



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE  
LA RECHERCHE SCIENTIFIQUE  
UNIVERSITÉ ABDELHAMID IBN BADIS - MOSTAGANEM

Faculté des Sciences Exactes et de l'Informatique  
Département de Mathématiques et d'Informatique  
Filière : Mathématiques

MEMOIRE DE FIN D'ÉTUDES  
Pour l'Obtention du Diplôme de Master en Mathématiques  
Option : **Modélisation Contrôle et Optimisation**

THEME :

***Résolution des EDPs par la Méthode des Ondelettes de  
Legendre en 2D***

Etudiante : M<sup>elle</sup> BENKAHLA Fatiha

M<sup>elle</sup> CHABOUNI Aci

Président:	BOKHARI Ahmed	M.A.A	U. Media
Encadreur:	ABLAOUI Naima	M.A.A	U. Mostaganem
Examineur:	BELHAMITI Omar	M.C.A	U. Mostaganem

Année Universitaire 2015/2016

---

# Résumé

---

Dans ce travail, on a rappelé les ondelettes de Legendre en deux dimensions ainsi que les matrices opérationnelles d'intégration associées à chaque variable. On a démontré que ces matrices sont inversibles.

Ensuite on a appliqué la décomposition en ondelettes de Legendre pour résoudre une équation aux dérivées partielles linéaire d'ordre deux et à coefficients constants. En faisant remarquer que cette résolution se ramène à la résolution d'un système algébrique, dont les inconnues sont les coefficients de cette décomposition.

Des tests numériques ont été effectués pour l'équation de Poisson.

La méthode a été généralisée aux équations dépendantes du temps en plus des deux variables spatiales.

---

# Remerciements

---

Tout d'abord nous remercions « ALLAH » Le tout puissant de nous avoir donné la santé, la volonté et le courage pour effectuer ce modeste travail.

Nous remercions nos parents pour leur contribution, leur soutien et leur patience.

Nous remercions notre encadreur *Mme ABLAOUI Naima* pour son aide, pour ses encouragements et sa présence durant la réalisation de notre travail .

Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Nous remercions les membres de jury, chacun à son nom, pour avoir accepté de juger notre travail.

Nos remerciements s'adressent également à *Mr.BELHAMITI.Omar*, pour sa générosité et sa grande patience.

Nous exprimons notre gratitude à tous ceux qui ont accepté de répondre à nos questions avec gentillesse comme *Melle HAMOU MAAMAR Maghnia*, et *Melle LADJAL Nouria*.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches, nos familles et amis, qui nous ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire.

Merci à toutes et à tous.

\*

UNIVERSITÉ ABDELHAMID IBN BADIS-MOSTAGANEM  
FACULTÉ DES SCIENCES EXACTES ET SCIENCES DE LA NATURE ET  
DE LAVIE  
DÉPARTEMENT DE MATHÉMATIQUES

Mémoire de Master

Spécialité : Modélisation, Contrôle et Optimisation

Thème

*Résolution des EDPs par la Méthode des Ondelettes de Legendre en 2D*

Présenté par

*M<sup>elle</sup> CHABOUNI Acia*

*M<sup>elle</sup> BENKAHLA Fatiha*

Soutenu le 05/2016

Devant le jury

Mr BOKHARI Ahmed	<b>Président</b>	M.A.A	<b>U. MEDIA.</b>
Mr BELHAMITI Omar	<b>Examineur</b>	M.C.A	<b>U. MOSTAGANEM.</b>
Mme ABLAOUI Naima	<b>Encadreur</b>	M.A.A	<b>U. MOSTAGANEM.</b>

# Table des matières

Liste des Tableaux	i
Liste des Figures	ii
Introduction	iii
<b>1 PRÉLIMINAIRE</b>	<b>1</b>
1.1 Les polynômes de Legendre . . . . .	1
1.2 Les ondelettes . . . . .	2
1.3 Les ondelettes de Legendre . . . . .	2
1.4 Approximation d'une fonction . . . . .	3
<b>2 RÉOLUTION D'UNE EDP EN 2D PAR LA MÉTHODE DES ONDE- LETTES DE LEGENDRE</b>	<b>6</b>
2.1 Les matrices opérationnelles d'intégration en deux dimensions . . . . .	6
2.1.1 La matrice opérationnelle d'intégration par rapport à $x$ . . . . .	6
2.1.2 Matrice opérationnelle d'intégration par rapport à $y$ . . . . .	9
2.2 Description de la méthode des ondelettes de Legendre . . . . .	11
2.2.1 Cas des conditions aux limites de type Dirichlet . . . . .	11
2.2.2 Cas des conditions aux limites de type mixte . . . . .	15
2.3 Résultats et tests numériques . . . . .	17
2.3.1 Équation de Poisson avec conditions de Dirichlet . . . . .	17

---

2.3.2	Équation de Poisson avec conditions mixtes . . . . .	20
<b>3</b>	<b>RÉSOLUTION D'UNE EDP EN 2D PAR LA MÉTHODE DES ONDO- LETTES DE LEGENDRE AVEC LE TEMPS</b>	<b>22</b>
	<b>Conclusion</b>	<b>29</b>
	<b>Bibliographie</b>	<b>30</b>
<b>4</b>	<b>Annexe</b>	<b>31</b>
4.1	Les sous programmes . . . . .	31
4.1.1	Polynôme de Legendre $px$ et $py$ . . . . .	31
4.1.2	L'ondelette de Legendre . . . . .	32
4.1.3	La matrice d'ondelette de Legendre . . . . .	32
4.1.4	Les matrices d'intégrations $P_x, P_y$ . . . . .	33
4.2	Le programme principal pour les conditions Dirichlet . . . . .	38
4.2.1	Les sous programmes . . . . .	41
4.3	Le programme principal pour les conditions mixtes . . . . .	44
4.3.1	Les sous programmes . . . . .	46

---

### Liste des Tableaux

---

**Tableau 1 :** Les résultats numériques pour le cas des conditions de Dirichlet.....(19)

**Tableau 2 :** Les résultats numériques pour le cas  $k = k' = 2$ .....(19)

**Tableau 3 :** Les résultats numériques pour le cas des conditions mixte.....(22)

**Tableau 4 :** Les résultats numériques pour le cas  $k = k' = 2$ .....(22)

### Liste des Figure

---

- Figure 1 :** Comparaison entre la solution approchée et exacte dans le cas de Dirichlet avec  $M = M' = 8$  .....(20)
- Figure 2 :** L'évolution du logarithme de l'erreur pour le cas Dirichlet avec  $k = k' = 2$  .....(20)
- Figure 3 :** Comparaison entre la solution approchée et exacte dans le cas mixte avec  $M = M' = 10$  .....(22)
- Figure 4 :** L'évolution du logarithme de l'erreur pour le cas mixte avec  $k = k' = 2$ .....(23)



---

# INTRODUCTION

---

Les équations différentielles ordinaires (EDO) et les équations aux dérivées partielles (EDP), sont à l'origine de la description de beaucoup de phénomènes naturels en physique, en chimie, en biologie ...

On dispose de plusieurs méthodes numériques pour résoudre ces équations, parmi lesquelles on cite la méthode des différences finies, la méthode des éléments finis ou alors les méthodes itératives.

Cela ne peut se faire sans une efficacité dans la programmation, car les systèmes réels sont très complexes, et nécessitent l'efficacité du choix de l'algorithme.

L'une des méthodes de résolution, qui commence à s'imposer, est la méthode des ondelettes de Legendre. Elle a été utilisée en 2000, par M. Razzaghi et S. Yousefi [3] pour la résolution des problèmes variationnelles en une dimension. Les mêmes auteurs l'ont appliqué pour résoudre un problème de contrôle optimal [4]. Elle a aussi été utilisée pour la résolution des équations intégral-différentielles comme l'équation intégrale de Fredholm [1].

Dans notre travail, on développe la méthode des ondelettes de Legendre en 2D, qui ont été définies en 2005 par H.Parsian [5]. Pour ce faire, on rappelle les matrices opérationnelles d'intégration notées dans Parsian  $P_x$  et  $P_y$ .

Une procédure calculatoire utilisant ces matrices est largement exposée pour résoudre les EDPs en les réduisant à des systèmes linéaires d'équations algébriques.

Ce travail est structuré de manière à présenter dans le premier chapitre quelques notions de base et outils utilisés tout au long de ce travail.

Le second chapitre est consacré à la résolution d'une EDP avec des conditions aux limites de type Dirichlet et de type mixtes. Des résultats numériques montrent l'efficacité de cette méthode.

Le chapitre qui suit est consacré à la résolution d'une EDP en 2D et en présence de la variable temps.

Enfin, on termine notre travail par une conclusion générale.

# PRÉLIMINAIRE

---

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

## 1.1 Les polynômes de Legendre

Ils sont introduits en 1784 par le mathématicien français Adrien-Marie Legendre(1752-1833)

**Définition 1.1.1** ( Voir [3]) *Les polynômes de Legendre sont définis sur  $[-1, 1]$  par la formule de récurrence suivante :*

$$\begin{cases} P_0(x) = 1, \\ P_1(x) = x, \\ P_{n+1}(x) = \frac{(2n+1)}{(n+1)}xP_n(x) - \frac{n}{(n+1)}P_{n-1}(x), n = 1, 2, \dots \end{cases} \quad (1.1.1)$$

Ils sont solution de l'équation différentielle, dite de Legendre, suivante

$$(1 - x^2)P_n'' - 2xP_n' + n(n + 1)P_n = [(1 - x^2)P_n']' + n(n + 1)P_n = 0, \quad n = 0, 1, 2, \dots$$

Leur forme analytique est donnée par

$$\forall m \in \mathbb{N}, \quad P_m(x) = \sum_{k=0}^m (-1)^{m+k} \frac{(m+k)!}{(m-k)! (k!)^2} x^k$$

**Proposition 1.1.1** *Les polynômes de Legendre vérifient la propriété très importante d'orthogonalité*

$$\int_{-1}^1 P_m(x)P_n(x)dx = \begin{cases} 0 & \text{si } m \neq n \\ \frac{2}{2m+1} & \text{si } m = n \end{cases} .$$

**Corollaire 1.1.1** soit  $x = 2^k\tau - 2n + 1$  et  $P_n$  le pôleynome de Legendre vérifie

$$2^k(2n + 1)P_n(\tau) = P'_{n+1}(\tau) - P'_{n-1}(\tau).$$

## 1.2 Les ondelettes

**Définition 1.2.1** Les ondelettes sont une famille de fonctions construites à partir de la dilatation et translation d'une fonction unique  $\Psi$  dans l'espace  $L^2(\mathbb{R})$ , appelée ondelette mère (Voir [3]). Elles sont définies par :

$$\begin{cases} \Psi_{a,b}(x) = |a|^{-\frac{1}{2}} \Psi\left(\frac{x-b}{a}\right) \\ a, b \in \mathbb{R}, a \neq 0. \end{cases}$$

**Remarque 1.2.1** On dit qu'une famille d'ondelettes est continue lorsque les paramètres de dilatation  $a$  et de translation  $b$  varient de façon continue.

On parle d'une famille d'ondelettes discrète, si les paramètres  $a$  et  $b$  prennent des valeurs discrètes.

La famille d'ondelettes discrète

$$\Psi_{n,m}(x) = |a_0|^{\frac{m}{2}} \Psi(a_0^m x - nb_0),$$

est obtenue pour les valeurs de

$$\begin{cases} a = a_0^{-m}, a_0 > 1 \\ b = nb_0 a_0^{-m}, b_0 > 1, \end{cases}$$

où  $n, m$  sont des entiers positifs.

## 1.3 Les ondelettes de Legendre

**Définition 1.3.1** Les ondelettes de Legendre  $\Psi_{n,m}(x) = \Psi(n, m, j, x)$  sont définies sur l'intervalle  $[0, 1]$  par

$$\Psi_{n,m}(x) = \begin{cases} \sqrt{m + \frac{1}{2}} 2^{\frac{j}{2}} P_m(2^j x - 2n + 1) & \text{si } \frac{n-1}{2^{j-1}} \leq x < \frac{n}{2^{j-1}}, \\ 0 & \text{sinon.} \end{cases} \quad (1.3.1)$$

Avec  $n = 1, 2, \dots, 2^{j-1}$  ( $j \in \mathbb{N}^*$ ), représente le nombre des niveaux de la décomposition de l'intervalle  $[0, 1]$  et  $m \geq 0$  où  $m$  est le degré du polynôme de Legendre (Voir [3]).

Le paramètre de dilatation  $a = 2^{-j}$  et le paramètre de translation  $b = (2n - 1)2^{-j}$ .

**Définition 1.3.2** Les ondelettes de Legendre en deux dimensions, introduites par [5], notées  $\Psi_{n,m,n',m'}(x, y) = \Psi(n, m, n', m', k, k', x, y)$  sont définies sur l'intervalle  $[0, 1] \times [0, 1]$  par

$$\Psi_{n,m,n',m'}(x, y) = \begin{cases} \sqrt{(m + \frac{1}{2})(m' + \frac{1}{2})} 2^{\frac{k+k'}{2}} \bar{P}_m(x) \bar{P}_{m'}(y), & \text{si } \frac{n-1}{2^{k-1}} \leq x < \frac{n}{2^{k-1}} \text{ et } \frac{n'-1}{2^{k'-1}} \leq y < \frac{n'}{2^{k'-1}} \\ 0 & \text{sinon.} \end{cases} \quad (1.3.2)$$

Avec  $n = 1, 2, \dots, 2^{k-1}$ ,  $n' = 1, 2, \dots, 2^{k'-1}$ ,  $m \geq 0$ ,  $m' \geq 0$ .

Où  $\bar{P}_m(x) = P_m(2^k x - 2n + 1)$ ,  $\bar{P}_{m'}(y) = P_{m'}(2^{k'} y - 2n' + 1)$ , avec  $P_m, P_{m'}$  sont les polynômes de Legendre de degrés  $m$  et  $m'$  respectivement.

## 1.4 Approximation d'une fonction

**Proposition 1.4.1** Toute fonction  $u(.,.) \in L^2([0, 1] \times [0, 1])$ , s'écrit sous la forme (Voir [2]) :

$$u(x, y) = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} \sum_{n'=1}^{\infty} \sum_{m'=0}^{\infty} C_{n,m,n',m'} \Psi_{n,m,n',m'}(x, y), \quad (1.4.1)$$

avec

$$C_{n,m,n',m'} = \langle u, \Psi_{n,m,n',m'} \rangle_{L^2([0,1] \times [0,1])} = \int_0^1 \int_0^1 u(x, y) \Psi_{n,m,n',m'}(x, y) dx dy.$$

**Remarque 1.4.1** La troncature de la série (1.4.1) conduit à

$$u(x, y) \simeq \sum_{n=1}^{2^{k-1}M-1} \sum_{m=0}^{2^{k-1}M-1} \sum_{n'=1}^{2^{k'-1}M'-1} \sum_{m'=0}^{2^{k'-1}M'-1} C_{n,m,n',m'} \Psi_{n,m,n',m'}(x, y)$$

qu'on peut exprimer sous la forme

$$u(x, y) = C^T \Psi(x, y) \quad (1.4.2)$$

où  $C$  et  $\Psi(x, y)$  sont deux vecteurs de dimension  $2^{k-1}2^{k'-1}MM' \times 1$  donnés par

$$C = \begin{bmatrix} c_{1,0,1,0}, \dots, c_{1,0,1,M'-1}, c_{1,0,2,0}, \dots, c_{1,0,2,M'-1}, \dots, c_{1,0,2^{k'-1},0}, \dots, c_{1,0,2^{k'-1},M'-1}, \\ \dots, c_{1,M-1,2^{k'-1},0}, \dots, c_{1,M-1,2^{k'-1},M'-1}, c_{2,0,1,0}, \dots, c_{2,0,1,M'-1}, c_{2,0,2,0}, \dots, c_{2,0,2,M'-1}, \\ \dots, c_{2,0,2^{k'-1},M'-1}, \dots, c_{2,M-1,1,0}, \dots, c_{2,M-1,2^{k'-1},M'-1}, \dots, c_{2^{k-1},0,1,0}, \dots, c_{2^{k-1},0,1,M'-1}, \\ c_{2^{k-1},0,2,0}, \dots, c_{2^{k-1},0,2,M'-1}, \dots, c_{2^{k-1},0,2^{k'-1},0}, \dots, c_{2^{k-1},M-1,2^{k'-1},M'-1} \end{bmatrix}^T \quad (1.4.3)$$

$$\Psi = \begin{bmatrix} \Psi_{1,0,1,0}, \dots, \Psi_{1,0,1,M'-1}, \Psi_{1,0,2,0}, \dots, \Psi_{1,0,2,M'-1}, \dots, \Psi_{1,0,2^{k'-1},0}, \dots, \Psi_{1,0,2^{k'-1},M'-1}, \\ \dots, \Psi_{1,M-1,2^{k'-1},0}, \dots, \Psi_{1,M-1,2^{k'-1},M'-1}, \Psi_{2,0,1,0}, \dots, \Psi_{2,0,1,M'-1}, \Psi_{2,0,2,0}, \dots, \Psi_{2,0,2,M'-1}, \\ \dots, \Psi_{2,0,2^{k'-1},M'-1}, \dots, \Psi_{2,M-1,1,0}, \dots, \Psi_{2,M-1,1,M'-1}, \Psi_{2,M-1,2,0}, \dots, \Psi_{2,M-1,2^{k'-1},M'-1}, \\ \dots, \Psi_{2^{k-1},0,1,0}, \dots, \Psi_{2^{k-1},0,1,M'-1}, \Psi_{2^{k-1},0,2,0}, \dots, \Psi_{2^{k-1},0,2,M'-1}, \dots, \Psi_{2^{k-1},M-1,2^{k'-1},M'-1} \end{bmatrix}^T \quad (1.4.4)$$

**Exemple 1.4.1** *Pour illustration, on peut approximer la fonction  $f(x, y) = 1$  dans la base d'ondelettes de Legendre.*

En effet,

$$1 \approx \sum_{n=1}^{2^{k-1}M-1} \sum_{m=0}^{2^{k'-1}M'-1} \sum_{n'=1}^{2^{k-1}M-1} \sum_{m'=0}^{2^{k'-1}M'-1} C_{n,m,n',m'} \Psi_{n,m,n',m'}(x, y) = C^T \Psi(x, y),$$

avec

$$C_{n,m,n',m'} = \langle 1, \Psi_{n,m,n',m'} \rangle_{L^2([0,1] \times [0,1])} = \int_0^1 \int_0^1 \Psi_{n,m,n',m'}(x, y) dx dy.$$

Si on prend à titre d'exemple  $k = k' = 2$  et  $M = M' = 2$ , on obtient les ondelettes qui interviennent dans le calcul des coefficients  $C_{n,m,n',m'}$  et qui sont :

$$\left. \begin{array}{l} \Psi_{1,0,1,0}(x, y) = 2 \\ \Psi_{1,0,1,1}(x, y) = 2\sqrt{3}(4y - 1) \\ \Psi_{1,1,1,0}(x, y) = 2\sqrt{3}(4x - 1) \\ \Psi_{1,1,1,1}(x, y) = 6(4x - 1)(4y - 1) \end{array} \right\} \text{pour } 0 \leq x, y < \frac{1}{2}$$

$$\left. \begin{array}{l} \Psi_{1,0,2,0}(x, y) = 2 \\ \Psi_{1,0,2,1}(x, y) = 2\sqrt{3}(4y - 1) \\ \Psi_{1,1,2,0}(x, y) = 2\sqrt{3}(4x - 1) \\ \Psi_{1,1,2,1}(x, y) = 6(4x - 1)(4y - 3) \end{array} \right\} \text{pour } 0 \leq x < \frac{1}{2}, \frac{1}{2} \leq y \leq 1$$

$$\left. \begin{array}{l} \Psi_{2,0,1,0}(x, y) = 2 \\ \Psi_{2,0,1,1}(x, y) = 2\sqrt{3}(4y - 1) \\ \Psi_{2,1,1,0}(x, y) = 2\sqrt{3}(4x - 3) \\ \Psi_{2,1,1,1}(x, y) = 6(4x - 3)(4y - 1) \end{array} \right\} \text{pour } \frac{1}{2} \leq x \leq 1, 0 \leq y \leq \frac{1}{2}$$

$$\left. \begin{array}{l} \Psi_{2,0,2,0}(x, y) = 2 \\ \Psi_{2,0,2,1}(x, y) = 2\sqrt{3}(4y - 3) \\ \Psi_{2,1,2,0}(x, y) = 2\sqrt{3}(4x - 3) \\ \Psi_{2,1,2,1}(x, y) = 6(4x - 1)(4y - 3) \end{array} \right\} \text{pour } \frac{1}{2} \leq x, y \leq 1$$

Calcul des coefficients

$$\left\{ \begin{array}{l} C_{1,0,1,0} = \int_0^1 \int_0^1 \Psi_{1,0,1,0}(x,y) dx dy = \frac{1}{2}, \quad C_{1,0,1,1} = \int_0^1 \int_0^1 \Psi_{1,0,1,1}(x,y) dx dy = 0 \\ C_{1,1,1,0} = \int_0^1 \int_0^1 \Psi_{1,1,1,0}(x,y) dx dy = 0, \quad C_{1,1,1,1} = \int_0^1 \int_0^1 \Psi_{1,1,1,1}(x,y) dx dy = 0 \\ C_{1,0,2,0} = \int_0^1 \int_0^1 \Psi_{1,0,2,0}(x,y) dx dy = \frac{1}{2}, \quad C_{1,0,2,1} = \int_0^1 \int_0^1 \Psi_{1,0,2,1}(x,y) dx dy = 0 \\ C_{1,1,2,0} = \int_0^1 \int_0^1 \Psi_{1,1,2,0}(x,y) dx dy = 0, \quad C_{1,1,2,1} = \int_0^1 \int_0^1 \Psi_{1,1,2,1}(x,y) dx dy = 0 \\ C_{2,0,1,0} = \int_0^1 \int_0^1 \Psi_{2,0,1,0}(x,y) dx dy = 0, \quad C_{2,0,1,1} = \int_0^1 \int_0^1 \Psi_{2,0,1,1}(x,y) dx dy = 0 \\ C_{2,1,1,0} = \int_0^1 \int_0^1 \Psi_{2,1,1,0}(x,y) dx dy = 0, \quad C_{2,1,1,1} = \int_0^1 \int_0^1 \Psi_{2,1,1,1}(x,y) dx dy = 0 \\ C_{2,0,2,0} = \int_0^1 \int_0^1 \Psi_{2,0,2,0}(x,y) dx dy = \frac{1}{2}, \quad C_{2,0,2,1} = \int_0^1 \int_0^1 \Psi_{2,0,2,1}(x,y) dx dy = 0 \\ C_{2,1,2,0} = \int_0^1 \int_0^1 \Psi_{2,1,2,0}(x,y) dx dy = 0, \quad C_{2,1,2,1} = \int_0^1 \int_0^1 \Psi_{2,1,2,1}(x,y) dx dy = 0 \end{array} \right. ,$$

donc la fonction  $f(x, y) = 1$ , se décompose

$$1 = \left[ \frac{1}{2}, 0, \frac{1}{2}, 0, 0, 0, 0, 0, \frac{1}{2}, 0, \frac{1}{2}, 0, 0, 0, 0, 0 \right] \Psi(x, y).$$

Avec les mêmes calculs on peut trouver les résultats de décomposition des fonctions  $f(x, y) = x$  et  $f(x, y) = xy$

$$x = \left[ \frac{1}{8}, 0, \frac{1}{8}, \frac{\sqrt{3}}{4}, \frac{\sqrt{3}}{24}, 0, \frac{\sqrt{3}}{24}, 0, \frac{3}{8}, 0, \frac{3}{8}, 0, \frac{\sqrt{3}}{24}, 0, \frac{\sqrt{3}}{24}, 0 \right] \Psi(x, y),$$

$$xy = \left[ \frac{1}{32}, \frac{\sqrt{3}}{96}, \frac{3}{32}, \frac{19\sqrt{3}}{96}, \frac{\sqrt{3}}{96}, \frac{1}{96}, \frac{\sqrt{3}}{32}, \frac{1}{96}, \frac{3}{32}, \frac{\sqrt{3}}{32}, \frac{9}{32}, \frac{\sqrt{3}}{32}, \frac{\sqrt{3}}{96}, \frac{1}{96}, \frac{\sqrt{3}}{32}, \frac{19}{96} \right] \Psi(x, y).$$

La méthode de résolution des EDPs, qu'on propose dans le chapitre suivant, est basée sur la décomposition en ondelettes de Legendre (1.4.2).

# RÉSOLUTION D'UNE EDP EN 2D PAR LA MÉTHODE DES ONDELETTES DE LEGENDRE

---

Lors de la résolution d'une EDP par la méthode des ondelettes de Legendre, on aura besoin d'introduire les matrices opérationnelles d'intégration, notées  $P_x, P_y$ , (Voir [6]). On commencera ce chapitre par le calcul de ces matrices.

## 2.1 Les matrices opérationnelles d'intégration en deux dimensions

### 2.1.1 La matrice opérationnelle d'intégration par rapport à $x$

**Théorème 2.1.1** *Soit  $\Psi(x, y)$  la famille d'ondelettes de Legendre en deux dimensions définie en (1.4.4).*

*La matrice opérationnelle d'intégration  $P_x$  (Voir [5]) est entièrement déterminée par la formule :*

$$\int_0^x \Psi(\tau, y) d\tau = P_x \Psi(x, y). \quad (2.1.1)$$

*La matrice  $P_x$  est de dimension  $2^{k-1}2^{k'-1}MM' \times 2^{k-1}2^{k'-1}MM'$  et est donnée par :*

$$P_x = \frac{1}{2^k} \begin{bmatrix} L & F & F & \dots & F \\ O & L & F & \dots & F \\ O & O & L & \dots & F \\ \vdots & \vdots & \vdots & \ddots & F \\ O & O & O & \dots & L \end{bmatrix}. \quad (2.1.2)$$

Où  $O$ ,  $L$  et  $F$  sont des matrices de taille  $2^{k'-1}MM' \times 2^{k'-1}MM'$ ,  $O$  est la matrice nulle et  $L$  et  $F$  sont données par

$$F = \begin{bmatrix} 2D & O' & O' & \dots & O' \\ O' & O' & O' & \dots & O' \\ O' & O' & O' & \dots & O' \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ O' & O' & O' & \dots & O' \end{bmatrix} \quad (2.1.3)$$

$$L = \begin{bmatrix} D & \frac{1}{\sqrt{3}}D & O' & \dots & \dots & O' \\ -\frac{1}{\sqrt{3}}D & O' & \frac{1}{\sqrt{3}\sqrt{5}}D & \dots & \dots & \vdots \\ O' & -\frac{1}{\sqrt{3}\sqrt{5}}D & O' & \ddots & \dots & O' \\ \vdots & \dots & \dots & \ddots & \dots & O' \\ O' & \dots & \dots & \ddots & O' & \frac{\sqrt{2M-3}}{(2M-3)\sqrt{2M-1}}D \\ O' & \dots & \dots & O' & -\frac{\sqrt{2M-1}}{(2M-1)\sqrt{2M-3}}D & O' \end{bmatrix}. \quad (2.1.4)$$

$D$  et  $O'$  sont des matrices de taille  $2^{k'-1}M' \times 2^{k'-1}M'$  avec  $O'$  la matrice nulle et  $D$  la matrice identité.

**Preuve.** à partir de l'intégrale  $\int_0^x \Psi(\tau, y) d\tau = P_x \Psi(x, y)$ , on a

pour  $m = 0$ , on obtient

$$\int_0^x \Psi_{n,0,n',m'}(\tau, y) d\tau = \begin{cases} \frac{1}{2^k} \left[ \frac{1}{\sqrt{3}} \Psi_{n,1,n',m'}(x, y) + \Psi_{n,0,n',m'}(x, y) \right], & x \in \left[ \frac{n-1}{2^{k-1}}, \frac{n}{2^{k-1}} \right]; \\ 0, & \hat{n} = 1, \dots, n-1, x \in \left[ \frac{\hat{n}-1}{2^{k-1}}, \frac{\hat{n}}{2^{k-1}} \right]; \\ \frac{1}{2^{k-1}} \Psi_{\hat{n},0,n',m'}(x, y), & \hat{n} = n+1, \dots, 2^{k-1}, x \in \left[ \frac{\hat{n}-1}{2^{k-1}}, \frac{\hat{n}}{2^{k-1}} \right]; \end{cases}$$

pour  $m \neq 0$ , on obtient

$$\int_0^x \Psi_{n,m,n',m'}(\tau, y) d\tau = \begin{cases} \frac{1}{2^k \sqrt{2m+1}} \left[ \frac{1}{\sqrt{2m+3}} \Psi_{n,m+1,n',m'}(x, y) - \frac{1}{\sqrt{2m-1}} \Psi_{n,m-1,n',m'}(x, y) \right], & \\ x \in \left[ \frac{n-1}{2^{k-1}}, \frac{n}{2^{k-1}} \right]; & \\ 0, & \text{ailleurs} \end{cases} \quad \square$$

**Exemple 2.1.1** On prend  $M = M' = 3$  et  $k = k' = 2$  pour le calcul de cette matrice pour  $m = 0$

$$\begin{aligned} \int_0^x \Psi_{1010}(\tau, y) d\tau &= \begin{cases} \frac{1}{2^2} \left( \frac{1}{\sqrt{3}} \Psi_{1110}(x, y) + \Psi_{1010}(x, y) \right) & x \in \left[ 0, \frac{1}{2} \right] \\ \frac{1}{2} \Psi_{2010}(x, y) & x \in \left[ \frac{1}{2}, 1 \right] \\ 0 & \text{ailleurs} \end{cases} \\ &= \frac{1}{2^2} \left[ 1, 0, 0, 0, 0, 0, \frac{1}{\sqrt{3}}, 0, \dots, 0, 2, 0, \dots, 0, 0 \right]^T \Psi(x, y), \end{aligned}$$



pour  $m \neq 0$

$$\begin{aligned} \int_0^x \Psi_{1110}(\tau, y) d\tau &= \begin{cases} \frac{1}{2^2} \frac{1}{\sqrt{3}} \left( \frac{1}{\sqrt{5}} \Psi_{1210}(x, y) - \Psi_{1010}(x, y) \right) & x \in \left[0, \frac{1}{2}\right] \\ 0 & \text{ailleurs} \end{cases} \\ &= \frac{1}{2^2} \left[ -1, 0, 0, \dots, 0, 0, \frac{1}{\sqrt{3}\sqrt{5}}, 0, 0, \dots, 0, 0 \right]^T \Psi(x, y). \end{aligned}$$

$$P_x = \frac{1}{2^2} \begin{bmatrix} L_{18 \times 18} & F_{18 \times 18} \\ O_{18 \times 18} & L_{18 \times 18} \end{bmatrix}, \text{ et } L_{18 \times 18} = \begin{bmatrix} D & \frac{1}{\sqrt{3}}D & 0 \\ -\frac{1}{\sqrt{3}}D & 0 & \frac{1}{\sqrt{3}\sqrt{5}}D \\ 0 & -\frac{1}{\sqrt{3}\sqrt{5}}D & 0 \end{bmatrix},$$

$$F_{18 \times 18} = \begin{bmatrix} 2D & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

**Proposition 2.1.1** *La matrice opérationnelle d'intégration  $P_x$ , définie par (2.1.2)-(2.1.4), est inversible.*

**Preuve.** Il suffit de démontrer que le déterminant de la matrice  $P_x$  est non nul. Comme  $P_x$  est triangulaire par blocs, son déterminant est le produit des déterminants des blocs diagonaux, alors il suffit que le déterminant de la matrice  $L$  soit non nul.

En effet,

$$\det(L) = (-1)^{2^{k'-1}M'} \prod_{m=1}^M \left( \frac{1}{2m-1} \right)^{2^{k'-1}M'} \neq 0.$$

Ce résultat se vérifie par récurrence □

**Exemple 2.1.2** *Calcul du déterminant de  $L$  pour  $M = M' = 3$  et  $k = k' = 2$*

$$\begin{aligned} \det \begin{bmatrix} D & \frac{1}{\sqrt{3}}D & O' \\ -\frac{1}{\sqrt{3}}D & O' & \frac{1}{\sqrt{3}\sqrt{5}}D \\ O' & -\frac{1}{\sqrt{3}\sqrt{5}}D & O' \end{bmatrix} &= \det \left( -\frac{1}{\sqrt{3}\sqrt{5}}D \right) \left[ \det(D) \det \left( \frac{1}{\sqrt{3}\sqrt{5}}D \right) \right] \\ &= (-1)^6 \left( \frac{1}{3} \right)^6 \left( \frac{1}{5} \right)^6 \\ &= \left( \frac{1}{3} \right)^6 \left( \frac{1}{5} \right)^6 \end{aligned}$$

### 2.1.2 Matrice opérationnelle d'intégration par rapport à $y$

**Théorème 2.1.2** Soit  $\Psi(x, y)$  la famille d'ondelettes de Legendre en deux dimensions définie en (1.4.4).

La matrice opérationnelle d'intégration  $P_y$  (Voir [5]) est entièrement déterminée par la formule :

$$\int_0^y \Psi(x, \tau) d\tau = P_y \Psi(x, y). \quad (2.1.5)$$

La matrice  $P_y$  est de dimension  $2^{k-1}2^{k'-1}MM' \times 2^{k-1}2^{k'-1}MM'$  et est donnée par :

$$P_y = \frac{1}{2^{k'}} \begin{bmatrix} P & O & O & \dots & O \\ O & P & O & \dots & O \\ O & O & P & \dots & O \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ O & O & O & \dots & P \end{bmatrix}. \quad (2.1.6)$$

Où  $O$ , et  $P$  sont des matrices de taille  $2^{k-1}MM' \times 2^{k-1}MM'$ ,  $O$  est la matrice nulle et  $P$  est donnée par :

$$P = \begin{bmatrix} L & F & F & \dots & F \\ O' & L & F & \dots & F \\ O' & O' & L & \dots & F \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ O' & O' & O' & \dots & L \end{bmatrix}, \quad (2.1.7)$$

$F$ ,  $L$  et  $O'$  sont des matrices de taille  $M' \times M'$  avec  $O'$  la matrice nulle et  $F$  et  $L$  sont définies par :

$$F = \begin{bmatrix} 2 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad L = \begin{bmatrix} 1 & \frac{1}{\sqrt{3}} & 0 & \dots & 0 \\ -\frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{3}\sqrt{5}} & \dots & 0 \\ 0 & -\frac{1}{\sqrt{3}\sqrt{5}} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & -\frac{\sqrt{2M'-1}}{(2M'-1)\sqrt{2M'-3}} & \frac{\sqrt{2M'-1}}{(2M'-1)\sqrt{2M'-3}} \\ & & & & 0 \end{bmatrix} \quad (2.1.8)$$

**Preuve.** À partir de l'intégrale  $\int_0^y \Psi(x, \tau) d\tau = P_y \Psi(x, y)$  on a

pour  $m' = 0$ , on obtient

$$\int_0^y \Psi_{n,m,n',0}(x, \tau) d\tau = \begin{cases} \frac{1}{2^{k'}} \left[ \frac{1}{\sqrt{3}} \Psi_{n,m,n',1}(x, y) + \Psi_{n,m,n',0}(x, y) \right], & y \in \left[ \frac{n'-1}{2^{k'-1}}, \frac{n'}{2^{k'-1}} \right]; \\ 0, & \hat{n}' = 1, \dots, n-1, y \in \left[ \frac{\hat{n}'-1}{2^{k'-1}}, \frac{\hat{n}'}{2^{k'-1}} \right]; \\ \frac{1}{2^{k-1}} \Psi_{n,m,\hat{n}',0}(x, y), & \hat{n}' = n'+1, \dots, 2^{k'-1}, y \in \left[ \frac{\hat{n}'-1}{2^{k'-1}}, \frac{\hat{n}'}{2^{k'-1}} \right]; \end{cases}$$

pour  $m' \neq 0$ , on obtient

$$\int_0^y \Psi_{n,m,n',m'}(x, \tau) d\tau = \begin{cases} \frac{1}{2^{k'}\sqrt{2m'+1}} \left[ \frac{1}{\sqrt{2m'+3}} \Psi_{n,m,n',m'+1}(x, y) - \frac{1}{\sqrt{2m'-1}} \Psi_{n,m,n',m'-1}(x, y) \right], \\ y \in \left[ \frac{n'-1}{2^{k'-1}}, \frac{n'}{2^{k'-1}} \right]; \\ 0, & \text{ailleurs} \end{cases} .$$

□

**Exemple 2.1.3** On prend  $M = M' = 3$  et  $k = k' = 2$  pour le calcul de cette matrice,

pour  $m' = 0$

$$\begin{aligned} \int_0^y \Psi_{1110}(x, \tau) d\tau &= \begin{cases} \frac{1}{2^2} \left( \frac{1}{\sqrt{3}} \Psi_{1111}(x, y) + \Psi_{1110}(x, y) \right) & y \in \left[ 0, \frac{1}{2} \right] \\ 0 & \text{ailleurs} \\ \frac{1}{2} \Psi_{1120}(x, y) & y \in \left[ 0, \frac{1}{2} \right] \end{cases} \\ &= \frac{1}{2^2} \left[ 0, 0, 0, 0, 0, 0, 1, \frac{1}{\sqrt{3}}, 0, 2, 0, 0, 0, 0, 0, 0, \dots, 0, 0 \right]^T \Psi(x, y), \end{aligned}$$

pour  $m' \neq 0$

$$\begin{aligned} \int_0^y \Psi_{1211}(x, \tau) d\tau &= \begin{cases} \frac{1}{2^2} \frac{1}{\sqrt{3}} \left( \frac{1}{\sqrt{5}} \Psi_{1212}(x, y) - \Psi_{1210}(x, y) \right) & y \in \left[ 0, \frac{1}{2} \right] \\ 0 & \text{ailleurs} \end{cases} \\ &= \frac{1}{2^2} \left[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -\frac{1}{\sqrt{3}}, 0, \frac{1}{\sqrt{3}\sqrt{5}}, 0, \dots, 0, 0 \right]^T \Psi(x, y). \end{aligned}$$

$$P_y = \frac{1}{2^2} \begin{bmatrix} P & O & O & O & O & O \\ O & P & O & O & O & O \\ O & O & P & O & O & O \\ O & O & O & P & O & O \\ O & O & O & O & P & O \\ O & O & O & O & O & P \end{bmatrix}, \text{ et } P_{6 \times 6} = \begin{bmatrix} L_{3 \times 3} & F_{3 \times 3} \\ O'_{3 \times 3} & L_{3 \times 3} \end{bmatrix},$$

$$L_{3 \times 3} = \begin{bmatrix} 1 & \frac{1}{\sqrt{3}} & 0 \\ -\frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{3}\sqrt{5}} \\ 0 & -\frac{1}{\sqrt{3}\sqrt{5}} & 0 \end{bmatrix}, \quad F_{3 \times 3} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

**Proposition 2.1.2** La matrice opérationnelle d'intégration  $P_y$ , définie par (2.1.6)-(2.1.8), est inversible.

**Preuve.** Il suffit de démontrer que le déterminant de la matrice  $P_y$  est non nulle.  $P_y$  étant une matrice diagonale, Il suffit de vérifier le déterminant de la matrice  $P$  est non nulle. Or la matrice  $P$  est triangulaire par blocs avec le bloc  $L$  sur sa diagonale. Comme

$$\det(L) = \prod_{m=1}^{M'} \left( \frac{1}{2m-1} \right) \neq 0$$

cela implique que  $\det(P) \neq 0$  par suite  $\det(P_y) \neq 0$ . □

## 2.2 Description de la méthode des ondelettes de Legendre

On considère la forme complète d'une EDP d'ordre 2 à coefficients constants

$$a_1 \frac{\partial u}{\partial x}(x, y) + a_2 \frac{\partial u}{\partial y}(x, y) + a_3 \frac{\partial^2 u}{\partial x^2}(x, y) + a_4 \frac{\partial^2 u}{\partial y^2}(x, y) + a_5 \frac{\partial^2 u}{\partial x \partial y}(x, y) + a_6 u(x, y) = f(x, y) \quad (2.2.1)$$

### 2.2.1 Cas des conditions aux limites de type Dirichlet

$$\begin{cases} u(0, y) = u_G(y) & u(x, 0) = u_B(x) \\ u(1, y) = u_D(y) & u(x, 1) = u_H(x) \end{cases} \quad (2.2.2)$$

Les fonctions  $u_G$ ,  $u_B$ ,  $u_D$  et  $u_H$  sont supposées régulières sur  $[0, 1]$ .

On utilise la décomposition à base d'ondelettes de Legendre (Voir [6]) pour résoudre (2.2.1).

On pose

$$\frac{\partial^4 u}{\partial x^2 \partial y^2}(x, y) = C^T \Psi(x, y). \quad (2.2.3)$$

Différentes intégrations de la formule (2.2.3), permettent d'obtenir l'expression des différents termes de l'EDP (2.2.1),  $\frac{\partial u}{\partial x}(x, y)$ ,  $\frac{\partial u}{\partial y}(x, y)$ ,  $\frac{\partial^2 u}{\partial x^2}(x, y)$ ,  $\frac{\partial^2 u}{\partial y^2}(x, y)$ ,  $\frac{\partial^2 u}{\partial x \partial y}(x, y)$  et  $u(x, y)$ , en fonction de  $C$  (qui devient l'inconnue),  $\Psi(x, y)$ ,  $P_x$ ,  $P_y$  et des conditions aux limites.

Ainsi, en intégrant la formule (2.2.3) deux fois par rapport à  $x$  sur  $[0, x]$  on obtient :

$$\int_0^x \frac{\partial^4 u}{\partial x^2 \partial y^2}(\tau, y) d\tau = \int_0^x C^T \Psi(\tau, y) d\tau$$

$$\frac{\partial^3 u}{\partial x \partial y^2}(x, y) = C^T P_x \Psi(x, y) + \frac{\partial^3 u}{\partial x \partial y^2}(0, y), \quad (2.2.4)$$

$$\frac{\partial^2 u}{\partial y^2}(x, y) = C^T P_x^2 \Psi(x, y) + x \frac{\partial^3 u}{\partial x \partial y^2}(0, y) + \frac{\partial^2 u}{\partial y^2}(0, y). \quad (2.2.5)$$

Le terme  $\frac{\partial^3 u}{\partial x \partial y^2}(0, y)$ , est une inconnue, pour le déterminer, on pose  $x = 1$  dans (2.2.5) on obtient :

$$\frac{\partial^3 u}{\partial x \partial y^2}(0, y) = \frac{\partial^2 u}{\partial y^2}(1, y) - C^T P_x^2 \Psi(1, y) - \frac{\partial^2 u}{\partial y^2}(0, y). \quad (2.2.6)$$

En remplaçant (2.2.6) dans (2.2.5) on obtient :

$$\frac{\partial^2 u}{\partial y^2}(x, y) = C^T P_x^2 (\Psi(x, y) - x\Psi(1, y)) + x \left( \frac{\partial^2 u}{\partial y^2}(1, y) - \frac{\partial^2 u}{\partial y^2}(0, y) \right) + \frac{\partial^2 u}{\partial y^2}(0, y).$$

En prenant en considération les conditions aux limites, on a :

$$\frac{\partial^2 u}{\partial y^2}(x, y) = C^T P_x^2 (\Psi(x, y) - x\Psi(1, y)) + x(u_D''(y) - u_G''(y)) + u_G''(y). \quad (2.2.7)$$

On intègre (2.2.7) deux fois par rapport à  $y$  sur  $[0, y]$  on obtient

$$\begin{aligned} \frac{\partial u}{\partial y}(x, y) &= C^T P_x^2 P_y (\Psi(x, y) - x\Psi(1, y)) + u_G'(y) - u_G'(0) \\ &+ x(u_D'(y) - u_D'(0) - u_G'(y) + u_G'(0)) + \frac{\partial u}{\partial y}(x, 0), \end{aligned} \quad (2.2.8)$$

et

$$\begin{aligned} u(x, y) &= C^T P_x^2 P_y^2 (\Psi(x, y) - x\Psi(1, y)) + u_G(y) - u_G(0) - yu_G'(0) + y\frac{\partial u}{\partial y}(x, 0) \\ &+ x(u_D(y) - u_D(0) - yu_D'(0) - u_G(y) + u_G(0) + yu_G'(0)) + u_B(x) \end{aligned} \quad (2.2.9)$$

Pour l'obtention du terme  $\frac{\partial^2 u}{\partial x \partial y}(x, y)$ , on remplace (2.2.6) dans (2.2.4) et on intègre par rapport à  $y$  sur  $[0, y]$

$$\frac{\partial^3 u}{\partial x \partial y^2}(x, y) = C^T P_x (\Psi(x, y) - P_x \Psi(1, y)) + u_D''(y) - u_G''(y),$$

$$\frac{\partial^2 u}{\partial x \partial y}(x, y) = C^T P_x P_y (\Psi(x, y) - P_x \Psi(1, y)) + u_D'(y) - u_D'(0) - u_G'(y) + u_G'(0) + \frac{\partial^2 u}{\partial x \partial y}(x, 0). \quad (2.2.10)$$

En intégrant la formule (2.2.3) deux fois par rapport à  $y$  sur  $[0, y]$  on obtient :

$$\int_0^y \frac{\partial^4 u}{\partial x^2 \partial y^2}(x, \tau) d\tau = \int_0^y C^T \Psi(x, \tau) d\tau$$

$$\frac{\partial^3 u}{\partial x^2 \partial y}(x, y) = C^T P_y \Psi(x, y) + \frac{\partial^3 u}{\partial x^2 \partial y}(0, y), \quad (2.2.11)$$

$$\frac{\partial^2 u}{\partial x^2}(x, y) = C^T P_y^2 \Psi(x, y) + y \frac{\partial^3 u}{\partial x^2 \partial y}(x, 0) + \frac{\partial^2 u}{\partial x^2}(x, 0). \quad (2.2.12)$$

Le terme  $\frac{\partial^3 u}{\partial x^2 \partial y}(x, 0)$ , est une inconnue, pour le déterminer, on pose  $y = 1$  et en prenant en considération les conditions aux limites dans (2.2.12) on obtient :

$$\frac{\partial^3 u}{\partial x^2 \partial y}(x, 0) = \frac{\partial^2 u_H}{\partial x^2}(x) - C^T P_y^2 \Psi(x, 1) - u_B''(x). \quad (2.2.13)$$

On injecte (2.2.13) dans (2.2.12), on obtient la formule finale de  $\frac{\partial^2 u}{\partial x^2}(x, y)$

$$\frac{\partial^2 u}{\partial x^2}(x, y) = C^T P_y^2 (\Psi(x, y) - y \Psi(x, 1)) + y(u_H''(x) - u_B''(x)) + u_B''(x). \quad (2.2.14)$$

On intègre (2.2.14) on obtient :

$$\begin{aligned} \frac{\partial u}{\partial x}(x, y) &= C^T P_y^2 P_x (\Psi(x, y) - y \Psi(x, 1)) + u_B'(x) - u_B'(0) \\ &\quad + y(u_H'(x) - u_H'(0) - u_B'(x) + u_B'(0)) + \frac{\partial u}{\partial x}(0, y). \end{aligned} \quad (2.2.15)$$

**Remarque 2.2.1** Les termes  $\frac{\partial u}{\partial y}(x, 0)$ ,  $\frac{\partial^2 u}{\partial x \partial y}(0, y)$ ,  $\frac{\partial^2 u}{\partial x \partial y}(x, 0)$  et  $\frac{\partial u}{\partial x}(0, y)$  sont des inconnus, on appliquera des schémas aux différences finies par rapport à  $x$  et à  $y$  pour les estimer lors de la résolution du système final.

Finalement en remplaçant maintenant (2.2.7) et (2.2.8), (2.2.9), (2.2.10), (2.2.14), (2.2.15) dans (2.2.1) on obtient

$$\left\{ \begin{array}{l} a_1 [C^T P_y^2 P_x (\Psi(x, y) - y \Psi(x, 1)) + y(u_H'(x) - u_H'(0) - u_B'(x) + u_B'(0)) \\ + u_B'(x) - u_B'(0) + \frac{\partial u}{\partial x}(0, y)] + a_2 [C^T P_x^2 P_y (\Psi(x, y) - x \Psi(1, y)) + x(u_D'(y) - u_D'(0)) \\ - u_G'(y) + u_G'(0) + u_G'(y) - u_G'(0) + \frac{\partial u}{\partial y}(x, 0)] + a_3 [C^T P_y^2 (\Psi(x, y) - y \Psi(x, 1)) \\ + y(u_H''(x) - u_B''(x)) + u_B''(x)] + a_4 [C^T P_x^2 (\Psi(x, y) - x \Psi(1, y)) + x(u_D''(y) \\ - u_G''(y)) + u_G''(y)] + a_5 [C^T P_x P_y (\Psi(x, y) - P_x \Psi(1, y)) + u_D'(y) - u_D'(0) \\ - u_G'(y) + u_G'(0) + \frac{\partial^2 u}{\partial x \partial y}(x, 0)] + a_6 [C^T P_x^2 P_y^2 (\Psi(x, y) - x \Psi(1, y)) + x(u_D(y) - u_D(0)) \\ - y u_D'(0) - u_G(y) + u_G(0) + y u_G'(0) + u_G(y) - u_G(0) - y u_G'(0) + y \frac{\partial u}{\partial y}(x, 0) \\ + u_B(x)] = f(x, y) \end{array} \right.$$

⇔

$$\left\{ \begin{array}{l} C^T [(a_1 P_y^2 P_x + a_2 P_x^2 P_y + a_3 P_y^2 + a_4 P_x^2 + a_5 P_x P_y + a_6 P_x^2 P_y^2) \Psi(x, y) - \\ (a_1 y P_y^2 P_x + a_3 y P_y^2) \Psi(x, 1) - (a_2 x P_x^2 P_y + a_4 x P_x^2 + a_5 P_x P_y P_x + a_6 x P_x^2 P_y^2) \Psi(1, y)] \\ = f(x, y) - [a_1 (y(u'_H(x) - u'_H(0) - u'_B(x) + u'_B(0)) + u'_B(x) - u'_B(0) + \frac{\partial u}{\partial x}(0, y)) \\ + a_2 (x(u'_D(y) - u'_D(0) - u'_G(y) + u'_G(0)) + u'_G(y) - u'_G(0) + \frac{\partial u}{\partial y}(x, 0)) \\ + a_3 (y(u''_H(x) - u''_B(x)) + u''_B(x)) + a_4 (x(u''_D(y) - u''_G(y)) + u''_G(y)) + a_5 (u'_D(y) \\ - u'_D(0) - u'_G(y) + u'_G(0) + \frac{\partial^2 u}{\partial x \partial y}(x, 0)) + a_6 (x(u_D(y) - u_D(0) - y u'_D(0) - \\ u_G(y) + u_G(0) + y u'_G(0)) + u_G(y) - u_G(0) - y u'_G(0) + y \frac{\partial u}{\partial y}(x, 0) + u_B(x))] \end{array} \right.$$

⇔

$$C^T \Lambda = g(x, y) \quad (2.2.16)$$

où le vecteur  $\Lambda$  et la fonction  $g(x, y)$  sont donnés par

$$\Lambda = (a_1 P_y^2 P_x + a_2 P_x^2 P_y + a_3 P_y^2 + a_4 P_x^2 + a_5 P_x P_y + a_6 P_x^2 P_y^2) \Psi(x, y) - (a_1 y P_y^2 P_x + a_3 y P_y^2) \Psi(x, 1) - (a_2 x P_x^2 P_y + a_4 x P_x^2 + a_5 P_x P_y P_x + a_6 x P_x^2 P_y^2) \Psi(1, y)$$

$$g(x, y) = f(x, y) - [a_1 (y(u'_H(x) - u'_H(0) - u'_B(x) + u'_B(0)) + u'_B(x) - u'_B(0) + \frac{\partial u}{\partial x}(0, y)) \\ + a_2 (x(u'_D(y) - u'_D(0) - u'_G(y) + u'_G(0)) + u'_G(y) - u'_G(0) + \frac{\partial u}{\partial y}(x, 0)) + a_3 (y(u''_H(x) \\ - u''_B(x)) + u''_B(x)) + a_4 (x(u''_D(y) - u''_G(y)) + u''_G(y)) + a_5 (u'_D(y) - u'_D(0) \\ - u'_G(y) + u'_G(0) + \frac{\partial^2 u}{\partial x \partial y}(x, 0)) + a_6 (x(u_D(y) - u_D(0) - y u'_D(0) - u_G(y) + u_G(0) \\ + y u'_G(0)) + u_G(y) - u_G(0) - y u'_G(0) + y \frac{\partial u}{\partial y}(x, 0) + u_B(x))]$$

**Remarque 2.2.2** L'équation scalaire (2.2.16) est transformée en un système de  $2^{k-1} 2^{k'-1} M M'$  d'équations en la considérant aux points de collocation  $(x_i, y_j)$   $1 \leq i \leq 2^{k-1} M$ ,  $1 \leq j \leq 2^{k'-1} M'$ .

La solution de ce système est le vecteur inconnu  $C$ , dont dépend la solution  $u(x, y)$  donnée par (2.2.9).

**Remarque 2.2.3** Si les conditions de Dirichlet sont données en  $x = 1$  et  $y = 1$ , on intègre (2.2.3) de  $x$  à 1 ou de  $y$  à 1. Ce qui donne

$$\int_x^1 \frac{\partial^4 u}{\partial x^2 \partial y^2}(\tau, y) d\tau = \int_0^1 C^T \Psi(\tau, y) d\tau - \int_0^x C^T \Psi(\tau, y) d\tau.$$

### 2.2.2 Cas des conditions aux limites de type mixte

$$\begin{cases} u(0, y) = h_2(y) & u(x, 0) = h_1(x) \\ \frac{\partial u}{\partial x}(0, y) = g_2(y) & \frac{\partial u}{\partial y}(x, 0) = g_1(x) \end{cases} \quad (2.2.17)$$

Les fonctions  $h_1$ ,  $h_2$ ,  $g_1$  et  $g_2$  sont supposées régulières sur  $[0, 1]$ .

On utilise la décomposition à base d'ondelettes de Legendre pour résoudre (2.2.1), où l'on pose :

$$\frac{\partial^4 u}{\partial x^2 \partial y^2}(x, y) = C^T \Psi(x, y). \quad (2.2.18)$$

En intégrant la formule (2.2.18) deux fois par rapport à  $x$  sur  $[0, x]$  on obtient :

$$\begin{aligned} \int_0^x \frac{\partial^4 u}{\partial x^2 \partial y^2}(\tau, y) d\tau &= \int_0^x C^T \Psi(\tau, y) d\tau \\ \frac{\partial^3 u}{\partial x \partial y^2}(x, y) &= C^T P_x \Psi(x, y) + \frac{\partial^3 u}{\partial x \partial y^2}(0, y) \\ &= C^T P_x \Psi(x, y) + \frac{\partial^2}{\partial y^2} \left( \frac{\partial u}{\partial x}(0, y) \right) \\ &= C^T P_x \Psi(x, y) + \frac{\partial^2}{\partial y^2} g_2(y) \quad (\text{d'après la 3}^{\text{ème}} \text{ condition (2.2.17)}) \end{aligned}$$

et

$$\frac{\partial^2 u}{\partial y^2}(x, y) = C^T P_x^2 \Psi(x, y) + x g_2''(y) + h_2''(y). \quad (2.2.19)$$

On intègre deux fois (2.2.19) par rapport à  $y$  sur  $[0, y]$  et on remplace par les conditions (2.2.17) on obtient :

$$\frac{\partial u}{\partial y}(x, y) = C^T P_x^2 P_y \Psi(x, y) + x[g_2'(y) - g_2'(0)] + h_2'(y) - h_2'(0) + g_1(x) \quad (2.2.20)$$

et

$$u(x, y) = C^T P_x^2 P_y^2 \Psi(x, y) + h_2(y) - h_2(0) - y h_2'(0) + y g_1(x) + h_1(x) + x[g_2(y) - g_2(0) - y g_2'(0)]. \quad (2.2.21)$$



Le terme de la dérivée croisée est le résultat d'une intégration de  $\frac{\partial^3 u}{\partial x \partial y^2}(x, y)$  par rapport à  $y$  sur  $[0, y]$

$$\frac{\partial^2 u}{\partial x \partial y}(x, y) = C^T P_x P_y \Psi(x, y) + g'_1(x) + g'_2(y) - g'_2(0). \quad (2.2.22)$$

En refaisant les mêmes étapes pour obtenir (2.2.19), mais en intégrant par rapport à  $y$  de  $[0, y]$ , on a

$$\frac{\partial^2 u}{\partial x^2}(x, y) = C^T P_y^2 \Psi(x, y) + y g''_1(x) + h''_1(x). \quad (2.2.23)$$

Une autre intégration par rapport à  $x$  sur  $[0, x]$  et les conditions (2.2.17) donnent

$$\frac{\partial u}{\partial x}(x, y) = C^T P_y^2 P_x \Psi(x, y) + y[g'_1(x) - g'_1(0)] + h'_1(x) - h'_1(0) + g_2(y) \quad (2.2.24)$$

Enfin, en portant (2.2.24), (2.2.23), (2.2.22), (2.2.21), (2.2.20), (2.2.19) dans (2.2.1), on a

$$\left\{ \begin{array}{l} a_1[C^T P_y^2 P_x \Psi(x, y) + y(g'_1(x) - g'_1(0)) + h'_1(x) - h'_1(0) + g_2(y)] \\ + a_2[C^T P_x^2 P_y \Psi(x, y) + x(g'_2(y) - g'_2(0)) + h'_2(y) - h'_2(0) + g_1(x)] \\ + a_3[C^T P_y^2 \Psi(x, y) + y g''_1(x) + h''_1(x)] + a_4[C^T P_x^2 \Psi(x, y) + x g''_2(y) + h''_2(y)] \\ + a_5[C^T P_x P_y \Psi(x, y) + g'_1(x) + g'_2(y) - g'_2(0)] + a_6[C^T P_x^2 P_y^2 \Psi(x, y) + h_2(y) - h_2(0) \\ - y h'_2(0) + y g_1(x) + h_1(x) + x(g_2(y) - g_2(0) - y g'_2(0))] = f(x, y) \end{array} \right.$$

$\Leftrightarrow$

$$\left\{ \begin{array}{l} C^T[(a_1 P_y^2 P_x + a_2 P_x^2 P_y + a_3 P_y^2 + a_4 P_x^2 + a_5 P_x P_y + a_6 P_x^2 P_y^2) \Psi(x, y)] = f(x, y) \\ - a_1 [y(g'_1(x) - g'_1(0)) + h'_1(x) - h'_1(0) + g_2(y)] - a_2 [x(g'_2(y) - g'_2(0)) \\ + h'_2(y) - h'_2(0) + g_1(x)] - a_3 [y g''_1(x) + h''_1(x)] - a_4 [x g''_2(y) + h''_2(y)] \\ - a_5 [g'_1(x) + g'_2(y) - g'_2(0)] - a_6 [h_2(y) - h_2(0) - y h'_2(0) + y g_1(x) \\ + h_1(x) + x(g_2(y) - g_2(0) - y g'_2(0))] \end{array} \right.$$

$\Leftrightarrow$

$$C^T \Lambda = g(x, y). \quad (2.2.25)$$

Où

$$\Lambda = [a_1 P_y^2 P_x + a_2 P_y P_x^2 + a_3 P_y^2 + a_4 P_x^2 + a_5 P_x P_y + a_6 P_x^2 P_y^2] \Psi(x, y)$$

et

$$\begin{aligned}
g(x, y) = & f(x, y) - a_1 [y(g'_1(x) - g'_1(0)) + h'_1(x) - h'_1(0) + g_2(y)] \\
& - a_2 [x(g'_2(y) - g'_2(0)) + h'_2(y) - h'_2(0) + g_1(x)] - a_3 [yg''_1(x) + h''_1(x)] \\
& - a_4 [xg''_2(y) + h''_2(y)] - a_5 [g'_1(x) + g'_2(y) - g'_2(0)] \\
& - a_6 [h_2(y) - h_2(0) - yh'_2(0) + yg_1(x) + h_1(x) + x(g_2(y) - g_2(0) - yg'_2(0))]
\end{aligned}$$

**Remarque 2.2.4** L'équation scalaire (2.2.25) est transformée en un système de  $2^{k-1}2^{k'-1}MM'$  d'équations en la considérant aux points de collocation  $(x_i, y_j)$   $1 \leq i \leq 2^{k-1}M$ ,  $1 \leq j \leq 2^{k'-1}M'$ .

La solution de ce système est le vecteur inconnu  $C$ , dont dépend la solution  $u(x, y)$  donnée par (2.2.21).

## 2.3 Résultats et tests numériques

### 2.3.1 Équation de Poisson avec conditions de Dirichlet

On considère l'équation de Poisson avec des conditions aux limites de type Dirichlet (Voir [6])

$$\frac{\partial^2 u}{\partial x^2}(x, y) + \frac{\partial^2 u}{\partial y^2}(x, y) + 2u(x, y) = 0$$

$$\begin{cases} u(0, y) = 0 & u(1, y) = \sin(1) \sin(y), \\ u(x, 0) = 0 & u(x, 1) = \sin(x) \sin(1). \end{cases}$$

L'équation de Poisson admet comme solution exacte :

$$u(x, y) = \sin(x) \sin(y).$$

On reprend la méthode de résolution de l'EDP (2.2.1) avec

$$a_1 = 0, a_2 = 0, a_3 = 1, a_4 = 1, a_5 = 0 \text{ et } a_6 = 2.$$

Les relations (2.2.7), (2.2.9) et (2.2.14) s'écrivent

$$\frac{\partial^2 u}{\partial x^2}(x, y) = C^T P_y^2(\Psi(x, y) - y\Psi(x, 1)) - y \sin(x) \sin(1),$$

$$\frac{\partial^2 u}{\partial y^2}(x, y) = C^T P_x^2(\Psi(x, y) - x\Psi(1, y)) - x \sin(1) \sin(y),$$

et

$$u(x, y) = C^T P_x^2 P_y^2 (\Psi(x, y) - x\Psi(1, y) - y\Psi(x, 1)) - xy C^T P_x^2 P_y^2 \Psi(1, 1) + x \sin(1) \sin(y) - xy \sin^2(1) + y \sin(x) \sin(1)$$

Le système (2.2.16) devient

$$\begin{aligned} & C^T [P_y^2 [\Psi(x, y) - y\Psi(x, 1)] + P_x^2 [\Psi(x, y) - x\Psi(1, y)] + \\ & 2P_x^2 P_y^2 [(\Psi(x, y) - x\Psi(1, y) - y\Psi(x, 1)) - xy\Psi(1, 1)]] \\ & = -x \sin(1) \sin(y) - y \sin(x) \sin(1) + 2yx \sin^2(1). \end{aligned} \quad (2.3.1)$$

### Résultats numériques

Pour différentes valeurs de  $M$ ,  $M'$  et de  $k$ ,  $k'$ , on résoud le système (2.3.1), pris aux points de collocation  $x_i = \frac{i-0.5}{M2^{k-1}}$ ,  $y_i = \frac{i-0.5}{M'2^{k'-1}}$  (Voir [2]). On obtient une solution approchée notée  $u_{app}$ . L'erreur de cette approximation est estimée par

$$Err = \|u_{ex} - u_{app}\|_2.$$

Où

$u_{ex}$  : représente la solution exacte.

$u_{app}$  : représente la solution obtenue par la méthode des ondelettes de Legendre

$\|\cdot\|_2$  : est la norme euclidienne.

les résultats obtenus sont résumés dans le tableau suivant.

<i>Erreur</i>	$M = M' = 2$	$M = M' = 3$	$M = M' = 5$	$M = M' = 6$	$M = M' = 8$
$k = k' = 1$	$3.3456e - 002$	$8.7652e - 003$	$1.0085e - 004$	$4.7398e - 006$	$1.4745e - 008$
$k = k' = 2$	$2.1711e - 002$	$1.8977e - 003$	$5.3338e - 006$	$1.9506e - 007$	$1.5504e - 010$
$k = k' = 3$	$1.1934e - 002$	$4.2563e - 004$	$2.9644e - 007$	$6.7233e - 009$	$3.1655e - 012$

Tableau1

On remarque que l'erreur diminue quand le nombre des niveaux et le nombre des points de collocation augmentent.

Pour plus de détails, on prend l'exemple de  $k = k' = 2$ . Les résultats de la solution exacte, approchée, l'erreur et du temps d'exécution sont exposés dans le Tableau 2

$M$	$M'$	$u_{ex}$	$u_{app}$	$Err$	$Temps\ CPU$
2	2	1.0811612013348	1.0594503451493	$2.1711e - 002$	0.587806 s
3	3	1.6297187703192	1.6278219237741	$1.8977e - 003$	0.640184 s
5	5	2.7229632687960	2.7229686003165	$5.3338e - 006$	1.495646 s
6	6	3.2689478769328	3.2689476819166	$1.9506e - 007$	3.883429 s
8	8	4.3604412510542	4.3604412512092	$1.5504e - 010$	136.197356 s

Tableau 2

Pour ce cas, la concordance entre la solution exacte et la solution approchée est clairement mise en évidence dans la représentation graphique suivante

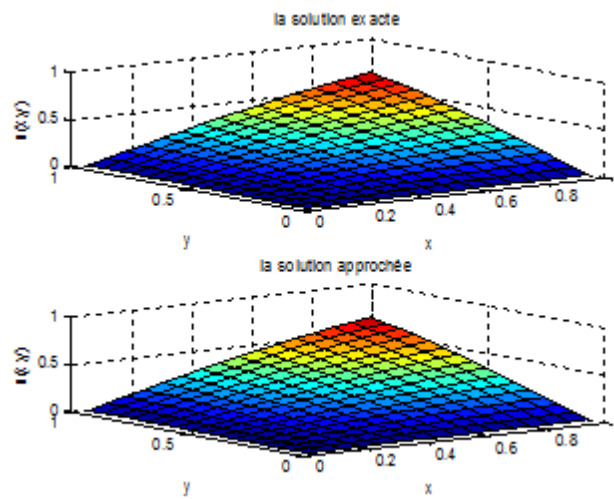


Figure 1

Toujours pour ce même cas, la décroissance de l'erreur vers zéro est visible dans le graphe

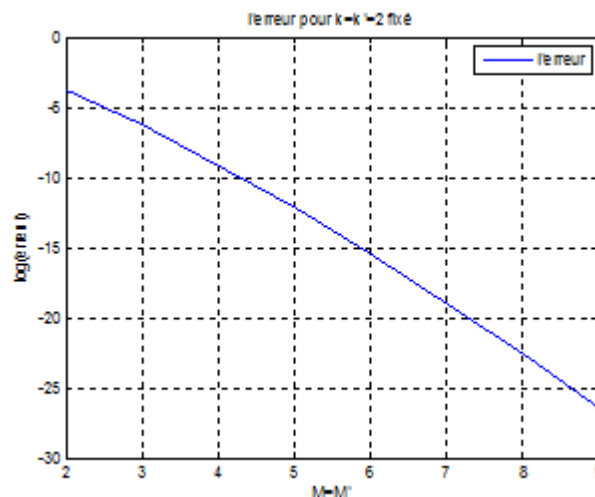


Figure 2 : L'erreur absolue

### 2.3.2 Équation de Poisson avec conditions mixtes

On reprend l'équation de Poisson avec des conditions aux limites de type mixte

$$\frac{\partial^2 u}{\partial x^2}(x, y) + \frac{\partial^2 u}{\partial y^2}(x, y) + 2u(x, y) = 0 \quad (2.3.2)$$

$$\begin{cases} u(0, y) = 0 & \frac{\partial u}{\partial x}(0, y) = \sin(y) \\ u(x, 0) = 0 & \frac{\partial u}{\partial y}(x, 0) = \sin(x) \end{cases} \quad (2.3.3)$$

dont la solution exacte est

$$u(x, y) = \sin(x) \sin(y).$$

On reprend la méthode de résolution de l'EDP (2.2.1) avec

$a_1 = 0$ ,  $a_2 = 0$ ,  $a_3 = 1$ ,  $a_4 = 1$ ,  $a_5 = 0$  et  $a_6 = 2$ .

Les relations (2.2.23), (2.2.19) et (2.2.21) s'écrivent

$$\frac{\partial^2 u}{\partial x^2}(x, y) = C^T P_y^2 \Psi(x, y) - y \sin(x),$$

$$\frac{\partial^2 u}{\partial y^2}(x, y) = C^T P_x^2 \Psi(x, y) - x \sin(y),$$

et

$$u(x, y) = C^T P_x^2 P_y^2 \Psi(x, y) - xy + x \sin(y) + y \sin(x).$$

Le système (2.2.25) devient

$$\begin{aligned} C^T [P_y^2 + P_x^2 + 2P_x^2 P_y^2] \Psi(x, y) &= 2xy + x \sin(y) + y \sin(x) - 2x \sin(y) - 2y \sin(x) \\ &= 2xy - x \sin(y) - y \sin(x). \end{aligned}$$

Les tests numériques pour ce cas (condition mixte) présentent des anomalies telles qu'on peut le remarquer dans les tableaux et les figures qui suivent.

<i>Erreur</i>	$M = M' = 2$	$M = M' = 3$	$M = M' = 4$	$M = M' = 7$	$M = M' = 10$
$k = k' = 1$	$1.1624e - 003$	$1.6192e - 003$	$6.4599e - 005$	$2.7542e - 007$	$1.3569e - 008$
$k = k' = 2$	$6.0021e - 004$	$4.0452e - 001$	$5.1789e - 004$	$6.2592e - 005$	$3.1469e - 005$

Tableau 3

Tableau des valeurs de la solution exacte, approchée, l'erreur et le temps d'exécution  
 $k = k' = 2$

$M$	$M'$	$u_{ex}$	$u_{app}$	$Err$	$Temps\ CPU$
2	2	1.0811612013348	1.0808867151279	$6.0021e - 004$	0.540786s
3	3	1.6297187703192	1.8212024477102	$4.0452e - 001$	0.561303s
4	4	2.1766605748093	2.1764717443751	$5.1789e - 004$	0.625259s
7	7	3.8147511520427	3.8147359859595	$6.2592e - 005$	3.733572s
10	10	5.4516179435622	5.4516110335336	$3.1469e - 005$	51.218380s

Tableau 4

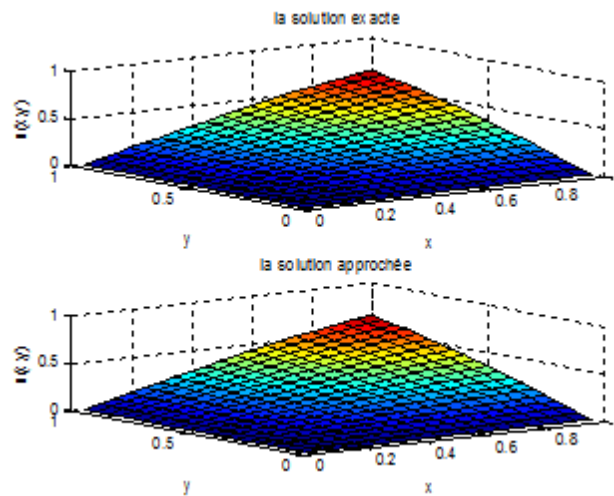


Figure 3

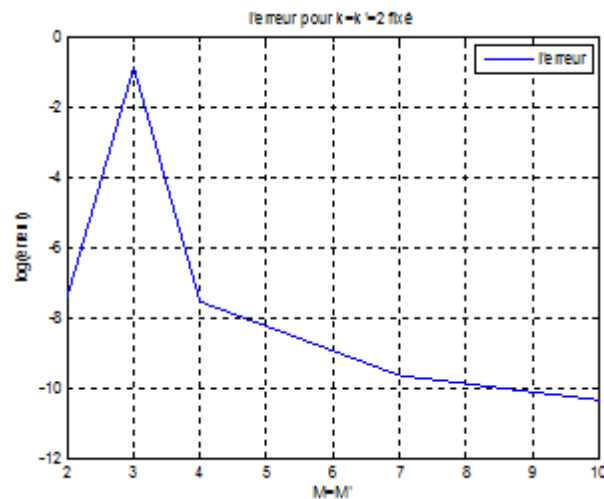


Figure 4 : L'erreur absolue

# RÉSOLUTION D'UNE EDP EN 2D PAR LA MÉTHODE DES ONDOULETTES DE LEGENDRE AVEC LE TEMPS

---

On propose de résoudre l'équation complète d'ordre deux à coefficients constants en présence du temps

$$\begin{aligned} \frac{\partial u}{\partial t}(t, x, y) + a_1 \frac{\partial u}{\partial x}(t, x, y) + a_2 \frac{\partial u}{\partial y}(t, x, y) + a_3 \frac{\partial^2 u}{\partial x^2}(t, x, y) + a_4 \frac{\partial^2 u}{\partial y^2}(t, x, y) + a_5 \frac{\partial^2 u}{\partial x \partial y}(t, x, y) + a_6 u(t, x, y) \\ = f(t, x, y) \end{aligned} \tag{3.0.1}$$

avec des conditions aux limites de type mixte

$$\begin{cases} u(t, x = 0, y) = h_2(t, y) & u(t, x, y = 0) = h_1(t, x), \\ \frac{\partial u}{\partial x}(t, x = 0, y) = g_2(t, y) & \frac{\partial u}{\partial y}(t, x, y = 0) = g_1(t, x), \end{cases}$$

et la condition initiale

$$u(t = 0, x, y) = u_0(x, y).$$

On choisit un pas de temps  $\Delta t$  et on considère les points de la discrétisation temporelle  $t_i = i\Delta t$ .

On applique un schéma aux différences finies progressif pour approcher la dérivée temporelle  $\frac{\partial u}{\partial t}(t, x, y)$ . On a alors

$$\frac{\partial u}{\partial t}(t_i, x, y) \simeq \frac{u(t_{i+1}, x, y) - u(t_i, x, y)}{\Delta t}. \quad (3.0.2)$$

L'équation (3.0.1) devient :

$$\begin{aligned} \frac{u(t_{i+1}, x, y) - u(t_i, x, y)}{\Delta t} = & f(t_i, x, y) - (a_1 \frac{\partial u}{\partial x}(t_i, x, y) + a_2 \frac{\partial u}{\partial y}(t_i, x, y) + a_3 \frac{\partial^2 u}{\partial x^2}(t_i, x, y) \\ & + a_4 \frac{\partial^2 u}{\partial y^2}(t_i, x, y) + a_5 \frac{\partial^2 u}{\partial x \partial y}(t_i, x, y) + a_6 u(t_i, x, y)). \end{aligned} \quad (3.0.3)$$

En tout point  $t_i$ , la fonction  $u(t_i, x, y)$  étant dans l'espace  $L^2([0, 1] \times [0, 1])$ , elle peut donc admettre la décomposition dans la base d'ondelettes de Legendre (Voir [1])

$$u(t_i, x, y) = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} \sum_{n'=1}^{\infty} \sum_{m'=0}^{\infty} C_{n,m,n',m'}(t_i) \Psi_{n,m,n',m'}(x, y),$$

où les coefficients  $C_{n,m,n',m'}(t_i)$  qui dépendent des points  $t_i$ ;  $i = 0, 1, 2, \dots$ , sont définis par :

$$C_{n,m,n',m'}(t_i) = \langle u(t_i, \cdot, \cdot), \Psi_{n,m,n',m'}(\cdot, \cdot) \rangle_{L^2([0,1] \times [0,1])} = \int_0^1 \int_0^1 u(t_i, x, y) \Psi_{n,m,n',m'}(x, y) dx dy.$$

pour tout  $t_i$ ;  $i = 0, 1, 2, \dots$ , la troncature de la série permet d'avoir l'approximation suivante :

$$u(t_i, x, y) \simeq \sum_{n=1}^{2^{k-1}M-1} \sum_{m=0}^{12^{k'-1}M'-1} \sum_{n'=1}^{\infty} \sum_{m'=0}^{\infty} C_{n,m,n',m'}(t_i) \Psi_{n,m,n',m'}(x, y) = C^T(t_i) \Psi(x, y)$$

L'équivalent des formules de (2.2.18) – (2.2.24) sont :

$$\frac{\partial^2 u}{\partial y^2}(t_i, x, y) = C^T(t_i) P_x^2 \Psi(x, y) + x \frac{\partial^2}{\partial y^2} g_2(t_i, y) + \frac{\partial^2}{\partial y^2} h_2(t_i, y) \quad (3.0.4)$$

$$\begin{aligned} \frac{\partial u}{\partial y}(t_i, x, y) = & C^T(t_i) P_x^2 P_y \Psi(x, y) + x \left[ \frac{\partial}{\partial y} g_2(t_i, y) - \frac{\partial}{\partial y} g_2(t_i, 0) \right] \\ & + \frac{\partial}{\partial y} h_2(t_i, y) - \frac{\partial}{\partial y} h_2(t_i, 0) + g_1(t_i, x) \end{aligned} \quad (3.0.5)$$

$$\frac{\partial^2 u}{\partial x^2}(t_i, x, y) = C^T(t_i) P_y^2 \Psi(x, y) + y \frac{\partial^2}{\partial y^2} g_1(t_i, x) + \frac{\partial^2}{\partial x^2} h_1(t_i, x) \quad (3.0.6)$$



$$\begin{aligned} \frac{\partial u}{\partial x}(t_i, x, y) &= C^T(t_i)P_y^2P_x\Psi(x, y) + y\left[\frac{\partial}{\partial x}g_1(t_i, x) - \frac{\partial}{\partial x}g_1(t_i, 0)\right] \\ &\quad + \frac{\partial}{\partial x}h_1(t_i, x) - \frac{\partial}{\partial y}h_1(t_i, 0) + g_2(t_i, y) \end{aligned} \quad (3.0.7)$$

$$\frac{\partial^2 u}{\partial x \partial y}(t_i, x, y) = C^T(t_i)P_xP_y\Psi(x, y) + \frac{\partial}{\partial x}g_1(t_i, x) + \frac{\partial}{\partial y}g_2(t_i, y) - \frac{\partial}{\partial y}g_2(t_i, 0) \quad (3.0.8)$$

$$\begin{aligned} u(t_i, x, y) &= C^T(t_i)P_x^2P_y^2\Psi(x, y) + h_2(t_i, y) - h_2(t_i, 0) - y\frac{\partial}{\partial y}h_2(t_i, 0) \\ &\quad + yg_1(t_i, x) + h_1(t_i, x) + x[g_2(t_i, y) - g_2(t_i, 0) - y\frac{\partial}{\partial y}g_2(t_i, 0)] \end{aligned} \quad (3.0.9)$$

$$\begin{aligned} u(t_{i+1}, x, y) &= C^T(t_{i+1})P_x^2P_y^2\Psi(x, y) + h_2(t_{i+1}, y) - h_2(t_{i+1}, 0) - y\frac{\partial}{\partial y}h_2(t_{i+1}, 0) \\ &\quad + yg_1(t_{i+1}, x) + h_1(t_{i+1}, x) + x[g_2(t_{i+1}, y) - g_2(t_{i+1}, 0) - y\frac{\partial}{\partial y}g_2(t_{i+1}, 0)]. \end{aligned} \quad (3.0.10)$$

On remplace (3.0.4)-(3.0.10) dans (3.0.3) on obtient

$$\begin{aligned} u(t_{i+1}, x, y) - u(t_i, x, y) &= \Delta t f(t_i, x, y) - \Delta t a_1 [C^T(t_i)P_y^2P_x\Psi(x, y) + y\left(\frac{\partial}{\partial x}g_1(t_i, x) - \frac{\partial}{\partial x}g_1(t_i, 0)\right) \\ &\quad + \frac{\partial}{\partial x}h_2(t_i, x) - \frac{\partial}{\partial y}h_1(t_i, 0) + g_2(t_i, y)] - \Delta t a_2 [C^T(t_i)P_x^2P_y\Psi(x, y) \\ &\quad + x\left(\frac{\partial}{\partial y}g_2(t_i, y) - \frac{\partial}{\partial y}g_2(t_i, 0)\right) + \frac{\partial}{\partial y}h_2(t_i, y) - \frac{\partial}{\partial y}h_2(t_i, 0) + g_1(t_i, x)] \\ &\quad - \Delta t a_3 [C^T(t_i)P_y^2\Psi(x, y) + y\frac{\partial^2}{\partial y^2}g_1(t_i, x) + \frac{\partial^2}{\partial x^2}h_1(t_i, x)] \\ &\quad - \Delta t a_4 [C^T(t_i)P_x^2\Psi(x, y) + x\frac{\partial^2}{\partial y^2}g_2(t_i, y) + \frac{\partial^2}{\partial y^2}h_2(t_i, y)] \\ &\quad - \Delta t a_5 [C^T(t_i)P_xP_y\Psi(x, y) + \frac{\partial}{\partial y}g_1(t_i, y) + \frac{\partial}{\partial y}g_2(t_i, y) - \frac{\partial}{\partial y}g_2(t_i, 0)] \\ &\quad - \Delta t a_6 [C^T(t_i)P_x^2P_y^2\Psi(x, y) + h_2(t_i, y) - h_2(t_i, 0) - y\frac{\partial}{\partial y}h_2(t_i, 0) \\ &\quad + yg_1(t_i, x) + h_1(t_i, x) + x(g_2(t_i, y) - g_2(t_i, 0) - y\frac{\partial}{\partial y}g_2(t_i, 0))]. \end{aligned}$$

On remplace  $u(t_{i+1}, x, y)$  et  $u(t_i, x, y)$  par leur valeur

$$\begin{aligned}
C^T(t_{i+1})P_x^2P_y^2\Psi(x, y) &= h_2(t_{i+1}, 0) - h_2(t_{i+1}, y) - yg_1(t_{i+1}, x) + y\frac{\partial}{\partial y}h_2(t_{i+1}, 0) - h_1(t_{i+1}, x) \\
&\quad - x[g_2(t_{i+1}, y) - g_2(t_{i+1}, 0) - y\frac{\partial}{\partial y}g_2(t_{i+1}, 0)] + \Delta tf(t_i, x, y) \\
&\quad - \Delta ta_1[C^T(t_i)P_y^2P_x\Psi(x, y) + y(\frac{\partial}{\partial x}g_1(t_i, x) - \frac{\partial}{\partial x}g_1(t_i, 0)) \\
&\quad + \frac{\partial}{\partial x}h_2(t_i, x) - \frac{\partial}{\partial y}h_1(t_i, 0) + g_2(t_i, y)] - \Delta ta_2[C^T(t_i)P_x^2P_y\Psi(x, y) \\
&\quad + x(\frac{\partial}{\partial y}g_2(t_i, y) - \frac{\partial}{\partial y}g_2(t_i, 0)) + \frac{\partial}{\partial y}h_2(t_i, y) - \frac{\partial}{\partial y}h_2(t_i, 0) \\
&\quad + g_1(t_i, x)] - \Delta ta_3[C^T(t_i)P_y^2\Psi(x, y) + y\frac{\partial^2}{\partial y^2}g_1(t_i, x) \\
&\quad + \frac{\partial^2}{\partial x^2}h_1(t_i, x)] - \Delta ta_4[C^T(t_i)P_x^2\Psi(x, y) + x\frac{\partial^2}{\partial y^2}g_2(t_i, y) + \frac{\partial^2}{\partial y^2}h_2(t_i, y)] \\
&\quad - \Delta ta_5[C^T(t_i)P_xP_y\Psi(x, y) + \frac{\partial}{\partial y}g_1(t_i, y) + \frac{\partial}{\partial y}g_2(t_i, y) - \frac{\partial}{\partial y}g_2(t_i, 0)] \\
&\quad - \Delta ta_6[C^T(t_i)P_x^2P_y^2\Psi(x, y) + h_2(t_i, y) - h_2(t_i, 0) - y\frac{\partial}{\partial y}h_2(t_i, 0) + \\
&\quad yg_1(t_i, x) + h_1(t_i, x) + x(g_2(t_i, y) - g_2(t_i, 0) - y\frac{\partial}{\partial y}g_2(t_i, 0))] \\
&\quad + C^T(t_i)P_x^2P_y^2\Psi(x, y) + h_2(t_i, y) - h_2(t_i, 0) - y\frac{\partial}{\partial y}h_2(t_i, 0) + yg_1(t_i, x) \\
&\quad + h_1(t_i, x) + x[g_2(t_i, y) - g_2(t_i, 0) - y\frac{\partial}{\partial y}g_2(t_i, 0)].
\end{aligned}$$

On obtient le système suivant

$$C^T(t_{i+1})P_x^2P_y^2\Psi(x, y) = C^T(t_i)M\Psi(x, y) + F(t_i, t_{i+1}, x, y), \quad (3.0.11)$$

avec

$$M = -\Delta t [a_1P_y^2P_x + a_2P_x^2P_y + a_3P_y^2 + a_4P_x^2 + a_5P_xP_y + a_6P_x^2P_y^2] + P_x^2P_y^2,$$

et

$$\begin{aligned}
 F(t_i, t_{i+1}, x, y) = & h_2(t_{i+1}, 0) - h_2(t_{i+1}, y) - yg_1(t_{i+1}, x) + y\frac{\partial}{\partial y}h_2(t_{i+1}, 0) - h_1(t_{i+1}, x) \\
 & -x[g_2(t_{i+1}, y) - g_2(t_{i+1}, 0) - y\frac{\partial}{\partial y}g_2(t_{i+1}, 0)] + \Delta tf(t_i, x, y) \\
 & -\Delta ta_1[y(\frac{\partial}{\partial x}g_1(t_i, x) - \frac{\partial}{\partial x}g_1(t_i, 0)) + \frac{\partial}{\partial x}h_2(t_i, x) - \frac{\partial}{\partial y}h_1(t_i, 0) + g_2(t_i, y)] \\
 & -\Delta ta_2[x(\frac{\partial}{\partial y}g_2(t_i, y) - \frac{\partial}{\partial y}g_2(t_i, 0)) + \frac{\partial}{\partial y}h_2(t_i, y) - \frac{\partial}{\partial y}h_2(t_i, 0) + g_1(t_i, x)] \\
 & -\Delta ta_3[y\frac{\partial^2}{\partial y^2}g_1(t_i, x) + \frac{\partial^2}{\partial x^2}h_1(t_i, x)] - \Delta ta_4[x\frac{\partial^2}{\partial y^2}g_2(t_i, y) + \frac{\partial^2}{\partial y^2}h_2(t_i, y)] \\
 & -\Delta ta_5[\frac{\partial}{\partial y}g_1(t_i, y) + \frac{\partial}{\partial y}g_2(t_i, y) - \frac{\partial}{\partial y}g_2(t_i, 0)] \\
 & -\Delta ta_6[h_2(t_i, y) - h_2(t_i, 0) - y\frac{\partial}{\partial y}h_2(t_i, 0) + yg_1(t_i, x) + h_1(t_i, x) + x(g_2(t_i, y) \\
 & -g_2(t_i, 0) - y\frac{\partial}{\partial y}g_2(t_i, 0))] + h_2(t_i, y) - h_2(t_i, 0) - y\frac{\partial}{\partial y}h_2(t_i, 0) + yg_1(t_i, x) \\
 & +h_1(t_i, x) + x[g_2(t_i, y) - g_2(t_i, 0) - y\frac{\partial}{\partial y}g_2(t_i, 0)].
 \end{aligned}$$

Le vecteur au point initial,  $C(0)$  est trouvé de la manière suivante

$$\begin{aligned}
 u(t_0, x, y) &= C^T(t_0)P_x^2P_y^2\Psi(x, y) + H(t_0, x, y) \\
 u_0(x, y) &= C^T(0)P_x^2P_y^2\Psi(x, y) + H(0, x, y)
 \end{aligned}$$

En considérant la décomposition des fonctions  $u_0(x, y)$  et  $H(0, x, y)$  suivant la base d'onde-  
lettes :

$$\begin{aligned}
 u_0(x, y) &= U_0^T\Psi(x, y) \\
 H(0, x, y) &= H^T\Psi(x, y)
 \end{aligned}$$

et en utilisant (3.0.9) on a

$$U_0^T\Psi(x, y) = C^T(0)P_x^2P_y^2\Psi(x, y) + H^T\Psi(x, y)$$

ce qui permet d'avoir le vecteur  $C(0)$

$$C(0) = ((P_x^2 P_y^2)^T)^{-1}(U_0 - H)$$

où

$$\begin{aligned} H(0, x, y) = & h_2(0, y) - h_2(0, 0) - y \frac{\partial}{\partial y} h_2(0, 0) + y g_1(0, x) \\ & + x [g_2(0, y) - g_2(0, 0) - y \frac{\partial}{\partial y} g_2(0, 0)]. \end{aligned}$$

### Exemple pour le temps

L'équation à considérée est

$$\frac{\partial u}{\partial t}(t, x, y) - \Delta u(t, x, y) = -\exp(-t_i)(x^2 y^2 + x^2 + y^2),$$

dont la solution exacte est :

$$u(t, x, y) = \exp(-t)x^2 y^2.$$

On considère la condition initiale :

$$u(t = 0, x, y) = u_0(x, y) = x^2 y^2$$

et les conditions mixtes

$$\begin{cases} u(t, x = 0, y) = u(t, x, y = 0) = 0, \\ \frac{\partial u}{\partial x}(t, x = 0, y) = 0 \quad \frac{\partial u}{\partial y}(t, x, y = 0) = 0. \end{cases}$$

On reprend la méthode de résolution de l'EDP (3.0.1) avec

$$a_1 = a_2 = 0, a_3 = -1, a_4 = -1, a_5 = a_6 = 0.$$

Les relations (3.0.4), (3.0.5), (3.0.6), (3.0.7) et (3.0.9), s'expriment

$$\frac{\partial^2 u}{\partial y^2}(t_i, x, y) = C^T(t_i) P_x^2 \Psi(x, y),$$

$$u(t_i, x, y) = C^T(t_i) P_x^2 P_y^2 \Psi(x, y),$$

$$\frac{\partial^2 u}{\partial x^2}(t_i, x, y) = C^T(t_i) P_y^2 \Psi(x, y),$$

et le système (3.0.11) devient

$$C^T(t_{i+1}) P_x^2 P_y^2 \Psi(x, y) = C^T(t_i) [P_x^2 P_y^2 + \Delta t P_x^2 + \Delta t P_y^2 +] \Psi(x, y) - \Delta t \exp(-t_i) (x^2 y^2 + x^2 + y^2).$$

Le vecteur initial  $C(0)$  est donné par

$$C(0) = [(P_x^2 P_y^2)^T]^{-1} U_{x^2 y^2}.$$

où  $U_{x^2 y^2}$  est le vecteur obtenu de la décomposition de la fonction  $u(x, y) = x^2 y^2$  en base d'ondelettes de Legendre  $x^2 y^2 = U_{x^2 y^2}^T \Psi(x, y)$ .

Il reste à résoudre numériquement ce système.

---

# CONCLUSION

---

Dans ce travail, on a manipulé une nouvelle "pour nous" méthode basée sur la décomposition en ondelettes de Legendre en deux dimensions.

On a donné une procédure pour estimer la solution d'une EDP avec des conditions aux limites de type Dirichlet et de type mixte.

L'avantage de la méthode proposée réside dans son efficacité pour un nombre réduit de points de collocation. Cependant, les résultats numériques obtenus, pour les conditions aux limites de Dirichlet, sont plus cohérents que ceux donnés dans le cas mixte.

La même approche est appliquée pour résoudre les EDPs en présence du temps.

Comme perspectives on propose :

d'adapter cette méthode de résolution pour les équations non linéaires et à coefficients non constants,

développer un programme pour traiter numériquement les équations en présence du temps.

# Bibliographie

- [1] **E.Banifatemi,M.Razzaghi** and **S.Yousefi**. Towo-dimensional Legendre Wavelets Methods for the Mixed Volterra-Fredholm integral Equations. journal of vibration and control 2007 13 :1667-1674.
- [2] **A.Imran ,Siraj-ul-Islam,B.Sarler**,wavelets collocation methods for the numerical solution of elleptic BV problems
- [3] **M.Razzaghi, S.Yousefi** : Legendre wavelets direct method for variational problems. Mathematics and Computers in Simulation53, 2000, 185-192.
- [4] **M.Razzaghi, S.Yousefi** The Legendre wavelets operational matrix of integration. International Journal of Systems Science 2001 ; 32 :495–502.
- [5] **H. Parsian**, two dimension Legendre wavelets and operational matrices of integration, Acta Mathematica Academiae Paedagogicae 21 (2005) 101-106.
- [6] **F.yin,J.song**,solving linear PDEs with the aid of two dimentional legendre wavelets

# Annexe

---

Cette partie concerne l'application Matlab

## 4.1 Les sous programmes

### 4.1.1 Polynôme de Legendre $px$ et $py$

```
function [px] = legendre_polyx(x,Mx)
px = zeros(Mx,1);
px(1) = 1;
px(2) = x;
    for i = 3 :Mx
        px(i) = 1/(i-1)*((2*i-3)*x*px(i-1)-(i-2)*px(i-2));
    end
end
function [py ]= legendre_polyy(y,My)
py = zeros(My,1);
py(1) = 1;
py(2) = y;
    for i = 3 :My
        py(i) = 1/(i-1)*((2*i-3)*y*py(i-1)-(i-2)*py(i-2));
    end
end
```



### 4.1.2 L'ondelette de Legendre

```
function [ psi ] = LegWav(x,y,kx,ky,nx,ny,mx,my,pxm,pym)
if (x>=(nx-1)/2^(kx - 1))&&(x<=(nx)/2^(kx - 1))&& (y >= (ny-1)/2^...
    (ky - 1))&&(y <= ny/2^(ky - 1))
    psi = sqrt((mx - .5)*(my - .5))*2^((kx + ky)/2)*pxm*pym;
else
    psi = 0;
end
end
```

Le vecteur d'ondelette de Legendre

```
function [ psiv ] = VectLegWav(x,y,Mx,My,kx,ky)
i = 0;
psiv = zeros(1,2^(kx - 1)*2^(ky - 1)*Mx*My);
for nx = 1 :2^(kx-1)
    for mx = 1 :Mx
        for ny = 1 :2^(ky-1)
            px = legendre_polyx(2^(kx)*x-2*nx+1,Mx);
            py = legendre_polyy(2^(ky)*y-2*ny+1,My);
            for my = 1 :My
                i = i+1;
                psiv(i) = LegWav(x,y,kx,ky,nx,ny,mx,my,px(mx),py(my));
            end
        end
    end
end
end
end
```

### 4.1.3 La matrice d'ondelette de Legendre

```
function [ psim ] = MatLegWav(x,y,Mx,My,kx,ky)
```

```

psim = zeros(2^(kx -1)*2^(ky -1)*Mx*My,2^(kx -1)*2^(ky -1)*Mx*My);
i = 0;
for ix= 1 :length(x)
    for iy =1 :length(y)
        i = i+1;
        psim( :,i) = VectLegWav(x(ix),y(iy),Mx,My,kx,ky);
    end
end
end
end

```

#### 4.1.4 Les matrices d'intégrations $P_x, P_y$

```

function [Px] = matrice_integracion(Mx,My,kx,ky)
%D la matrice identité
D=eye(2^(ky-1)*My);
L = tri_diag_matrixx(Mx,My,kx,ky);
F = zeros(Mx);
F(1,1)=2;
Px =(1/2^kx)*repeat_matrixx(F,L,kx);
Px=kron(Px,D);
end
function [Py] = matrice_integraciony(My,ky,Mx,kx)
L = tri_diag_matrixy(My,Mx);
if Mx==My
    F = zeros(My);
    if ky > 1
        F(1,1)= 2;
    else if ky == 1
        F(1,1) = 0;
    end
end
end
end

```

```

    May = eye(2^(kx-1)*My);
    Py =(1/2^ky )* kron(May,repeat_matrixxy(F,L,kx,ky,Mx,My));
else
    F = zeros(Mx,My);
    if ky > 1
        F(1,1)= 2;
    else if ky == 1
        F(1,1) = 0;
        end
    end
    if(Mx < My)
        May = eye(2^(ky-1)*My,2^(kx-1)*Mx);
    elseif (Mx > My)
        May = eye(2^(ky-1)*My,2^(kx-1)*Mx);
        end
        Py =(1/2^ky )*kron(May,repeat_matrixxy(F,L,kx,ky,Mx,My));
    end
end

```

La matrice  $L_x, L_y$

```

function [L] = tri_diag_matrixx(Mx,My,kx,ky)
if kx >= ky
    if Mx==My
        v = zeros(Mx*My-2,1);
        for i = 1 :Mx*My -2
            v(i) = 1/sqrt(4*(i)^2-1);
        end
        L= diag(v,1) + diag([1,zeros(1,Mx*My-2)]) - diag(v,-1);
        L=L(1 :Mx,1 :Mx);
    elseif Mx <My

```

```

v = zeros(Mx*My-2,1);
for i = 1 :Mx*My -2
    v(i) = 1/sqrt(4*(i)^2-1);
end
L= diag(v,1) + diag([1,zeros(1,Mx*My-2)]) - diag(v,-1);
L=L(1 :Mx,1 :Mx);
else
    v = zeros(Mx*My-2,1);
    for i = 1 :Mx*My-2
        v(i) = 1/sqrt(4*(i)^2-1);
    end
L= diag(v,1) + diag([1,zeros(1,Mx*My-2)]) - diag(v,-1);
L=L(1 :My+1,1 :My+1);
end
else if kx < ky
    if Mx==My
        v = zeros(1,Mx*My - 2);
        for i = 1 :Mx*My - 2
            v(i) = 1/sqrt(4*(i)^2-1);
        end
L = diag(v,1) + diag([1,zeros(1,Mx*My - 2)]) - diag(v,-1);
L=L(1 :Mx,1 :Mx);
elseif Mx < My
    v = zeros(Mx*My-2,1);
    for i = 1 :Mx*My -2
        v(i) = 1/sqrt(4*(i)^2-1);
    end
L= diag(v,1) + diag([1,zeros(1,Mx*My-2)]) - diag(v,-1);
L=L(1 :Mx,1 :Mx);
else

```

```

    v = zeros(Mx*My-2,1);
        for i = 1 :Mx*My-2
            v(i) = 1/sqrt(4*(i)^2-1);
        end
    L= diag(v,1) + diag([1,zeros(1,Mx*My-2)]) - diag(v,-1);
    L=L(1 :My+1,1 :My+1);
    end
else
    if Mx==My
        v = zeros(1,Mx*My - 2);
        for i = 1 :Mx*My - 2
            v(i) = 1/sqrt(4*(i)^2-1);
        end
    L = diag(v,1) + diag([1,zeros(1,Mx*My - 2)]) - diag(v,-1);
    L=L(1 :Mx-1,1 :Mx-1);
    elseif Mx < My
        v = zeros(Mx*My-2,1);
            for i = 1 :Mx*My -2
                v(i) = 1/sqrt(4*(i)^2-1);
            end
        L= diag(v,1) + diag([1,zeros(1,Mx*My-2)]) - diag(v,-1);
        L=L(1 :My,1 :My);
    else
        v = zeros(Mx*My-2,1);
            for i = 1 :Mx*My-2
                v(i) = 1/sqrt(4*(i)^2-1);
            end
        L= diag(v,1) + diag([1,zeros(1,Mx*My-2)]) - diag(v,-1);
        L=L(1 :My-1,1 :My-1);
    end
end

```

```

    end
end
function [L] = tri_diag_matrixy(My,Mx)
    if Mx == My
        v = zeros(My-1,1);
        for i = 1 :My-1
            v(i) = 1/sqrt(4*(i)^2-1);
        end
        L= diag(v,1) + diag([1,zeros(1,My-1)]) - diag(v,-1);
    else if Mx <My
        v = zeros(1,My-1);
        for i = 1 :My-1
            v(i) = 1/sqrt(4*(i)^2-1);
        end
        L = diag(v,1) + diag([1,zeros(1,My-1)]) - diag(v,-1);
        L=L(1 :Mx, :);
    else
        v = zeros(Mx-1,1);
        for i = 1 :My
            v(i) = 1/sqrt(4*(i)^2-1);
        end
        L = diag(v,1) + diag([1,zeros(1,Mx-1)]) - diag(v,-1);
        L=L( :,1 :My);
    end
end
end
end

```

```

les sous programmes repeat Mtrix x, y function [Max] = repeat_matrixx(F,L,kx)
nc = size(F);
Max = zeros(2^(kx - 1)*nc);

```

```

    for i = 1 :2^(kx - 1)
        Max((i - 1)*nc + 1 :i*nc,(i - 1)*nc + 1 :i*nc) = L;
        for m = i+1 :2^(kx - 1)
            Max((i - 1)*nc + 1 :i*nc,(m - 1)*nc + 1 :m*nc) = F;
        end
    end
end

function [May] = repeat_matrixy(F,L,kx,ky,Mx,My)
if Mx==My
    nc = size(F);
    May = zeros(2^(ky - 1)*nc);
    for i = 1 :2^(ky - 1)
        May((i - 1)* nc + 1 :i * nc,(i - 1)*nc + 1 :i*nc ) = L;
        for m = i +1 :2^(ky - 1)
            May((i - 1)*nc + 1 :i*nc,(m - 1)*nc + 1 :m*nc) = F;
        end
    end
end
else
    a = triu(ones(2^(kx - 1),2^(ky - 1)),1);
    f=kron(a,F);
    b=eye(2^(kx - 1),2^(ky - 1));
    l=kron(b,L);
    May=f+l;
end
end
end

```

## 4.2 Le programme principal pour les conditions Dirichlet

tic

clc

```

clear all
close all
format short e
%% TestLegWav.m
%Dans ce programme on calcule le polynôme de Legendre et les ondelettes
% de Legendre et on détermine les matrices opérationnelles d'intégration
% et en trouve la solution de système C'A=g
% (Mx =M, My=M') le degré du polynôme de Legendre(x,y)
%(kx=k ,ky=k') les niveaux
%%exemple de l'équation de Poisson %%%
%%pour les conditions de Dirichlet %%%
%% Set Prarameters
Mx = 12;
My = 12;
kx = 2;
ky = 2;
%% Construction of vect x & y
    for i=1 :Mx*2^(kx-1)
        x(i)=(i-0.5)/(Mx*2^(kx-1));
    end
x;
    for ii=1 :My*2^(ky-1)
        y(ii)=(ii-0.5)/(My*2^(ky-1));
    end
y;
%% Psi
[ psim ] = MatLegWav(x,y,Mx,My,kx,ky);
psim;
%% Px
[Px] = matrice_integration(Mx,My,kx,ky);

```



```
%% Py
[Py] = matrice_integriony(My,ky,Mx,kx);
Px;
Py;
%% fctg (la fonction g(x,y) en tout point de collocation)
[ fctg ] = Fonctiong(x,y,Mx,My,kx,ky);
%% vectdeny(calcul de l'ondelette de legendre aux points de collocation
% et aux points (xi,1), (1,yi) et (1,1)
[ v,s,t ] = vectdeny( x,y,Mx,My,kx,ky );
%% numat (la matrice A en tout point de collocation)
[ numat ] = Matnu(v,s,t,kx,ky,Mx,My,Px,Py,psim );
%% C la solution du problème
C=pinv(numat')*fctg;
[Uapp ] = sol_app( C,Px,Py,psim,kx,ky,Mx,My,x,y,v,s,t);
[Uex ] = sol_exa(x,y);
n=length(x);
m=length(y);
[X,Y]=meshgrid(x,y);
subplot(2,2,1)
surf(X,Y,Uex)
xlabel('x')
ylabel('y')
zlabel('u(x,y)')
grid on
title('la solution exacte')
[ uap ] = Matapp(n,m,Uapp);
subplot(2,2,2)
surf(X,Y,uap')
xlabel('x')
ylabel('y')
```

```

xlabel('u(x,y)')
grid on
title('la solution approchée')
subplot(2,2,[3 4])
surf(X,Y,abs(Uex-uap'))
xlabel('x')
ylabel('y')
xlabel('Err')
grid on
title('l'erreur')
uex=norm(Uex)
uapp=norm(uap)
err = norm(Uex-uap')
toc

```

### 4.2.1 Les sous programmes

```

%la matrice psi aux point 1
function [ v,s,t ] = vectdeny( x,y,Mx,My,kx,ky )
v=zeros(2^(kx-1)*2^(ky-1)*Mx*My);
s=zeros(2^(kx-1)*2^(ky-1)*Mx*My);
t=zeros(2^(kx-1)*2^(ky-1)*Mx*My);
i=0;
    for iy =1 :length(y)
        for ix =1 :length(x)
            i=i+1;
            V1=x(ix) .*MatLegWav(1,y(iy),Mx,My,kx,ky);
            v(:,i)=V1(:,1);
        end
    end
end
i=0;

```

```

for iy =1 :length(y)
    for ix =1 :length(x)
        i=i+1;
        V2=y(iy).* MatLegWav(x(ix),1,Mx,My,kx,ky) ;
        s( :,i)=V2( :,1) ;
    end
end
i=0;
for iy =1 :length(y)
    for ix =1 :length(x)
        i=i+1;
        V3=x(ix)*y(iy).* MatLegWav(1,1,Mx,My,kx,ky) ;
        t( :,i)=V3( :,1) ;
    end
end
end
function [ numat ] = Matnu(v,s,t,kx,ky,Mx,My,Px,Py,psim )
%numat les vecteurs correspondants aux points de collocation
numat = zeros(2^(kx -1)*2^(ky -1)*Mx*My,2^(kx -1)*2^(ky -1)*Mx*My) ;
i = 0 ;
for j=1 :2^(kx -1)*2^(ky -1)*Mx*My
    i = i+1 ;
    numat( :,i) = (Py^2+Px^2+2.*(Px^2*Py^2))*psim( :,j)-(Py^2+2.*(Px^2*Py^2))*s( :,j)...
        -(Px^2+2.*(Px^2*Py^2))*v( :,j)-(2.*(Px^2*Py^2))*t( :,i) ;
end
end
function [ fctg ] = Fonctiong(x,y,Mx,My,kx,ky)
fctg = zeros(2^(kx -1)*2^(ky -1)*Mx*My,1) ;
i = 0 ;
for iy = 1 :length(y)

```

```

    for ix =1 :length(x)
        i = i+1;
        fctg(i, :) =( -x(ix)*sin(1)*sin(y(iy)) - y(iy)*sin(x(ix))*sin(1) ...
            +2*x(ix)*y(iy)*sin(1)*sin(1));
    end
end
end
%La solution approchée
function [Uapp ] = sol_app( C,Px,Py,psim,kx,ky,Mx,My,x,y,v,s,t)
Uap = zeros(1,2^(kx-1)*Mx*2^(ky-1)*My);
u=C*(Px^2*Py^2*(psim-v-s))- C*(Px^2*Py^2*t);
i=0;
for iy = 1 :length(y)
    for ix =1 :length(x)
        i=i+1;
        Uap(:,i) = -x(ix)*y(iy)*sin(1)+x(ix)*sin(1)*sin(y(iy))-x(ix)*y(iy)*sin(1).*sin(1)...
            +y(iy)*x(ix)*sin(1)+y(iy)*sin(x(ix))*sin(1);
    end
end
Uapp = u+ Uap;
end
function [ uap ] = Matapp(n,m,Uapp)
uap= zeros(n,m);
i = 0;
for ix =1 :m
    i = i+1;
    uap(:,i) =Uapp(1,(ix-1)*n+1 :ix*n);
end
end
%la solution exacte

```

```
function [Uex ] = sol_exa(x,y)
    for iy = 1 :length(y)
        for ix =1 :length(x)
            Uex(iy,ix) =sin(x(ix))* sin(y(iy));
        end
    end
end
end
```

### 4.3 Le programme principal pour les conditions mixtes

```
tic
clc
clear all
close all
format short e
%% TestLegWav.m
%Dans ce programme on calcule le polynôme de Legendre et les ondelettes
% de Legendre et on détermine les matrices opérationnelles d'intégration
% et en trouve la solution de système C'A=g
% (Mx =M, My=M') le degré du polynôme de Legendre(x,y)
%(kx=k ,ky=k') les niveaux
%%exemple de l'équation de Poisson %%%
%%pour les condition mixtes%%%%%%%%
%% Set Prameters
Mx = 10;
My =10;
kx = 2;
ky = 2;
%% Construction of vect x & y
for i=1 :Mx*2^(kx-1)
    x(i)=(i-0.5)/(Mx*2^(kx-1));
```

```
end
x;
for ii=1 :My*2^(ky-1)
    y(ii)=(ii-0.5)/(My*2^(ky-1));
end
y;
%% Psi
[ psim ] = MatLegWav(x,y,Mx,My,kx,ky);
psim;
%% Px
[Px] = matrice_integracion(Mx,My,kx,ky);
%% Py
[Py] = matrice_integraciony(My,ky,Mx,kx);
Px;
Py;
%% fctg (la fonction g(x,y) en tout point de collocation)
[ fctg ] = Fonctiong(x,y,Mx,My,kx,ky);
%% numat (la matrice A en tout point de collocation)
[ numat ] = Matnu(kx,ky,Mx,My,Px,Py,psim );
%% C la solution du problème
C = pinv(numat)*fctg;
[Uapp ] = sol_app( C,Px,Py,psim,kx,ky,Mx,My,x,y);
[Uex ] = sol_exa(x,y);
n=length(x);
m=length(y);
[X,Y]=meshgrid(x,y);
subplot(2,2,1)
surf(X,Y,Uex)
xlabel('x')
ylabel('y')
```

```

xlabel('u(x,y)')
grid on
title('la solution exacte')
[ uap ] = Matapp(n,m,Uapp);
subplot(2,2,2)
surf(X,Y,uap')
xlabel('x')
ylabel('y')
xlabel('u(x,y)')
grid on
title('la solution approchée')
subplot(2,2,[3 4])
surf(X,Y,abs(Uex-uap'))
grid on
title('l'erreur')
uex=norm(Uex)
uapp=norm(uap)
err = norm(Uex-uap')
toc

```

### 4.3.1 Les sous programmes

```

function [ numat ] = Matnu(kx,ky,Mx,My,Px,Py,psim )
%numat les vecteurs correspondants aux points de collocation
numat = zeros(2^(kx -1)*2^(ky -1)*Mx*My,2^(kx -1)*2^(ky -1)*Mx*My);
i = 0;
    for j=1 :2^(kx -1)*2^(ky -1)*Mx*My
        i = i+1;
        numat( :,i) = (Py^2 + Px^2 + 2*Px^2*Py^2)*psim( :,j);
    end
end

```

```

function [ fctg ] = Fonctiong(x,y,Mx,My,kx,ky)
fctg = zeros(2^(kx -1)*2^(ky -1)*Mx*My,1);
i = 0;
    for iy = 1 :length(y)
        for ix =1 :length(x)
            i = i+1;
            fctg(i, :) = -x(ix)*sin(y(iy)) - y(iy)*sin(x(ix)) + 2*x(ix)*y(iy);
        end
    end
end
%la solution approchée
function [Uapp ] = sol_app( C,Px,Py,psim,kx,ky,Mx,My,x,y)
Uap = zeros(1,2^(kx-1)*Mx*2^(ky-1)*My);
u = C'*Px^2*Py^2*psim;
i=0;
    for iy = 1 :length(y)
        for ix =1 :length(x)
            i=i+1;
            Uap( :,i) = x(ix)*y(iy)-x(ix)*sin(y(iy))- y(iy)*sin(x(ix));
        end
    end
Uapp = u - Uap;
end
function [ uap ] = Matapp(n,m,Uapp)
uap= zeros(n,m);
i = 0;
    for ix =1 :m
        i = i+1;
        uap( :,i) =Uapp(1,(ix-1)*n+1 :ix*n);
    end
end

```



```
end
%la solution exacte
function [Uex ] = sol_exa(x,y)
    for iy = 1 :length(y)
        for ix =1 :length(x)
            Uex(iy,ix) =sin(x(ix))* sin(y(iy));
        end
    end
end
end
```

---