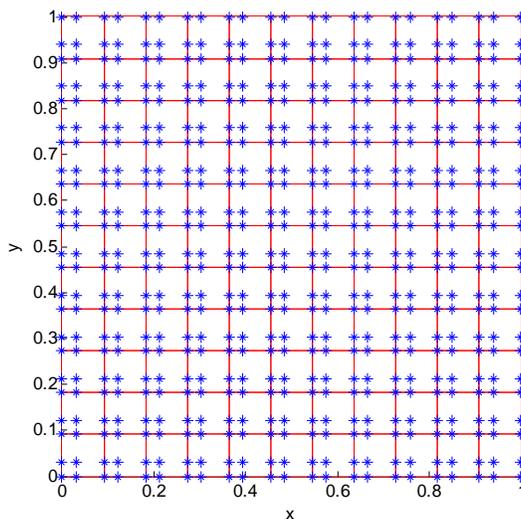


Figure 5 : Discretisation du domaine par MEF et MCO

On note par p le nombre de nœuds total (les nœuds des éléments finis + les points de collocation) :

$$p = (n_{x_e} - 1)(n_{y_e} - 1)(n_{x_c} + 2)(n_{y_c} + 2) - (n_{y_e} - 1)(n_{x_e} - 2)(n_{y_c} + 2) - (n_{x_e} - 1)(n_{y_e} - 2)(n_{x_c} + 2) + (n_{y_e} - 2)(n_{x_e} - 2).$$

Le graphe suivant montre la forme de la solution exacte

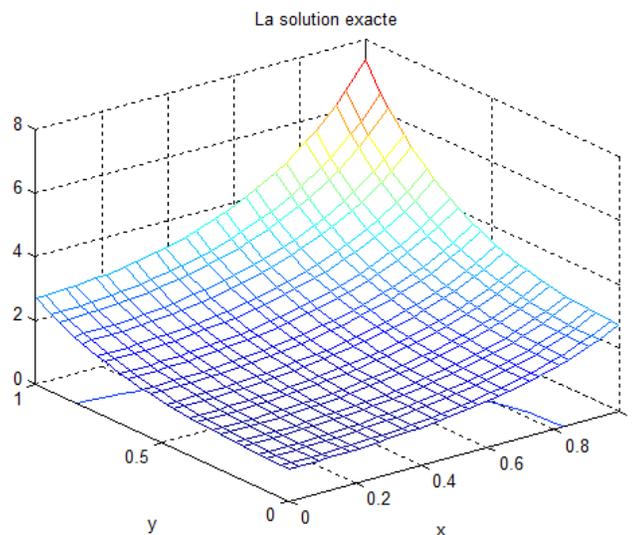


Figure 6 : Solution exacte

En utilisant la méthode de collocation orthogonale sur les éléments finis, on obtient le graphe de la solution approchée suivant :

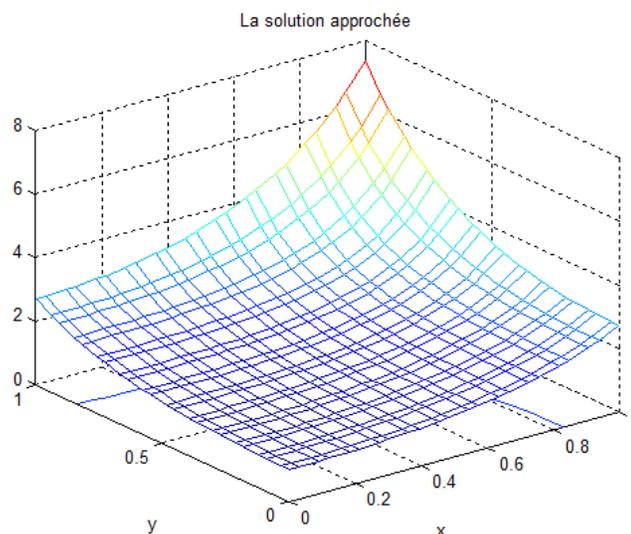


Figure 7 : Solution approché

À première vue, on constate qu'il n'y a pas de différence majeure entre la solution analytique et celle obtenue avec notre méthode. C'est la raison qui nous pousse à étudier l'erreur de cette méthode.

Erreur globale d'approximation

Après avoir obtenu la solution approchée, on peut calculer l'erreur produite par la méthode avec la formule suivante :

$$err = \sqrt{\sum_{i=1}^{nc+2} \sum_{j=1}^{nc+2} (erreur(i, j))^2} \quad (5.0.3)$$

et l'erreur ponctuel donnée par :

$$erreur(i, j) = U_{exacte}(i, j) - U_{approchée}(i, j), \quad (5.0.4)$$

avec $i, j = 1, \dots, p$.

On veut étudier l'erreur d'approximation en partant de $p = 49$ jusqu'à 550 noeuds, on a alors le tableau suivant :

noeuds total(p)	49	81	121	169	225	289	361	441	529
<i>Erreur</i>	0.0204	0.0175	0.0137	0.0102	0.0073	0.0050	0.0033	0.0021	0.0015
<i>t</i> (second)	0.0235	0.0350	0.0591	0.1288	0.3194	0.7250	1.5540	3.2671	4.8649

ce qui se traduit graphiquement par :

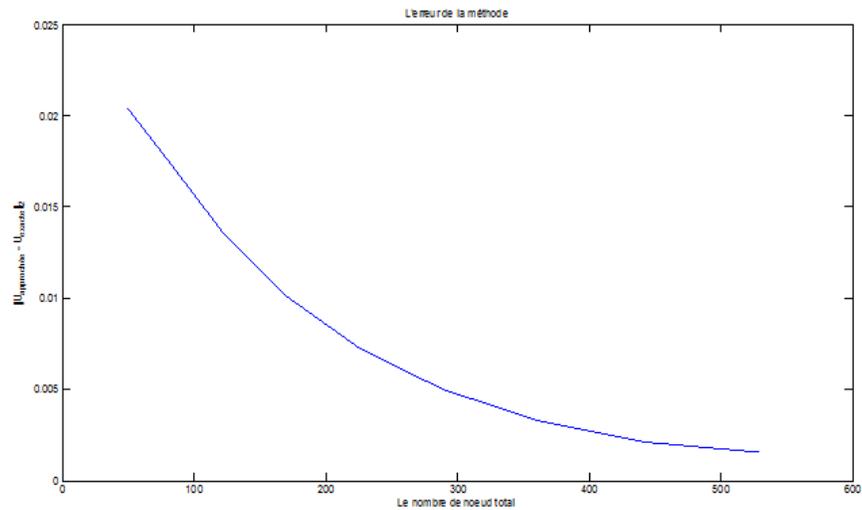


Figure 8 : Erreur de la methode

On constate qu'en augmentant p l'erreur diminue ce qui prouve la convergence et la stabilité numérique pour cet exemple.

Dans ce qui suit on va regarder la rapidité de calcul de la méthode en fonction de p

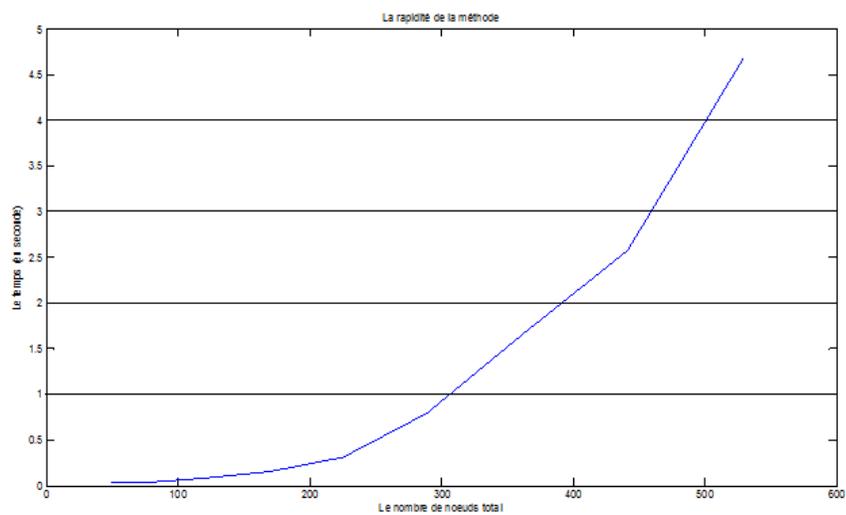


Figure 9 : la rapidite de l'erreur d'approximation.

On constate que plus que p augmente le temps de calcul augmente. En effet, on regarde sur le graphe qu'à $p = 300$ nœuds, $t = 1$ seconde et pour $p = 500$ nœuds (moins du double de 300 nœuds) le temps c'est multiplier par 4.

Conclusion

La méthode de collocation orthogonale sur les éléments finis, a prouvé son efficacité en fournissant une bonne approximation de la solution du problème aux limites. On le voit clairement dans l'application numérique, la convergence, la stabilité numérique de la solution et la précision des résultats prouvent la haute qualité de cette méthode pour un certain nombre de nœuds total.

Une augmentation excessive des points de collocation intérieurs provoque une certaine instabilité numérique, pour y remédier les perspectives sont nombreuses. Parmi ces dernières l'intégration d'une base d'ondelettes à la méthode de collocation. Au quatrième chapitre, nous avons introduit une méthode de discrétisation à base d'ondelettes et nous prévoyons sa concrétisation pour la méthode de collocation en perspectives.

Annexes

Définition 5.0.1 *L'interpolation est un calcul mathématique qui permet de prédire une valeur inconnue à partir de valeurs connues. Ce calcul est utilisé pour estimer les valeurs inconnues de tout point de données géographiques, etc.*

5.1 Polynôme de *Lagrange*

Définition 5.1.1 *Les polynômes de Lagrange en une seule dimension ont la forme suivante :*

$$l_i(x) = \prod_{\substack{k=1 \\ k \neq i}}^{nx+2} \frac{x - x_k}{x_i - x_k},$$

où $nx + 2$ est le nombre de points de collocation.

De plus ils vérifient la propriété suivante :

$$l_i(x_j) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{sinon} \end{cases}$$

Posons

$$\prod_{s=1}^{nx+1} (x - x_s) = P_{nx+2}(x),$$

ainsi, on peut réécrire les polynôme de *Lagrange* sous la forme suivante :

$$l_i(x) = \frac{P_{nx+2}(x)}{(x - x_i) P'_{nx+2}(x_i)}$$

tel que $P'_{nx+2}(x_i)$ désigne la dérivée première de $P_{nx+2}(x)$ au points de collocation x_i . La dérivée première, la deuxième et la troisième de $P_{nx+2}(x)$ sont données par :

$$\begin{aligned}
P'_{nx+2}(x) &= \sum_{m_1=1}^{nx+2} \frac{1}{(x-x_{m_1})} \left(\prod_{\substack{s=1 \\ m_1 \neq s}}^{nx+2} (x-x_s) \right), \\
P''_{nx+2}(x) &= 2! \sum_{m_2=1}^{nx+2} \frac{1}{(x-x_{m_2})} \sum_{\substack{m_1=1 \\ m_2 \neq m_1}}^{nx+2} \frac{1}{(x-x_{m_1})} \left(\prod_{\substack{s=1 \\ m_1 \neq s}}^{nx+2} (x-x_s) \right), \\
P'''_{nx+2}(x) &= 3! \sum_{m_3=1}^{nx+2} \frac{1}{(x-x_{m_3})} \sum_{\substack{m_2=1 \\ m_3 \neq m_2}}^{nx+2} \frac{1}{(x-x_{m_2})} \sum_{\substack{m_1=1 \\ m_2 \neq m_1}}^{nx+2} \frac{1}{(x-x_{m_1})} \left(\prod_{\substack{s=1 \\ m_1 \neq s}}^{nx+2} (x-x_s) \right).
\end{aligned}$$

Alors, les dérivées première et deuxième de $l_i(x)$ sont :

$$\frac{dl_i}{dx}(x) = l'_i(x) = \frac{P'_{nx+2}(x)}{(x-x_i)P'_{nx+2}(x_i)} - \left(\frac{P_{nx+1}(x)}{(x-x_i)^2 P'_{nx+2}(x_i)} \right),$$

et

$$\frac{d^2 l_i}{dx^2}(x) = l''_i(x) = \frac{P''_{nx+2}(x)}{(x-x_i)P'_{nx+2}(x_i)} - \frac{2l'_i(x)}{(x-x_i)},$$

car

$$\frac{P_{nx+2}(x_k)}{(x_k-x_i)^2 P'_{nx+2}(x_i)} = 0.$$

Notons par

$$\begin{aligned}
A_{x_j,i} &= \frac{dl_i}{dx}(x_j), & A_{y_j,i} &= \frac{dl_i}{dy}(y_j) \\
B_{x_j,i} &= \frac{d^2 l_i}{dx^2}(x_j), & B_{y_j,i} &= \frac{d^2 l_i}{dy^2}(y_j)
\end{aligned}$$

ainsi la dérivée première et la deuxième de $l_i(x)$ (resp. $l_i(y)$) aux points de collocation x_j (resp. y_j) est défini comme suit :

$$\left\{ \begin{array}{l} A_{x_j,j} = \lim_{x_i \rightarrow x_j} \frac{dl_i}{dx}(x_j) = \frac{P''_{nx+2}(x_j)}{2P'_{nx+2}(x_j)} \\ A_{x_j,i} = \frac{P'_{nx+2}(x_i)}{(x-x_i)P'_{nx+2}(x_j)}, \text{ pour } i \neq j, \end{array} \right.$$

et

$$\left\{ \begin{array}{l} B_{x_j,j} = \lim_{x_i \rightarrow x_j} \frac{d^2 l_i}{dx^2}(x_j) = \frac{P'''_{nx+2}(x_j)}{3P'_{nx+2}(x_j)} \\ B_{x_j,i} = \frac{P''_{nx+2}(x_j)}{(x_j-x_i)P'_{nx+2}(x_i)} - \frac{2l'_i(x_j)}{(x_j-x_i)}, \text{ pour } i \neq j, \end{array} \right.$$

$$B_{x_j,i} = \frac{P''_{nx+2}(x_j)}{(x_j-x_i)P'_{nx+2}(x_i)} - \frac{2}{(x_j-x_i)} \left(\frac{P'_{nx+2}(x_i)}{(x-x_i)P'_{nx+2}(x_j)} \right)$$

avec $i, j = 1, \dots, nx + 2$.

Le procédé est le même pour la dérivée partielle par rapport à y .

Définissant maintenant le polynôme de *Lagrange* L_{ij} en deux dimensions par :

$$L_{ij}(x, y) = l_i(x).l_j(y), \quad i = 1, \dots, nx + 2 ; \quad j = 1, \dots, nx + 2.$$

5.2 Fonction Gamma

Définition 5.2.1 La fonction Gamma notée par $\Gamma(z)$ est définie par l'intégrale :

$$\Gamma(z) = \int_0^{+\infty} \exp(-t)t^{z-1} dt$$

pour tout complexe z dont la partie réelle est strictement positive.

5.3 Produit de convolution

Définition 5.3.1 Soient f, g deux fonctions de carré sommables, on définit le produit de convolution de ces deux fonctions par

$$f(x) * g(x) = \int_{\mathbb{R}} f(t)g(x - t)dt.$$

5.4 Base de Riesz

Définition 5.4.1 Une famille de vecteurs $\{x_n, n \in \mathbb{N}\}$ est une base de Riesz d'un espace de Hilbert H si : elle est libre, s'il existe deux constantes $\alpha > 0, \beta > 0$ tel que :

$$\alpha \left(\sum_n |a_n|^2 \right)^{\frac{1}{2}} \leq \left\| \sum_n a_n x_n \right\|_H \leq \beta \left(\sum_n |a_n|^2 \right)^{\frac{1}{2}},$$

le quotient $\frac{\alpha}{\beta}$ est le conditionnement de la base de Riesz.

Dans le cas où $\alpha = \beta = 1$ on obtient une base orthonormée.

Alors pour toute suite de scalaires $(a_n)_{n \in \mathbb{N}}$:

$$\forall f \in H : f = \sum_{n \in \mathbb{N}} a_n x_n.$$

5.5 Transformée de Fourier

Définition 5.5.1 On appelle transformée de Fourier d'une fonction $f \in L^1(\mathbb{R})$

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t) \exp(-i\omega t) dt,$$

La fonction \hat{f} peut être vue comme le spectre en fréquences de la fonction f .

L'espace $C^p(\mathbb{R})$ est l'espace des fonctions continues et p fois continûment dérivables.

5.6 Méthode LU

La méthode **LU** consiste à factoriser une matrice A qui est inversible du système $Ax = b$ en un produit d'une matrice triangulaire unitaire inférieure L et une matrice triangulaire supérieure U . Dans ce cas notre système peut être décomposé en deux systèmes triangulaires faciles à résoudre (puisque les matrices sont triangulaires) avec une variable intermédiaire y i.e

$$\begin{aligned} Ax = b &\Leftrightarrow LUx = b \\ &\Leftrightarrow \begin{cases} Ly = b \\ Ux = y \end{cases} \end{aligned}$$

posons :

$$\begin{cases} A = (a_{ij})_{1 \leq i, j \leq n} \\ L = (l_{ij})_{1 \leq i, j \leq n} \\ U = (u_{ij})_{1 \leq i, j \leq n} \end{cases}$$

où n est la dimension de la matrice A .

L'algorithme suivante montre le calcul de deux matrices triangulaires L et U

Etape 1 : Choisir l_{11} et u_{11} de telles façons que

$$l_{11}u_{11} = a_{11}$$

Etape 2 : Déterminer les coefficients de la première ligne de U par :

$$u_{1j} = \frac{a_{1j}}{l_{11}} \text{ pour } j = 2, 3, \dots, n$$

Etape 3 : Déterminer les coefficients de la première colonne de L par :

$$l_{j1} = \frac{a_{j1}}{u_{11}} \text{ pour } j = 2, 3, \dots, n$$

Etape 4 : pose $i = 2$

Etape 5 : Choisir l_{ii} et u_{ii} de telles façons que

$$l_{ii} u_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{ik} u_{ki}.$$

Etape 6 : Si $i < n$, aller à l'étape 7.

Si $i = n$, aller à l'étape 10.

Etape 7 : Déterminer les coefficients de la $i^{\text{ème}}$ colonne de L par :

$$l_{ji} = \frac{\left(a_{ji} - \sum_{k=1}^{i-1} l_{jk} u_{ki} \right)}{u_{ii}} \text{ pour } j = i + 1, i + 2, \dots, n.$$

Etape 8 : Déterminer les coefficients de la $i^{\text{ème}}$ ligne de U par :

$$u_{ij} = \frac{\left(a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \right)}{l_{ii}} \text{ pour } j = i + 1, i + 2, \dots, n.$$

Étape 9 : Mettre $i = i + 1$ et aller à l'étape 5.

Étape 10 : Fin.

Une fois calculées, les matrices \mathbf{L} et \mathbf{U} , résoudre le système linéaire consiste simplement à résoudre successivement les deux systèmes triangulaires :

$$\begin{cases} \mathbf{L}y = b \\ \mathbf{U}x = y \end{cases}$$

5.7 Programmation

La programmation sous Matlab a été faite de la manière suivante :

Plusieurs fichiers ont été créés comme la discrétisation par la méthode des éléments finis (**disc_MEF**), la discrétisation par la MCO (**disc_MCO**) et d'autres sous programmes utilisés pour discrétiser l'équation différentielle. Après la prise en compte des conditions aux limites on utilise la factorisation **LU** pour trouver la solution recherchée. Ces sous programmes ou fichiers seront appelés par la suite dans le programme principale (**pp**). Nous allons voir le contenu de chaque fichier :

Le programme principale(pp)

```
clear all
close all
% ===== LES DONNÉES =====
% <>-<>-<>-<>-<>-<>-<> Les donnees Physique >-<>-<>-<>-<>-<>-<>
a = 0;
b = 1;
c = 0;
d = 1;
% <>-<>-<>-<>-<>-<>-<> Les données Mathématique >-<>-<>-<>-<>-<>-<>
nxe = 12;
nye = 12;
nxc = 1;
nyc = 1;
px = (nxe-1)*(nxc+1)+1;
py = (nye-1)*(nyc+1)+1;
p = (nxe-1)*(nye-1)*(nxc+2)*(nyc+2)-(nye-1)*(nxe-2)*(nyc+2)-(nxe-1)*(nye-2)...
*(nxc+2)+(nye-2)*(nxe-2);
AG = zeros(p,p);
BG = zeros(p);
% =====
% discrétisation par éléments finis
[xL,xe,ye,hxe,hye] = discr_MEF(a,b,c,d,nxe,nye)
[noe,elt,element,elemfini,xg]=maillage(a,b,c,d,nxe,nye,nxc,nyc);
%-----
% discrétisation par MCO
nel = (nxc+2)*(nyc+2);
mx = 0; my = 1; pop = nxe;
for m = 1 : elt
    if mod(m,pop)==0 & m >= pop
        mx = 0;
        my = my +1;
        pop = pop + nxe-1;
    end
    mx=mx+1;
    x1 = xe((mx-1)+1)
```

```

x2 = xe((mx-1)+2)
y1 = ye((my-1)+1)
y2 = ye((my-1)+2)
[xcc,ycc] = discr_MCO(x1,x2,y1,y2,nxc,nyc)
for i=1 :length(xcc)
    for j=1 :length(ycc)
        xc(i)=(xcc(i)-xcc(1))/hxe;
        yc(j)=(ycc(j)-ycc(1))/hye;
    end
end
end
%-----
% description des equations par la MCO
AL = Matrice_Locale(xc,yc,nxc,nyc,hxe,hye);
BL = Vect_Locale(xc,yc,nxc,nyc,hxe,hye);
end
% Numérotation globale des points de collocation et de noeuds
[noe,elt,element,elemfini,xg] = maillage(a,b,c,d,nxe,nye,nxc,nyc);
%-----
% l'assemblage des elements finis
for i = 1 :nel
    for j = 1 :nel
        AG(element(i,m),element(j,m)) = AG(element(i,m),element(j,m)) + AL(i,j);
        BG(element(i,m)) = BG(element(i,m)) + BL(i);
    end
end
end
%-----
% renumérotation des points
[x_g,xx,yy] = x_global(a,b,c,d,nxe,nye,nxc,nyc);
% imposer les conditions aux limites
[AG,BG] = impose( p,xx,yy,nxe,nye,nxc,nyc,AG,BG);
%-----
% resolutions du systeme lineaire
U = matLU(p,AG,BG);
% calcul de la solution exacte
for l =1 :py
    for k =1 :px
        %sol_exact(l,k) = sin(pi*xx(k))*sin(2*pi*yy(l));
        sol_exact(l,k) = exp(xx(k)^2+yy(l)^2);
    end
end
end
UM = transVectMat(px,py,U);
err = erreur(px,py,sol_exact,UM);
[X,Y] = meshgrid(xx,yy);
figure(1)
meshc(X,Y,UM);

```

```
figure(2)
meshc(X,Y,sol_exact);
figure(3)
meshc(X,Y,err);
```

Ce programme fait rappeler les sous programmes suivants :

```
maillage
function [noe,elt,element,elemfini,xg] = maillage(a,b,c,d,nxe,nye,,nxc,nyc)
% décrétisations du domaine [a,b]×[c,d]
% nxe : le nombre des noeuds suivant l'axe des x
% nye : le nombre des noeuds suivant l'axe des y
% elt : le nombre des elements finis
%-----
[xL,xe,ye,hxe,hye] = discr_MEF(a,b,c,d,nxe,nye);
om = 0;
for j = 1 :nye
    for i = 1 :nxe
        om = om + 1;
        xg(1,om)= xe(i);
        xg(2,om)= ye(j);
    end
end
%-----
noe = (nxe*(nxc+1)-nxc)^2;
elt = 0;
for j = 1 :nye-1
    for i = 1 :nxe-1
        elt = elt +1;
        for k = 1 :nyc+2
            for l = 1 :nxc+2
                element((k-1)*(nxc+2)+1,(j-1)*(nxe-1)+i) = (k-1)*...
                    (nxe*(nxc+1)-nxc)+(i-1)*(nxc+1)+1+(j-1)*(nyc+1)...
                    *(nxe*(nxc+1)-nxc);
            end
        end
    end
end
el = 0;
for j = 1 :nye-1
    for i = 1 :nxe-1
        el = el +1;
        elemfini(1,el) = (j-1)*nxe + i;
        elemfini(2,el) = (j-1)*nxe + i+1;
        elemfini(3,el) = (j-1)*nxe + i+1+nxe;
        elemfini(4,el) = (j-1)*nxe + i +nxe;
```

```

    end
end
%-----
for m = 1 :elt
    x1 = xg(1,element(1,m));
    x2 = xg(1,element(2,m));
    x3 = xg(1,element(3,m));
    x4 = xg(1,element(4,m));
    y1 = xg(2,element(1,m));
    y2 = xg(2,element(2,m));
    y3 = xg(2,element(3,m));
    y4 = xg(2,element(4,m));
    x = [x1,x2,x3,x4,x1];
    y = [y1,y2,y3,y4,y1];
    plot(x,y,'r')
    hold on
end
%-----
discr_MEF
function [xL,xe,ye,hxe,hye] = discr_MEF(a,b,c,d,nxe,nye)
% Décritisation du domaine [a,b]×[c,d]...
% a : borne inférieure suivant l'axe des x
% b : borne supérieure suivant l'axe des y
% c : borne inférieure suivant l'axe des y
% d : borne supérieure suivant l'axe des x
% nxe : nombre des noeuds suivant l'axe des x
% nye : nombre des noeuds suivant l'axe des y
%-----
hxe = (b - a) / ( nxe - 1);
hye = (d - c) / ( nye - 1);
%discrétisation suivant l'axe des x
xe(1) = a;
for i = 2 :nxe
    xe(i) = xe(i-1) + hxe;
end
%discrétisation suivant l'axe des y
ye(1) = c;
for j = 2 :nye
    ye(j) = ye(j-1) + hye;
end
%-----
for j = 1 :nye
    for i = 1 :nxe
        xL(1, nxe*(j - 1)+ i) = xe(i);
        xL(2, nxe*(j - 1)+ i) = ye(j);
    end
end

```

```

    end
end

```

```

discr_MCO

```

```

function [xc,yc] = discr_MCO(x1,x2,y1,y2,nxc,nyc)
% nxc : le nombre de points de collocation intérieur suivant l'axe des x
% nyc : le nombre de points de collocation intérieur suivant l'axe des y
% discrétisations de l'élément [x1,x2]×[y1,y2]
% discrétisation du l'axe des x et y
%-----
hxc = (b - a) / ( nxc + 1);
hyc = (d - c) / ( nyc + 1);
% discrétisation suivant l'axe des x
x = Racines_Jacobi(nxc);
y = Racines_Jacobi(nyc);
xc(1) = x1;
for i = 2 :nxc+1
    xc(i) = xc(i-1) + hxc;
end
xc(nxc+2) = x2;
x = Rac_Jacobi(nxc);
% discrétisation suivant l'axe des y
yc(1) = y1;
for j = 2 :nyc+1
    yc(j) = yc(j-1) + hyc;
end
yc(nxc+2) = y2;
%yc = Rac_Jacobi(nyc);

```

```

function [x_g,xx,yy] = x_global(a,b,c,d,nxe,nye,nxc,nyc)
[xL,xe,ye,hxe,hye] = discr_MEF(a,b,c,d,nxe,nye);
[xc,yc] = discr_MCO(xe(1),xe(2),ye(1),ye(2),nxc,nyc);
px = (nxe-1)*(nxc+1)+1;
py = (nye-1)*(nyc+1)+1;
% =====
% racines de JACOBI
for i = 1 :nxc+1
    hx(i) = xc(i+1)-xc(i);
end
for i = 1 :nyc+1
    hy(i) = yc(i+1)-yc(i);
end
xx(1) = a;
k = 0;
pop = nxc+3;

```

```

for i = 2 :px
    if mod(i,pop) ==0 & i>=pop
        k = 0;
        pop = pop + nxc+1;
    end
    k = k+1;
    xx(i) = xx(i-1)+ hx(k);
end
yy(1) = c;
kk = 0;
pip = nyc+3;
for i = 2 :py
    if mod(i,pip) ==0 & i>=pip
        kk = 0;
        pip = pip+nyc+1;
    end
    kk=kk+1;
    yy(i) = yy(i-1)+hy(kk);
end
%=====
hxc = (xe(2)-xe(1))/(nxc+1);
hyc = (ye(2)-ye(1))/(nyc+1);
%xx(1) = a;
%for i = 2 :px
    % xx(i) = a +(i-1)* hxc;
%end
%yy(1) = c;
%for i = 2 :py
    % yy(i) = c + (i-1)*hyc;
%end
%=====
for j = 1 : py
    for i = 1 :px
        m = (j-1)*px + i;
        x_g(1,m) = xx(i);
        x_g(2,m) = yy(j);
        %plot (xx(i),yy(j),'*')
    end
end
%-----
calcul de polynôme de Lagrange aux points de collocation xi
function lag = Lag( i,x,xi)
nc =length(x)-2;
p =1;
for j =1 :nc+2

```

```

    if i~=j
        p = p*(x-xi(j))/(xi(i)-xi(j));
    end
end
lag=p;

```

```

% la 1ere diréevée de du Polynome de Lagrange % à x
function Ax = mat_A(x)
nc =length(x);
for i =1 :nc
    pn(i,1)=x(i)-x(1);
    pn1(i,1)=1;
    pn2(i,1)=0;
    pn3(i,1)=0;
end
for j =2 :nc
    for i=1 :nc
        pn(i,j)=(x(i)-x(j))*pn(i,j-1);
        pn1(i,j)=(x(i)-x(j))* pn1(i,j-1)+pn(i,j-1);
        pn2(i,j)=(x(i)-x(j))* pn2(i,j-1)+2*pn1(i,j-1);
        pn3(i,j)=(x(i)-x(j))* pn3(i,j-1)+3*pn2(i,j-1);
    end
end
for i=1 :nc
    for j=1 :nc
        if i==j
            Ax(i,j)=pn2(i,nc)/(2*pn1(i,nc));
        else
            Ax(i,j)=pn1(i,nc)/((x(i)-x(j))*pn1(j,nc));
        end
    end
end
end

```

```

% la 2ème diréevée de du Polynome de Lagrange % à x
function Bx = mat_B(x)
nc =length(x);
for i =1 :nc
    pn(i,1)=x(i)-x(1);
    pn1(i,1)=1;
    pn2(i,1)=0;
    pn3(i,1)=0;
end
for j =2 :nc
    for i =1 :nc
        pn(i,j)=(x(i)-x(j))*pn(i,j-1);
    end
end

```

```

    pn1(i,j)=(x(i)-x(j))* pn1(i,j-1)+pn(i,j-1);
    pn2(i,j)=(x(i)-x(j))* pn2(i,j-1)+2*pn1(i,j-1);
    pn3(i,j)=(x(i)-x(j))* pn3(i,j-1)+3*pn2(i,j-1);
  end
end
for i = 1 :nc
  for j = 1 :nc
    if i == j
      Bx(i,j) = pn3(i,nc)/(3*pn1(i,nc));
    else
      Bx(i,j) = (pn2(i,nc) / ((x(i) - x(j)) * pn1(j,nc))) - ((2 / (x(i) - x(j))...
        * pn1(i,nc) / pn1(j,nc)) / (x(i) - x(j)));
    end
  end
end
end

```

%calcul du polynome de Lagrange aux points de collocation

function Mx=mat_Mx(x)

nc=length(x)-2;

for i =1 :nc+2

 for j =1 :nc+2

 if i==j

 Mx(i,j)=1;

 else

 Mx(i,j) = lag(i,x(j),x);

 end

 end

end

%calcul du polynome de Lagrange aux points de collocation

function My=mat_My(y)

nc=length(y)-2;

for i =1 :nc+2

 for j =1 :nc+2

 if i==j

 My(i,j)=1;

 else

 My(i,j) = lag(i,y(j),y);

 end

 end

end

polynome de Jacobi

function Jn = Jn(n,Gamma,x)

% n : désigne le degré du polynome de Jacobi

```

%-----
pr = (-1) ^ n;
for j = 1 : n
    pr = pr + (-1) ^ (n - j) * Gamma(j) * x ^ j;
end
Jn = pr;

```

```

%calcul les racine du polynome de Jacobi...
function xc = Rac_Jacobi(n)
itermax =100;
alpha = 1;
beta =0;
tol =1.0e-10;
x(1) =0;
xc(1) =0;
k=1;
err1 =1;
%calcul de Gamma
Gamma(1) = n*(n + 1 + alpha + beta)/(1 + beta);
for i = 2 : n
    Gamma(i) = (n - i + 1)*(n + i + alpha + beta)/(i*(i + beta)) * Gamma(i-1);
end
while err1>=tol
    k =k+1;
    x(k) = x(k-1)-Jn(n,Gamma,x(k-1))/dJn(n,Gamma,x(k-1));
    err1 = abs(x(k)-x(k-1));
end
xc(2) = x(k);
l = 1;
for j =2 :n
    k =1;
    l = l+1;
    y(1) =xc(l)+1.0e-4;
    err2 =1;
    while err2>=tol
        k =k+1;
        rest = calcul(n,l,y(k-1),xc,alpha,beta);
        y(k) = y(k-1) - rest;
        err2 = abs(y(k)-y(k-1));
    end
    xc(l+1) = y(k);
end
xc(n+2) = 1;

```

```

function rest = calcul(n,k,x,xc,alpha,beta)

```

```

g(1) = (beta+1)/(alpha+beta+2);
for i = 2 : n
    g(i) = 0.5*(1- (alpha^2-beta^2)/((2* i+alpha+beta-1)^2-1));
end
h(1) = 0;
h(2) = (alpha+1)*(beta+1)/((alpha+beta+2)^2*(alpha+beta+3));
for i = 3 : n
    h(i) = (i-1)*(i+alpha-1)*(i+beta-1)*(i+alpha+beta-1)/((2*i+alpha+beta-1)...
        *(2*i+alpha+beta-2)^2*(2*i+alpha+beta-3));
end
P = 1;
dP0 = 0;
P(1) = (x-g(1))*P0;
dP(1) = P0+(x-g(1))*dP0;
P(2) = (x-g(2))*P(1)-h(2)*P0;
dP(2) = P(1)+(x-g(2))*dP(1)-h(2)*dP0;
for j = 3 : n
    P(j)=(x-g(j))*P(j-1)-h(j)*P(j-2);
    dP(j)=P(j-1)+(x-g(j))*dP(j-1)-h(j)*dP(j-2);
end
pr = 0;
for i =1 :k
    pr= pr+1/(x-xc(i));
end
rest = (P(n)/dP(n)) / (1-(P(n) / dP(n))* pr);

```

```

% dirévée du polynome de Jacobi...
function dJn = dJn(n,Gamma,x)
pr = (-1)^(n-1) * Gamma(1);
for j = 2 : n
    pr = pr+(-1)^(n-j)*j* Gamma(j)* x^(j-1);
end
dJn = pr;

```

```

%calcul de la fonction Gamma ...
%n désigne le degré du Jn....
function Gamma = Gamma (i,alpha ,beta)
alpha = 1;
beta = 0;
Gamma(1) = n * (n + 1 + alpha + beta)/(1 + beta);
for i = 2 : n
    Gamma(i) = (n - i + 1)*(n + i + alpha + beta)/(i*(i + beta)) * Gamma(i-1);
end

```

```

function x = matLU(n,A,b)

```

```

[n,n]= size(A) ;
l = zeros(n,n) ;
u = zeros(n,n) ;
for i = 1 :n
    l(i,i) = 1 ;
end
%calcul de la matrice u
for r = 1 :n
    for j = r :n
        s1 =0;
        for k =1 :r-1
            s1 = s1 + l(r,k)*u(k,j) ;
        end
        u(r,j) = A(r,j) - s1 ;
    end
    for i = r+1 : n
        s2 = 0 ;
        for k = 1 : r - 1
            s2 = s2 +l(i,k) * u(k,r) ;
        end
        l(i,r) = (A(i,r)-s2) / u(r,r) ;
    end
end
end
for i = 1 :n
    c = 0 ;
    for j =1 :i-1
        c =c+l(i,j)*x(j) ;
    end
    x(i)=(b(i)-c)/l(i,i) ;
end
end
for i = n :-1 :1
    p = 0 ;
    for j =i+1 : n
        p =p+u(i,j)*x(j) ;
    end
    x(i) =(x(i)-p)/u(i,i) ;
end
end

```

```

%calcul de matrice locale
function AL = Matrice_Locale(x,y)
% AL : etant la matrice locale
% Mx : calcul du polynome de Lagrange aux points de collocation xc
% My : calcul du polynome de Lagrange aux points de collocation yc
% nc : le nombre des points de collocation intérieur
nc = length(x)-2 ;

```

```

[a,b1,b2,c,f] = coeff(x,y);
Ax = mat_A(x);
Ay = mat_A(y);
Bx = mat_B(x);
By = mat_B(y);
for i = 1 :nc+2
    for j =1 :nc+2
        Mx(i,j) = lag(i,x(j),x);
        My(i,j) = lag(i,y(j),y);
    end
end
for i = 1 :nc+2
    for k = 1 :nc+2
        for j = 1 :nc+2
            for l = 1 :nc+2
                AL((l-1)*(nc+2)+k,(j-1)*(nc+2)+i) = a(k,l)*(Bx(i,k)*My(j,l)+Mx(i,k)*By(j,l))...
                    +b1(k,l)*Ax(i,k)*My(j,l)+b2(k,l)*Mx(i,k)*Ay(j,l)+c(k,l)*Mx(i,k)*My(j,l);
            end
        end
    end
end
end
end

```

```

function Ax = mat_A(x)
% la 1ere diréevée du Polynome de Lagrange % à x
nc =length(x);
for i =1 :nc
    pn(i,1)=x(i)-x(1);
    pn1(i,1)=1;
    pn2(i,1)=0;
    pn3(i,1)=0;
end
for j =2 :nc
    for i=1 :nc
        pn(i,j)=(x(i)-x(j))* pn(i,j-1);
        pn1(i,j)=(x(i)-x(j))* pn1(i,j-1)+pn(i,j-1);
        pn2(i,j)=(x(i)-x(j))* pn2(i,j-1)+2*pn1(i,j-1);
        pn3(i,j)=(x(i)-x(j))* pn3(i,j-1)+3*pn2(i,j-1);
    end
end
for i = 1 :nc
    for j = 1 :nc
        if i == j
            Ax(i,j) = pn2(i,nc)/(2*pn1(i,nc));
        else
            Ax(i,j) = pn1(i,nc)/((x(i)-x(j))*pn1(j,nc));
        end
    end
end

```

```

        end
    end
end

function Bx = mat_B(x)
% la 2ème dirévée de du Polynome de Lagrange % à x
% nc : le nombre des points de collocation intérieur
% Bx : calcul la 2ème dirévée de du Polynome de Lagrange % à x
nc = length(x);
for i = 1 :nc
    pn(i,1) = x(i)-x(1);
    pn1(i,1) = 1;
    pn2(i,1) = 0;
    pn3(i,1) = 0;
end
for j = 2 :nc
    for i = 1 :nc
        pn(i,j) = (x(i)-x(j))*pn(i,j-1);
        pn1(i,j) = (x(i)-x(j))* pn1(i,j-1)+pn(i,j-1);
        pn2(i,j) = (x(i)-x(j))* pn2(i,j-1)+2*pn1(i,j-1);
        pn3(i,j) = (x(i)-x(j))* pn3(i,j-1)+3*pn2(i,j-1);
    end
end
for i = 1 :nc
    for j = 1 :nc
        if i == j
            Bx(i,j) = pn3(i,nc)/(3*pn1(i,nc));
        else
            Bx(i,j) = (pn2(i,nc)/((x(i)-x(j))* pn1(j,nc)))-((2/(x(i)-x(j))*pn1(i,nc)...
            /pn1(j,nc))/(x(i)-x(j)));
        end
    end
end
end

```

la prise en compte les conditions aux limites

```

function [AG,BG] = impose( p,nxe,nye,nc,x,y,AG,BG)
% nc : le nombre des points de collocation intérieur
% nxc : le nombre de points de collocation intérieur suivant l'axe des x
% nyc : le nombre de points de collocation intérieur suivant l'axe des y
% ind : un indicateur pour détecter le noeud ou on doit imposer une valeur
% val_impose : un vecteur pour la valeur à imposé
% AG : la matrice globale
% BG : le second membre globale
%
%
```

```

[u0y,u1y,ux0,ux1] = condition_lim(nxc,nyc,nxe,nye,x,y);
for i = 1 : p
    ind(i) = 0;
    val_impose(i) = 0;
end
% la condition au limite en x = 0 et y quelconque
k =0;
for i = 1 : (nex-1)*(nc+1)+1
    k = k+1;
    ind(i) = 1;
    val_impose(i) = u0y(k);
end
% la condition en x=1 et y quelconque
k =0;
for i = p : (nex-1)*(nc+1) : nxe
    k = k+1;
    ind(i) = 1;
    val_impose(i) = u1y(k);
end
% la condition en x quelconque et y = 0
for i = 1 : (nye-1)*(nc+1)+1 : p-((nye-1)*(nc+1)+1)
    ind(i) = 1;
    val_impose(i) = ux0(i);
end
% la condition en x quelconque et y = 1
k =0;
for i = p-((nye-1)*(nc+1)+1) :p
    k = k+1;
    ind(i) = 1;
    val_impose(i) = ux1(k);
end
for i =1 :p
    if ind(i) == 1
        for j = 1 :p
            if j ~ = i
                BG(j) = BG(j)-AG(j,i)* val_impose(i);
                AG(i,j)=0;
            end
        end
        for k =1 :p
            AG(i,k) =0;
        end
        AG(i,i) = 1;
        BG(i) = val_impose(i);
    end
end

```

end

les conditions aux limites imposées

```
function [u0y,u1y,ux0,ux1] = condition_lim(nxc,nyc,nxe,nye,x,y)
```

```
for i = 1 :(nxe-1)*(nxc+1)+1
```

```
    ux0(i) = exp(x(i)^2);
```

```
    ux1(i) = exp(1+x(i)^2);
```

```
end
```

```
for i = 1 :(nye-1)*(nyc+1)+1
```

```
    u0y(i) = exp(y(i)^2);
```

```
    u1y(i) = exp(1 + y(i)^2);
```

```
end
```

```
function BL = Vect_Locale(x,y,nxc,nyc,hxe,hye)
```

```
% nxe : nombre des pont de collocation suivant l'axe des x
```

```
% nyc : nombre des pont de collocation suivant l'axe des y
```

```
% nxc = length(x)-2;
```

```
% nyc = length(y)-2;
```

```
[a,b1,b2,c,f] = coeff(nxc,nyc,x,y);
```

```
for j = 1 :nyc+2
```

```
    for i = 1 :nxc+2
```

```
        BL((j-1)*(nxc+2)+i) = f(i,j);
```

```
    end
```

```
end
```

```
%norme matricielle...
```

```
function nr2M = nr2Matrix(n,M)
```

```
Pr = 0;
```

```
for i = 1 :n
```

```
    for j = 1 :n
```

```
        Pr = Pr + M(i,j)^2;
```

```
    end
```

```
end
```

```
nr2M = sqrt(Pr);
```

```
function norme2 = norme2(n,y)
```

```
s = 0;
```

```
for i = 1 : n
```

```
    s = s + y(i)^2;
```

```
end
```

```
norme2 = sprt(s);
```

Bibliographie

- [1] **A. Khenanne** : *Méthode des Elements Finis*, Enoncé des principes de base, OPU, 1997.
- [2] **A. M. Pascal, V. Perrier** : *Orthonormal wavelet bases adapted for partial differential equations with boundary conditions*. SIAM J. MATH. ANAL. °c 1998 Society for Industrial and Applied Mathematics 29, (1998) 1040–1065.
- [3] **A.M. Tahar** : *Méthodes Numériques Tome 1, Méthode des différences finies Méthodes intégrales et variationnelles*. OPU 2007
- [4] **A. Quarteroni, R. Sacco, F. Saleri** : *Méthodes Numériques algorithmes, analyse et applications*. Springer-Verlag Itatia, Milano 2007.
- [5] **Ch. Daveau** : *Méthode des éléments finis*. Mémoire de Master, Université de Cergy-Pontoise, 2006.
- [6] **J. C. Pesquet, B. P- Popescu** : *Ondelettes et Applications. Techniques de l'ingénieur, France 2012*.
- [7] **J. Demaeyer, M. Bebronne & S. Forthomme** : *Les Ondelettes. Deuxième Candidature en Sciences Physiques, Université Libre de Bruxelles*.
- [8] **D. Gouri, T. Gilbert** : *Une Présentation de la Méthode des Elements Finis*, LES PRESSES DE L'UNIVERSITE LAVAL QUEBEC MALOINE S.A EDITEUR PARIS, 1981.
- [9] **F. Golay ANAM/ MNC** : *Modélisation et calcul scientifique : La méthode des éléments finis*. Université de TOULON.
- [10] **F. Jędrzejewski** : *Introduction aux méthodes numériques* Deuxième édition, Springer, 2005.
- [11] **G. Thomas** : *Méthode des éléments finis en mécanique des structures*. PRESSES POLYTECHNIQUES ET UNIVERSITAIRES ROMANDES, 2000.
- [12] **M. A. Hajj, S. Melkonian, and R. Vaillancourt** : *Two-Dimensional Wavelet Bases For Partial Differential operators and Applications*, December 2003.
- [13] **M. Stéphane** : *Une exploration des signaux en ondelettes*. Editions de l'école polytechnique, Décembre, 2009.
- [14] **O. Belhamiti** : *A new approach to solve a set of nonlinear split boundary value problems*. *Commun Nonlinear Sci Numer Simulat* 17 (2012) 555–565

-
- [15] **O. Belhamiti** : *Implémentation de la Méthode de collocation orthogonal sur les éléments finis*. Thèse de magister, Université de Mostaganem 2003.
- [16] **O. Hervé** : *Méthode des éléments finis. Centrale Nantes 28/09/2008*.
- [17] **S. Arora, S.S. Dhaliwal, V.K. Kukreja** : Solution of two point boundary value problems using orthogonal collocation on finite elements. *Applied Mathematics and computation* 171 (2005) 358-370.
- [18] **S. M. Karakach** : *La Méthode Formelle De Collocation à Base d'Ondelettes Pour La Résolution D'une Équation aux Dérivées Partielles*. Mémoire de Master, Université de Mostaganem, Juillet 2011.
- [19] **T. Pierre** : *Éléments finis pour l'ingénieur Grands principes et petites recettes*. LAVOISIER, 2006.
- [20] **Y. Meyer** : *Wavelets and Operators*. Cambridge University Press, 2004.