

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABDELHAMID BEN BADIS DE MOSTAGANEM
FACULTÉ DES SCIENCES EXACTES ET DE L'INFORMATIQUE
DÉPARTEMENT DE MATHÉMATIQUES ET INFORMATIQUE
FILÈRE : MATHÉMATIQUES



UNIVERSITE
Abdelhamid Ibn Badis
MOSTAGANEM

MÉMOIRE DE FIN D'ÉTUDES

Pour l'Obtention du Diplôme de Master en Mathématiques délivré par

Université de Mostaganem

Spécialité "Modélisation, Contrôle et Optimisation"

présenté par :

Nadjet AYAD

Formule d'intégration en deux dimensions basée sur les ondelettes de Legendre

soutenu le 27 Juin 2021 devant le jury composé de :

Président :	Louiza TABHARIT	MCB	UMAB
Examineur :	Leila BOUZID	MCB	UMAB
Encadreur :	Naima ABLAOUI-LAHMAR	MCB	UMAB

Année Universitaire : 2020 / 2021

M
A
S
T
E
R

Dédicaces

Je dédie ce travail a :

Ma mère et mon père qui sont mes points fortes et faibles en même temps dans ma vie.

Toutes mes sœurs, mon frère, les enfants de ma grande sœur et tout ma famille.

ma deuxième famille, toutes mes professeurs, mes amies et mes collègues.

Remerciements

Premièrement et avant tout nous remercions «Allah» Le tout puissant et le louons de nous avoir accordé la santé, le bien-être, la volonté de préparer ce travail et pour toutes ses innombrables bénédictions .

Je remercie mes parents pour les efforts qu'ils ont déployés pour préserver ma santé, mes principes humains et ma morale, ainsi que pour leur rôle dans le développement de ma vie pour le mieux.

Je remercie mon encadreur Mme ABLAOUI-LAHMAR Naima pour ses directives de base et son grand effort dans ce travail.

Je remercie tous mes professeurs et particulièrement ALIMERINA Houria, BOUZID Leila, HAMMOUMAAMAR Maghnia pour leurs conseils et orientations.

Enfin, je tiens à remercier toute ma famille, mes amis et collègues.

Table des matières

Liste des tableaux	5
Table des figures	6
Introduction	7
1 L'intégration numérique	10
1 Introduction	10
2 Formules des quadratures de type Newton-Côtes	10
2.1 Méthode de Newton-Côtes simples :	10
2.2 Lois de Newton-Côtes composites	13
3 Formules des cubatures	15
3.1 Formule de cubature de type trapèzes	15
3.2 Formule de cubature de type Simpson	16
2 Formule d'intégration des ondelettes de Legendre	18
1 Introduction	18
2 Définitions et notions préliminaires	18
2.1 Les polynômes de Legendre	18
2.2 Les ondelettes de Legendre	19
2.3 L'approximation dans l'espace $L^2([0, 1[\times]0, 1])$	19
3 Les matrices opérationnelles d'intégration en deux dimensions	20
3.1 La matrice opérationnelle d'intégration par rapport à x	20
3.2 Matrice opérationnelle d'intégration par rapport à y	24
4 La formule d'intégration des ondelettes de Legendre	25
3 Les tests numériques	27
1 Exemple 1	27
2 Exemple 2	29
3 Exemple 3	30
Conclusion et Perspectives	32
Bibliographie	32
Annexe	36
4 Les sous programmes	36
4.1 Polynôme de Legendre p_x et p_y	36
4.2 L'ondelette de Legendre	36
4.3 Le vecteur d'ondelette de Legendre	36

4.4	La matrice d'ondelette de Legendre	37
4.5	Les matrices d'intégrations P_x ; P_y	37
4.6	Les sous programmes des données, la solution exacte, le vecteur second membre, trapèze et Simpson :	41
5	Le programme principale	43

Liste des tableaux

1.1	Tableau des erreurs	13
3.1	Le tableau des erreurs pour $f(x, y) = x^4 + y^3$ et $k = k' = 2$:	28
3.2	Le tableau des erreurs pour $f(x, y) = x^4 + y^3$ et $k = k' = 3$:	28
3.3	Le tableau des erreurs pour $f(x, y) = x^2 e^y$ et $k = k' = 2$:	29
3.4	Le tableau des erreurs pour $f(x, y) = x^2 e^y$ et $k = k' = 3$:	29
3.5	Le tableau des erreurs pour $f(x, y) = \frac{x^2}{1+y^2}$ et $k = k' = 2$:	30
3.6	Le tableau des erreurs pour $f(x, y) = \frac{x^2}{1+y^2}$ et $k = k' = 3$:	30

Table des figures

1	Interprétation géométrique de l'intégrale double	9
1.1	Méthode du rectangle ($n = 0$)	11
1.2	Méthode des trapèzes ($p = 1$)	12
1.3	Méthode de Simpson simple ($p = 2$)	13
2.1	représentation de la subdivision du carré unité pour $k=k'=2$	22
3.1	Logarithme de l'erreur absolue pour $f(x, y) = x^4 + y^3$	28
3.2	Logarithme de l'erreur absolue pour $f(x, y) = x^2 e^y$	29
3.3	Logarithme de l'erreur absolue pour $f(x, y) = \frac{x^2}{1+y^2}$	31

Introduction

Le but de ce travail est de présenter une nouvelle formule numérique pour estimer l'intégrale double

$$I = \int_a^A \int_b^B f(x, y) dx dy,$$

où f est une fonction intégrable sur le pavé $[a, A] \times [b, B]$ et a, A, b, B sont des réels donnés.

Cette formule d'intégration est basée sur les matrices opérationnelles d'intégration, P_x, P_y , associées aux ondelettes de Legendre bidimensionnelles. Ces matrices ont été introduites en 2005 par H.Parsian [6]. Elles sont connues pour avoir été utilisées dans la résolution des problèmes aux limites d'ordre deux et en deux dimensions [5].

On commence par aborder le calcul général de l'intégrale d'une fonction $f(x)$ sur un domaine fini délimité par des bornes finies a et b (les cas des bornes infinies n'est donc pas couvert ici) :

$$I = \int_a^b f(x) dx.$$

On veut évaluer l'intégrale d'une fonction f sur un intervalle $[a; b]$. Si l'on connaît sa primitive F , alors

$$\int_a^b f(x) dx = F(b) - F(a) \text{ où } F' = f.$$

Mais dans de nombreux cas, la primitive U ne peut pas être connue comme cela est le cas pour $\int e^{-x^2} dx$ et $\int \frac{\sin x}{x} dx$.

Dans certains cas très limités, une telle intégrale peut être calculée analytiquement (à la main). Cependant, ce n'est que très rarement possible, et le plus souvent un des cas suivants se présente :

-Le calcul analytique est long, compliqué et barbatif.

-Le résultat de l'intégrale est une fonction compliquée qui fait appel à d'autres fonctions elles-même longues à évaluer.

-Cette intégrale n'a pas d'expression analytique.

Dans tous ces cas, on préférera calculer numériquement la valeur de l'intégrale I .

L'idée principale est de trouver des méthodes qui permettent de calculer rapidement une valeur approchée I_{app} de l'intégrale à calculer ($I \approx I_{app}$).

Une méthode bien connue consiste par exemple à diviser l'aire sous la courbe en un grand nombre de petits rectangles d'aire I_{appk} et de les sommer. Le résultat

$$I_{app} = \sum_k I_{appk}$$

est alors une approximation de l'intégrale I . Cette approximation est d'autant meilleure que la largeur h des rectangles tend vers 0, c'est à dire :

$$\lim_{h \rightarrow 0} (I - I_{app}) = 0.$$

Cette méthode dite des rectangles est un exemple parmi d'autres. On le verra, mais on verra aussi d'autres méthodes, plus générales et plus performantes[3].

Pour presque toutes les méthodes (sauf la méthode de Monte-Carlo), l'intégrale numérique est calculée à partir de l'évaluation de la fonction $f(x)$ en un nombre de points $n + 1$ distincts : $f_k = f(x_k)$, $k = 0, \dots, n$. Elle s'écrit alors :

$$I_{app} = (b - a) \sum_{k=0}^n w_k f_k$$

Dans ce cas, on parle de méthodes de quadrature. On va voir deux types de méthodes différentes :

1- Les méthodes de Newton-Cotes simples

2- Les méthodes de Newton-Cotes composites.

La performance d'une méthode se juge en comparant

- la précision du résultat : Celle-ci se caractérise en estimant l'erreur ε entre l'approximation et la valeur réelle de l'intégrale :

$$\varepsilon = I - I_{app}$$

La valeur de l'erreur ne peut pas être calculée exactement puisqu'en général, on ne connaît pas l'intégrale I que l'on cherche à estimer. Cependant, une majoration peut souvent être estimée.

- La rapidité d'exécution nécessaire pour atteindre ce résultat. De manière générale, toutes les méthodes peuvent atteindre de très grandes précisions. Cependant, le temps de calcul augmente avec la précision.

Ce temps n'augmente pas de la même manière pour toutes les méthodes si bien que certaines s'avèrent plus efficaces que d'autres. En particulier, le temps de calcul des méthodes de quadrature est proportionnel au nombre de points où la fonction $f(x)$ est évaluée[3]. On rappelle que notre objectif est l'estimation de l'intégrale double

$$I = \int_a^A \int_b^B f(x, y) dx dy,$$

qui a pour interprétation graphique, la figure suivante :

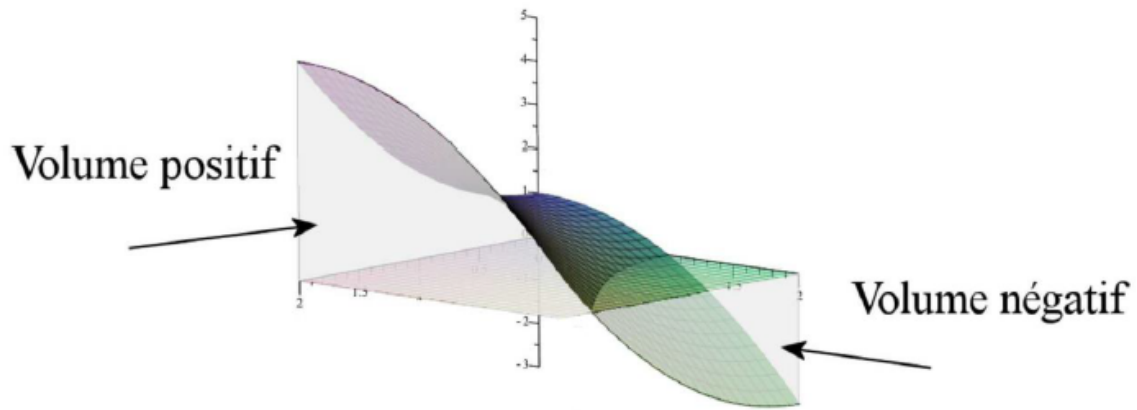


FIGURE 1 – Interprétation géométrique de l'intégrale double

Pour obtenir les formules de cubature, c'est à dire les formules numériques pour estimer les intégrales doubles, on s'appuie sur le théorème de Fubini en appliquant deux fois l'une des formules de quadratures successivement (comme cela a été fait dans [1]). Les formules de cubature de types trapèzes et Simpson composites vont être décrites dans le détail ultérieurement.

Ce manuscrit est structuré en trois chapitres :

Dans le premier chapitre, un rappel des méthodes d'intégration numériques est fait, où l'on s'intéresse aux formules de quadratures et formules du cubatures de type Newton-côtes. Dans le deuxième chapitre, on rappelle les définitions principales des polynômes de Legendre, des ondelettes de Legendre et on détermine par les matrices opérationnelles d'intégration notées dans Parsian P_x et P_y . Un résultat sur l'approximation des fonctions est rappelé aussi. On termine ce chapitre par l'établissement de la nouvelle formule d'intégration numérique.

Dans le dernier chapitre, on fait un traitement numérique sous forme d'exemples académiques. La nouvelle méthode d'intégration est comparée avec les méthodes composées de simpson et de trapèzes. Les résultats numériques obtenus, montrent l'efficacité de cette méthode [2].

On termine ce travail par une conclusion générale et des perspectives.

Chapitre 1

L'intégration numérique

1 Introduction

Dans ce chapitre, on va rappeler les méthodes d'intégration numériques les plus connues et les plus utilisées pour estimer $\int_a^b f(x)dx$ qui sont la méthode du rectangle, des trapèzes et la méthode de Simpson.

Le principe général des méthodes de Newton-Côtes simples est d'approximer la fonction $f(x)$ à intégrer par un polynôme $P(x) \approx f(x)$. Si cette approximation est suffisamment bonne alors, l'intégrale de ce polynôme : $I_{app} = \int_a^b P(x)dx$ sera une bonne approximation de $I = \int_a^b f(x)dx$. L'avantage est que l'on sait calculer analytiquement la valeur exacte de I_{app} . Dans ces méthodes, on choisit $P(x)$ comme étant des polynômes d'interpolation de degré $n \geq 0$.

Des polynômes de degrés différents définissent des méthodes différentes aux performances différentes.

On va voir les plus courantes, c'est à dire les méthodes d'ordres les plus bas[3].

2 Formules des quadratures de type Newton-Côtes

2.1 Méthode de Newton-Côtes simples :

2.1.1 La méthode du rectangle ($n = 0$)

Cette méthode utilise le polynôme de degré le plus bas, à savoir le polynôme constant : $P_0(x) = f(x_0) = f_0$, où x_0 est un point choisit dans l'intervalle $[a, b]$.

L'intégrale $I = \int_a^b f(x)dx$ est alors approchée par

$$I_{app0} = \int_a^b P_0(x)dx = (b-a)f_0.$$

Il s'agit de l'aire du rectangle de cotés $(b-a)$ et f_0 qui est représentée par la figure (1.1). Cette intégrale numérique nécessite une unique évaluation de la fonction f en x_0 . L'erreur de cette approximation est donnée par

$$\varepsilon_0 = \frac{h^2}{2} f'(\zeta); \text{ avec } \zeta \in]a, b[$$

qui peut être majorée par

$$|\varepsilon_0| \leq \frac{h^2}{2} \text{Sup}_{[a,b]}(|f'(x)|); \text{ où } h = b - a[3].$$

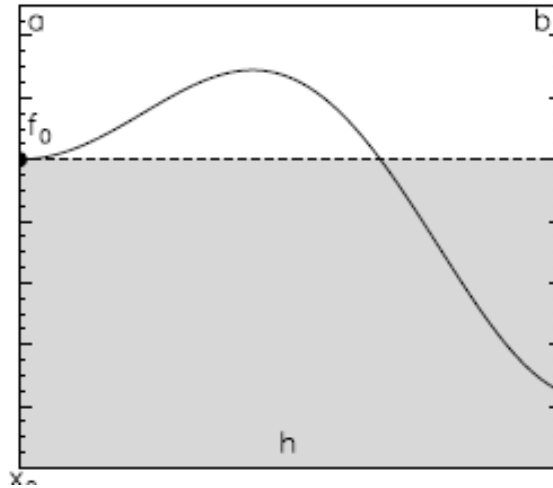


FIGURE 1.1 – Méthode du rectangle ($n = 0$)

Remarque 1.1 Dans le cas particulier où x_0 est le point milieu de l'intervalle $[a, b]$, cette méthode est appelée formule d'intégration du point milieu, elle s'exprime par

$$I = \int_a^b f(x) dx \simeq (b - a) f\left(\frac{a + b}{2}\right).$$

Elle a l'avantage de bien exploiter la symétrie de la fonction à intégrer[3].

2.1.2 La méthode des trapèzes ($n = 1$)

Pour approximer la fonction f , cette méthode utilise le polynôme P_1 d'ordre 1; (la droite) qui passe par les points $(a, f(a))$ et $(b, f(b))$ et qui est donné par :

$$P_1(x) = f_0 \frac{x - b}{a - b} + f_1 \frac{x - a}{b - a},$$

où $f_0 = f(a)$ et $f_1 = f(b)$.

L'intégrale approchée $I_{app1} = \int_a^b P_1(x) dx$ se calcule facilement et donne :

$$I = \int_a^b f(x) dx \simeq (b - a) \frac{f_0 + f_1}{2}.$$

Géométriquement, il s'agit de l'aire du trapèze illustrée dans la figure (1.2).

Cette méthode nécessite deux évaluations de la fonction f en a et en b [3].

L'erreur de cette approximation est donnée par :

$$\varepsilon_1 = \frac{h^3}{12} f''(\zeta); \text{ avec } \zeta \in]a, b[$$

qui peut être majorée par

$$|\varepsilon_1| \leq \frac{h^3}{12} \text{Sup}_{[a,b]}(|f''(x)|); \text{ où } h = b - a[3].$$

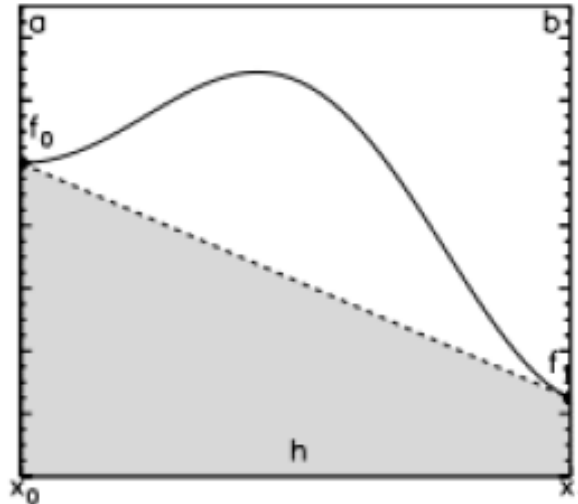


FIGURE 1.2 – Méthode des trapèzes ($p = 1$)

2.1.3 La méthode de Simpson simple ($n = 2$)

Pour approximer la fonction f , cette méthode utilise le polynôme P_2 de degré 2 (la parabole) qui passe par les trois points : $(a, f(a))$, $(\frac{a+b}{2}, f(\frac{a+b}{2}))$ et $(b, f(b))$ qui est donné par :

$$P_2(x) = f_0 \frac{(x - \frac{a+b}{2})(x - b)}{(a - \frac{a+b}{2})(a - b)} + f_1 \frac{(x - a)(x - b)}{(\frac{a+b}{2} - a)(\frac{a+b}{2} - b)} + f_2 \frac{(x - a)(x - \frac{a+b}{2})}{(b - a)(b - \frac{a+b}{2})},$$

où $f_0 = f(a)$, $f_1 = f(\frac{a+b}{2})$ et $f_2 = f(b)$.

L'intégrale approchée $I_{app2} = \int_a^b P_2(x) dx$ se calcule alors simplement et donne :

$$I = \int_a^b f(x) dx \approx (b - a) \frac{f_2 + 4f_1 + f_0}{6}.$$

Cette méthode nécessite trois évaluations de la fonction f en a , $(a + b)/2$ et b et est représentée géométriquement par la figure (1.3).

L'erreur de cette approximation est donnée par

$$\varepsilon_2 = -\frac{h^5}{90} f^{(4)}(\zeta), \text{ avec } \zeta \in]a, b[.$$

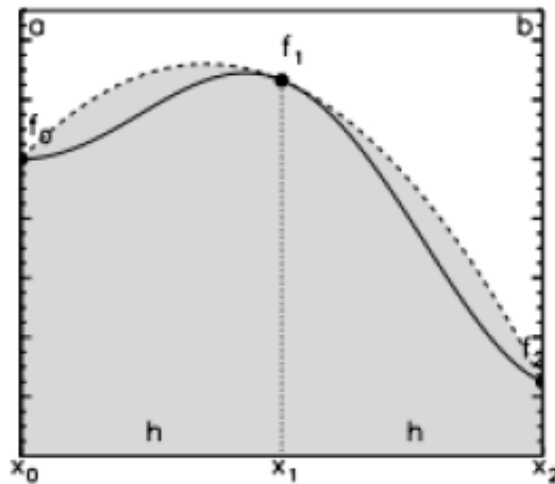
[3]

Remarque 1.2 Pour les différentes méthodes :

1. l'erreur n'est pas connue car la valeur de $\zeta \in]a, b[$ reste indéterminée. Cependant, on peut la majorer,
2. plus l'intervalle $[a, b]$ est petit, plus l'erreur est faible car l'erreur décroît en fonction de $h = b - a$,
3. de manière générale, la méthode de Simpson est plus précise que celle du rectangle et des trapèzes car les erreurs associées à ces méthodes vérifient

$$\frac{\varepsilon_2}{\varepsilon_0} = \theta(h^3), \quad \frac{\varepsilon_2}{\varepsilon_1} = \theta(h^2). \quad (1.1)$$

[3]

FIGURE 1.3 – Méthode de Simpson simple ($p = 2$)

Exemple 1.1 On va appliquer ces trois méthodes pour estimer les deux intégrales

$$I_1 = \int_0^1 \frac{dx}{1+x^2} \text{ et } I_2 = \int_0^{\frac{1}{2}} \frac{dx}{\sqrt{1-x^2}}.$$

On connaît la valeur exacte de ces intégrales, à savoir $I_1 = \arctan(1) = \frac{\pi}{4}$ et $I_2 = \arcsin(\frac{1}{2}) = \frac{\pi}{6}$.

Les erreurs obtenues suite à l'application de ces méthodes sont résumées dans le tableau :

Les erreurs	$\frac{1}{1+x^2}$	$\frac{1}{\sqrt{1-x^2}}$
ϵ_0	-0.01460183	-0.0537514936
ϵ_1	0.035398163	-0.015076359
ϵ_2	0.002064830	-0.0002247889

TABLEAU 1.1 – Tableau des erreurs

Remarque 1.3 Comme on le voit dans l'exemple précédent, les méthodes de Newton-Cotes simples ne permettent pas, à elles-seules, d'atteindre des précisions suffisantes sur des intervalles $[a, b]$ finis et en général, ne sont pas utilisées dans ce cas. En revanche, elles deviennent précises lorsque $|b - a| \rightarrow 0$. C'est cette idée qui constitue alors la base élémentaire des méthodes composites présentées dans la section suivante.

2.2 Lois de Newton-Côtes composites

2.2.1 Principe

L'idée est donc de découper le domaine total d'intégration $[a, b]$ en un nombre fini m de sous intervalles de petite longueur. Dans chaque sous domaine, on applique une des méthodes de Newton-Cotes simples.

Soit I_{appk} l'approximation de l'intégrale de la fonction f sur le k ième domaine pour $k = 1, \dots, m$ on en déduit une approximation de l'aire totale par une simple somme :

$$I_{app} = \sum_{k=0}^{m-1} I_{appk}.$$

Lorsque m est suffisamment grand, la longueur $(b-a)/m$ des intervalles devient aussi petite que l'on veut, si bien que ces méthodes peuvent atteindre des précisions aussi grandes que nécessaire, sans pour autant se heurter au problème des polynômes de grand degré.

Plus ce nombre est élevé, plus la méthode est précise, mais plus lente.

En utilisant sur chaque intervalle des méthodes de Newton-Côtes simples de degrés différents, on obtient des méthodes composites aux propriétés et aux performances différentes.

Dans certains cas, ces m intervalles peuvent être espacés de manière non régulière pour mieux représenter une zone où la fonction $f(x)$ varie beaucoup, mais dans cette partie, on se limite au cas d'intervalles réguliers[3].

2.2.2 La méthode des trapèzes composite

La méthode des trapèzes composite applique la méthode des trapèzes simple ($n = 1$) sur chacun des m sous-intervalles de $[a, b]$. Soit $(x_i)_{0 \leq i \leq m}$ la subdivision associée à l'intervalle $[a, b]$ vérifiant :

$$a = x_0 < x_1 < \dots < x_m = b.$$

Dans chaque sous intervalle $[x_k, x_{k+1}]$; $k = 0, \dots, m-1$, l'intégrale $\int_{x_k}^{x_{k+1}} f(x) dx$ est approchée par :

$$I_{app1,k} = (x_{k+1} - x_k) \frac{f_k + f_{k+1}}{2}; \text{ où } f_k = f(x_k).$$

L'intégrale totale étant la somme des $I_{app1,k}$ pour $k = 0, \dots, m-1$ est alors approchée par :

$$I_{app1} = \frac{h}{2} (f_0 + 2 \sum_{k=1}^{m-1} I_{appk} + I_{appm}); \text{ où } h = \frac{b-a}{m}.$$

Par conséquent, l'erreur étant la somme de toutes les erreurs, elle est donnée par :

$$\epsilon_1 = -\frac{h^3}{12} m f''(\zeta), \text{ avec } \zeta \in]a, b[.$$

[3]

2.2.3 La méthode de Simpson composite

Pour la méthode de Simpson composite, le nombre total de sous-intervalles est $n = 2m$.

Dans l'intervalle $[a, b]$, on choisit la subdivision $(x_i)_{0 \leq i \leq 2m}$ vérifiant :

$$a = x_0 < x_1 < \dots < x_{2m} = b$$

Dans chaque sous domaine $[x_k, x_{k+2}]$, l'intégrale est approchée par :

$$I_{app2,k} = (x_{k+2} - x_k) \frac{f_k + 4f_{k+1} + f_{k+2}}{6}$$

Si bien que l'intégrale totale est approchée par :

$$I_{app2} = \frac{h}{3} \left(f_0 + 4 \sum_{k=0}^{n/2-1} f_{2k+1} + 2 \sum_{k=0}^{n/2-1} f_{2k} + f_n \right); \text{ où } h = \frac{b-a}{2m},$$

l'erreur étant simplement la somme de toutes les erreurs, elle est exprimée par

$$\epsilon_2 = -\frac{h^5}{90} m f^{(4)}(\zeta).$$

[3]

Remarque 1.4 1. On peut construire des méthodes composites d'ordres plus élevés en utilisant des polynômes d'interpolation de degrés plus élevés.

2. La précision de toutes les méthodes de Newton-Cote augmente avec le nombre de points utilisés.
3. Numériquement, chaque addition génère une petite erreur d'arrondi machine (l'erreur relative est de l'ordre de 10^{-17} pour des réels en doubles précision). Lorsque l'on somme beaucoup de nombres, les erreurs de chaque addition s'ajoutent et l'erreur relative totale augmente. Dans le cas des méthodes de Newton-Cotes composites, une erreur due aux arrondis machine s'ajoute donc aux erreurs de troncations étudiées précédemment. Et cette erreur augmente avec le nombre de points. En pratique, on ne peut donc pas augmenter infiniment le nombre d'intervalles.
4. Notre objectif étant l'approximation des intégrales doubles, on va présenter les formules de cubatures associées aux formules de Newton-Cote composites[3].

3 Formules des cubatures

On choisit pour domaine d'intégration le rectangle $R = \{(x, y) \in \mathbb{R}^2; a \leq x \leq A; b \leq y \leq B\}$ où a, A, b et B sont des réels donnés. L'approximation de l'intégrale

$$\int_{(R)} f(x, y) dx dy = \int_a^A \int_b^B f(x, y) dy dx,$$

est obtenue par l'application successive des formules de Newton-Côtes composites[1].

3.1 Formule de cubature de type trapèzes

On divise respectivement chacun des segments $[a, A]$ et $[b, B]$ en n et m parties égales par les points

$$x_i = a + ik; y_j = b + jh \text{ où } i = 0, 1, 2, \dots, n, j = 0, 1, 2, \dots, m \text{ où } k = \frac{A-a}{n}, h = \frac{B-b}{m}.$$

D'après le théorème de Fubini on a :

$$\int_{(R)} f(x, y) dx dy = \int_a^A \left(\int_b^B f(x, y) dy \right) dx. \quad (1.2)$$

En appliquant la formule de quadrature de trapèze composite par rapport à la deuxième variable y , on obtient :

$$\int_{(R)} f(x, y) dx dy \simeq \int_a^A \left[\frac{B-b}{2m} \left(f(x, y_0) + 2 \sum_{j=1}^{m-1} f(x, y_j) + f(x, y_m) \right) \right] dx.$$

On applique la formule de trapèze une deuxième fois par rapport à la variable x , on obtient :

$$\int_{(R)} f(x, y) dx dy \simeq \frac{A-a}{2n} \left[\frac{B-b}{2m} \left[(f(x_0, y_0) + 2 \sum_{j=1}^{m-1} f(x_0, y_j) + f(x_0, y_m)) + 2 \sum_{i=1}^{n-1} (f(x_i, y_0) + 2 \sum_{j=1}^{m-1} f(x_i, y_j) + f(x_i, y_m)) + (f(x_n, y_0) + 2 \sum_{j=1}^{m-1} f(x_n, y_j) + f(x_n, y_m)) \right] \right]$$

En arrangeant les termes, on a

$$\int_{(R)} f(x, y) dx dy \simeq \frac{(A-a)(B-b)}{4nm} [(f(x_0, y_0) + f(x_0, y_m) + f(x_n, y_0) + f(x_n, y_m)) + 2 \sum_{j=1}^{m-1} (f(x_0, y_j) + f(x_n, y_j)) + 2 \sum_{i=1}^{n-1} (f(x_i, y_0) + f(x_i, y_m)) + 4 \sum_{i=1}^{n-1} \sum_{j=1}^{m-1} f(x_i, y_j)].$$

On note $f(x_i, y_j) = f_{i,j}$, alors la formule de cubature de trapèzes est donnée par

$$\int_{(R)} f(x, y) dx dy \simeq \frac{kh}{4} \left(f_{0,0} + f_{0,m} + f_{n,0} + f_{n,m} + 2 \left(\sum_{j=1}^{m-1} (f_{0,j} + f_{n,j}) + \sum_{i=1}^{n-1} (f_{i,0} + f_{i,m}) + 2 \sum_{i=1}^{n-1} \sum_{j=1}^{m-1} f_{i,j} \right) \right) \quad [1] \quad (1.3)$$

3.2 Formule de cubature de type Simpson

On va suivre les mêmes étapes qui ont permis d'aboutir à la formule (1.3).

On divise respectivement chacun des segments $[a, A]$ et $[b, B]$ en $2n$ et $2m$ parties égales par les points

$$x_i = a + ik, y_j = b + jh \text{ où } i = 0, 1, 2, \dots, 2n, j = 0, 1, 2, \dots, 2m \text{ où } k = \frac{A-a}{2n}, h = \frac{B-b}{2m}.$$

D'après le théorème de Fubini on a :

$$\int_{(R)} f(x, y) dx dy = \int_a^A \left(\int_b^B f(x, y) dy \right) dx. \quad (1.4)$$

En appliquant la formule de quadrature de Simpson composite par rapport à la deuxième variable y , on obtient :

$$\int_{(R)} f(x, y) dx dy \simeq \int_a^A \left[\frac{h}{3} \left(f(x, y_0) + 2 \sum_{j=1}^{m-1} f(x, y_{2j}) + 4 \sum_{j=0}^{m-1} f(x, y_{2j+1}) + f(x, y_{2m}) \right) \right] dx.$$

On applique la formule de Simpson une deuxième fois par rapport à la variable x , on obtient la formule de cubature de Simpson

$$\int_{(R)} f(x, y) dx dy \simeq \frac{hk}{9} \left[\begin{aligned} & \left(f(x_0, y_0) + 2 \sum_{j=1}^{m-1} f(x_0, y_{2j}) + 4 \sum_{j=0}^{m-1} f(x_0, y_{2j+1}) + f(x_0, y_{2m}) \right) \\ & + \left(f(x_{2n}, y_0) + 2 \sum_{j=1}^{m-1} f(x_{2n}, y_{2j}) + 4 \sum_{j=0}^{m-1} f(x_{2n}, y_{2j+1}) + f(x_{2n}, y_{2m}) \right) \\ & + 2 \sum_{i=1}^{n-1} \left(f(x_{2i}, y_0) + 2 \sum_{j=1}^{m-1} f(x_{2i}, y_{2j}) + 4 \sum_{j=0}^{m-1} f(x_{2i}, y_{2j+1}) + f(x_{2i}, y_{2m}) \right) \\ & + 4 \sum_{i=0}^{n-1} \left(f(x_{2i+1}, y_0) + 2 \sum_{j=1}^{m-1} f(x_{2i+1}, y_{2j}) + 4 \sum_{j=0}^{m-1} f(x_{2i+1}, y_{2j+1}) + f(x_{2i+1}, y_{2m}) \right) \end{aligned} \right].$$

On note $f(x_i, y_j) = f_{i,j}$, cette formule prend l'expression suivante :

$$\int_{(R)} f(x, y) dx dy \simeq \frac{hk}{9} \left[\begin{aligned} & f_{0,0} + f_{0,2m} + f_{2n,0} + f_{2n,2m} + 2 \sum_{j=1}^{m-1} (f_{0,2j} + f_{2n,2j}) + 4 \sum_{j=0}^{m-1} (f_{0,2j+1} + f_{2n,2j+1}) \\ & + 2 \sum_{i=1}^{n-1} (f_{2i,0} + f_{2i,2m}) + 4 \sum_{i=1}^{n-1} \sum_{j=1}^{m-1} f_{2i,2j} + 8 \sum_{i=1}^{n-1} \sum_{j=0}^{m-1} (f_{2i,2j+1} + f_{2i+1,2j}) \\ & + 4 \sum_{i=0}^{n-1} (f_{2i+1,0} + f_{2i+1,2m}) + 16 \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} f_{2i+1,2j+1} \end{aligned} \right].$$

On peut reformuler cette approximation comme suit

$$\int_{(R)} f(x, y) dx dy \simeq \frac{kh}{9} \sum_{i=0}^n \sum_{j=0}^m \lambda_{ij} f_{ij}. \quad (1.5)$$

tel que les coefficients λ_{ij} sont les éléments correspondants á la matrice

$$S = \begin{bmatrix} 1 & 4 & 2 & 4 & 2 & \dots & 4 & 2 & 4 & 1 \\ 4 & 16 & 8 & 16 & 8 & \dots & 16 & 8 & 16 & 4 \\ 2 & 8 & 4 & 8 & 4 & \dots & 8 & 4 & 8 & 2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 2 & 8 & 4 & 8 & 4 & \dots & 8 & 4 & 8 & 2 \\ 4 & 16 & 8 & 16 & 8 & \dots & 16 & 8 & 16 & 4 \\ 1 & 4 & 2 & 4 & 2 & \dots & 4 & 2 & 4 & 1 \end{bmatrix} [1]. \quad (1.6)$$

Chapitre 2

Formule d'intégration des ondelettes de Legendre

1 Introduction

L'objectif de ce chapitre est d'élaborer une nouvelle formule d'intégration numérique basée sur les ondelettes de Legendre en deux dimensions ainsi que sur les matrices opérationnelles d'intégration associées à chaque variable d'intégration. Pour cela, on commence par rappeler les outils mathématiques de base ; les polynômes de Legendre, les ondelettes de Legendre en 2D, les résultats d'approximation,...

2 Définitions et notions préliminaires

2.1 Les polynômes de Legendre

Définition 2.1 Les polynômes de Legendre, notés L_n ; $n \geq 0$, sont des polynômes de degré n , définis dans [9] sur l'intervalle $[-1, 1]$, comme étant les solutions de l'équation différentielle, dite de Legendre

$$(1 - x^2) \frac{d^2 L_n}{dx^2}(x) - 2x \frac{dL_n}{dx}(x) + n(n+1)L_n(x) = 0; \quad -1 < x < 1.$$

qui est équivalente à :

$$[(1 - x^2)L_n'(x)]' + n(n+1)L_n(x) = 0; \quad -1 < x < 1.$$

Proposition 2.1 Les polynômes de Legendre $\{L_n\}_{n \geq 0}$ définis dans ([8], [4]) sur l'intervalle $[-1, 1]$ vérifient la relation de récurrence

$$\begin{cases} L_0(x) = 1; L_1(x) = x \\ \forall n \geq 1 : (2n+1)xL_n(x) = (n+1)L_{n+1}(x) - nL_{n-1}(x) \end{cases} .$$

Proposition 2.2 Les polynômes de Legendre $\{L_n\}_{n \geq 0}$ sont caractérisés par deux autres représentations :

1. Ils sont définis par la formule dite de Rodrigue

$$L_n(x) = \frac{1}{n!2^n} \frac{d^n}{dx^n} (x^2 - 1)^n; \quad n \geq 0.$$

2. Ils admettent la représentation intégrale

$$L_n(x) = \frac{1}{\pi} \int_0^\pi |x + \sqrt{1-x^2} \cos(t)| dt.$$

2.2 Les ondelettes de Legendre

Dans toute la suite, on note par k, k' deux entiers positifs non nuls fixés. Le pavé $I = [0, 1] \times [0, 1]$ est subdivisé en $2^{k-1}2^{k'-1}$ sous intervalles notés $I_{n,n'}$, tels que

$$[0, 1] \times [0, 1] = \bigcup_{1 \leq n \leq 2^{k-1}, 1 \leq n' \leq 2^{k'-1}} I_{n,n'},$$

avec

$$I_{n,n'} = \left[\frac{n-1}{2^{k-1}}, \frac{n}{2^{k-1}} \right] \times \left[\frac{n'-1}{2^{k'-1}}, \frac{n'}{2^{k'-1}} \right]$$

Les ondelettes sont une famille de fonctions construites à partir de la dilatation et de la translation d'une fonction de moyenne finie, appelée ondelette mère. Si cette ondelette mère est notée Ψ , si a est le paramètre de dilatation et b le paramètre de translation, alors la famille d'ondelettes notées $\Psi_{a,b}$, est définie par :

$$\Psi_{a,b}(x) = |a|^{-\frac{1}{2}} \Psi\left(\frac{x-b}{a}\right); a \neq 0$$

Définition 2.2 (Ondelettes de Legendre en deux dimensions)[6]

Les ondelettes de Legendre en deux dimensions, sont définies sur le carré $I = [0, 1] \times [0, 1]$ comme suit :

$$\Psi_{nmn'm'}(x, y) = \begin{cases} 2^{\frac{k}{2}} 2^{\frac{k'}{2}} \sqrt{m + \frac{1}{2}} \sqrt{m' + \frac{1}{2}} L_m(2^k x - 2n + 1) L_{m'}(2^{k'} y - 2n' + 1) \\ \text{si } \left(\frac{n-1}{2^{k-1}} \leq x < \frac{n}{2^{k-1}} \text{ et } \frac{n'-1}{2^{k'-1}} \leq y < \frac{n'}{2^{k'-1}}\right) \\ 0 \text{ ailleurs} \end{cases}, \quad (2.1)$$

où $m, m' \geq 0, n = 1, \dots, 2^{\frac{k}{2}}, n' = 1, \dots, 2^{\frac{k'}{2}}$ et L_m est le polynôme de Legendre de degré m .

Proposition 2.3 La famille d'ondelettes définie dans (2.1), est orthonormée dans $L^2(I)$;

$$\int_0^1 \int_0^1 \Psi_{nmn'm'}(x, y) \Psi_{n_1 m_1 n'_1 m'_1}(x, y) dx dy = \delta_{nn_1} \delta_{mm_1} \delta_{n'n'_1} \delta_{m'm'_1},$$

où

$$\delta_{nm} = \begin{cases} 1 \text{ si } m = n \\ 0 \text{ sinon} \end{cases}$$

est le symbole de Kronecker.

2.3 L'approximation dans l'espace $L^2([0, 1] \times [0, 1])$

2.3.1 La décomposition dans $L^2([0, 1] \times [0, 1])$

Théorème 2.1 La famille $\{\Psi_{nmn'm'}\}_{1 \leq n \leq 2^{k-1}, 1 \leq n' \leq 2^{k'-1}, m, m' \geq 0}$ est une base hilbertienne de $L^2([0, 1] \times [0, 1])$.

Toute fonction $f(x, y) \in L^2([0, 1] \times [0, 1])$, peut être décomposée comme :

$$f(x, y) = \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{\infty} \sum_{n'=1}^{2^{k'-1}} \sum_{m'=0}^{\infty} C_{nmn'm'} \Psi_{nmn'm'}(x, y), \quad (2.2)$$

où les coefficients de cette décomposition sont donnés par :

$$C_{nmn'm'} = \langle f, \Psi_{nmn'm'} \rangle_{L^2(I)} = \int_0^1 \int_0^1 f(x, y) \Psi_{nmn'm'}(x, y) dx dy,$$

et $\langle \cdot, \cdot \rangle$ est le produit scalaire de $L^2([0, 1] \times [0, 1])$.

Remarque 2.1 Pour des considérations numériques, on choisit deux entiers M, M' et on considère la troncature de la série (2.2) ce qui conduit à l'approximation par la série finie :

$$f(x, y) \simeq \sum_{n=1}^{2^{k-1}M-1} \sum_{m=0}^{2^{k'-1}M'-1} \sum_{n'=1}^{2^{k-1}M-1} \sum_{m'=0}^{2^{k'-1}M'-1} C_{nmn'm'} \Psi_{nmn'm'}(x, y), \quad (2.3)$$

qu'on peut exprimer sous la forme contractée :

$$f(x, y) = C^T \Psi(x, y), \quad (2.4)$$

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

$$C = \begin{pmatrix} C_{1,0,1,0}, \dots, C_{1,0,1,M'-1}, C_{1,0,2,0}, \dots, C_{1,0,2,M'-1}, \dots, C_{1,0,2^{k'-1},0}, \dots, \\ C_{1,0,2^{k'-1},M'-1}, \dots, C_{1,M-1,2^{k'-1},0}, \dots, C_{1,M-1,2^{k'-1},M'-1}, C_{2,0,1,0}, \dots, \\ C_{2,0,1,M'-1}, \dots, C_{2^{k-1},M-1,2^{k'-1},0}, \dots, C_{2^{k-1},M-1,2^{k'-1},M'-1} \end{pmatrix}. \quad (2.5)$$

et

$$\Psi = \begin{pmatrix} \Psi_{1,0,1,0}, \dots, \Psi_{1,0,1,M'-1}, \Psi_{1,0,2,0}, \dots, \Psi_{1,0,2,M'-1}, \dots, \Psi_{1,0,2^{k'-1},0}, \dots, \\ \Psi_{1,0,2^{k'-1},M'-1}, \dots, \Psi_{1,M-1,2^{k'-1},0}, \dots, \Psi_{1,M-1,2^{k'-1},M'-1}, \Psi_{2,0,1,0}, \dots, \\ \Psi_{2,0,1,M'-1}, \dots, \Psi_{2^{k-1},M-1,2^{k'-1},0}, \dots, \Psi_{2^{k-1},M-1,2^{k'-1},M'-1} \end{pmatrix}. \quad (2.6)$$

3 Les matrices opérationnelles d'intégration en deux dimensions

Soit $\Psi(x, y)$ la famille d'ondelettes de Legendre en deux dimensions donnée en (2.6).

3.1 La matrice opérationnelle d'intégration par rapport à x

Théorème 2.2 La matrice opérationnelle d'intégration P_x [6] est entièrement déterminée par la formule :

$$\int_0^x \Psi(\tau, y) d\tau = P_x \Psi(x, y). \quad (2.7)$$

La matrice P_x est d'ordre $2^{k-1}2^{k'-1}MM'$ et est donnée par :

$$P_x = \frac{1}{2^k} \begin{bmatrix} L & F & F & \dots & F \\ O & L & F & \dots & F \\ O & O & L & \dots & F \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ O & O & O & \dots & L \end{bmatrix},$$

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

$$F = \begin{bmatrix} 2D & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$

$$L = \begin{bmatrix} D & \frac{1}{\sqrt{3}}D & 0 & 0 & \dots & 0 & 0 \\ -\frac{1}{\sqrt{3}}D & 0 & \frac{1}{\sqrt{3}\sqrt{5}}D & 0 & \dots & 0 & 0 \\ 0 & -\frac{1}{\sqrt{3}\sqrt{5}}D & 0 & \frac{1}{\sqrt{3}\sqrt{5}\sqrt{7}}D & \dots & 0 & 0 \\ 0 & 0 & -\frac{1}{\sqrt{3}\sqrt{5}\sqrt{7}}D & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \frac{1}{\sqrt{2M-3}\sqrt{2M-5}} & 0 \\ 0 & 0 & 0 & \ddots & -\frac{1}{\sqrt{2M-3}\sqrt{2M-5}} & 0 & \frac{1}{\sqrt{2M-1}\sqrt{2M-3}} \\ 0 & 0 & 0 & \dots & 0 & -\frac{1}{\sqrt{2M-1}\sqrt{2M-3}} & 0 \end{bmatrix}. \quad (2.8)$$

D est la matrice identité et 0 est la matrice nulle, les deux sont d'ordre $2^{k'-1}M'$.

Preuve. On rappelle l'expression : $\Psi_{nmn'm'}(x, y) = \Psi_{nm}(x)\Psi_{n'm'}(y)$ [11].

Pour la détermination de la matrice P_x , on exprime $\int_0^x \Psi_{nmn'm'}(\tau, y) d\tau$ en fonction des éléments du vecteur $\Psi(x, y)$. Ces expressions sont données dans [11] par :

Pour $m = 0$:

$$= \int_0^x \Psi_{n0n'm'}(\tau, y) d\tau = \begin{cases} 0 & \text{si } x \in [\frac{\tilde{n}-1}{2^{k-1}}, \frac{\tilde{n}}{2^{k-1}}[; \tilde{n} = 1, \dots, n-1, \forall y \\ \frac{1}{2^k} (\frac{1}{\sqrt{3}}\Psi_{n1n'm'}(x, y) + \Psi_{n0n'm'}(x, y)) & \\ \text{si } (x, y) \in [\frac{n-1}{2^{k-1}}, \frac{n}{2^{k-1}}[\times [\frac{n'-1}{2^{k'-1}}, \frac{n'}{2^{k'-1}}[\\ \frac{1}{2^{k-1}}\Psi_{n0n'm'}(x, y) & \text{si } x \in [\frac{\tilde{n}-1}{2^{k-1}}, \frac{\tilde{n}}{2^{k-1}}[; \tilde{n} = n+1, \dots, 2^{k-1}, \forall y \end{cases},$$

qui prend la forme

$$= \int_0^x \Psi_{n0n'm'}(\tau, y) d\tau = \frac{1}{2^k} (\frac{1}{\sqrt{3}}\Psi_{n1n'm'}(x, y) + \Psi_{n0n'm'}(x, y) + 2 \sum_{\tilde{n}=n+1}^{\tilde{n}=2^{k-1}} \Psi_{n0n'm'}(x, y)).$$

Pour $1 \leq m \leq M-2$,

$$= \int_0^x \Psi_{nmn'm'}(\tau, y) d\tau = \begin{cases} 0 & \text{si } x \leq \frac{n-1}{2^{k-1}} \text{ et } \forall y \\ \frac{1}{2^k} \frac{1}{\sqrt{2m+1}} (\frac{1}{\sqrt{2m+3}}\Psi_{nm+1n'm'}(x, y) - \frac{1}{\sqrt{2m+1}}\Psi_{nm-1n'm'}(x, y)) & \\ \text{si } (x, y) \in [\frac{n-1}{2^{k-1}}, \frac{n}{2^{k-1}}[\times [\frac{n'-1}{2^{k'-1}}, \frac{n'}{2^{k'-1}}[\\ 0 & \text{si } \frac{n}{2^{k-1}} \leq x \text{ et } \forall y \end{cases} = \frac{1}{2^k} \frac{1}{\sqrt{2m+1}} (\frac{1}{\sqrt{2m+3}}\Psi_{nm+1n'm'}(x, y) - \frac{1}{\sqrt{2m+1}}\Psi_{nm-1n'm'}(x, y))$$

Pour $m = M - 1$,

$$\int_0^x \Psi_{nM-1n'm'}(\tau, y) d\tau \simeq \begin{cases} 0 & \text{si } x \leq \frac{n-1}{2^{k-1}} \text{ et } \forall y \\ \frac{-1}{2^k \sqrt{2m+1} \sqrt{2m+3}} \Psi_{nM-2n'm'}(x, y) & \text{si } (x, y) \in \left[\frac{n-1}{2^{k-1}}; \frac{n}{2^{k-1}} \right] \times \left[\frac{n'-1}{2^{k'-1}}; \frac{n'}{2^{k'-1}} \right] \\ 0 & \text{si } \frac{n}{2^{k-1}} \leq x \text{ et } \forall y \end{cases}$$

$$\simeq \frac{-1}{2^k \sqrt{2m+1} \sqrt{2m+3}} \Psi_{nM-2n'm'}(x, y).$$

■

Exemple 2.1 Pour faciliter la compréhension de la preuve, on va prendre l'exemple où $k = k' = 2$ et $M = M' = 3$.

Le carré unité est subdivisé en $2^{k-1}2^{k'-1} = 4$ sous domaines comme cela est illustré dans la figure(2.1)

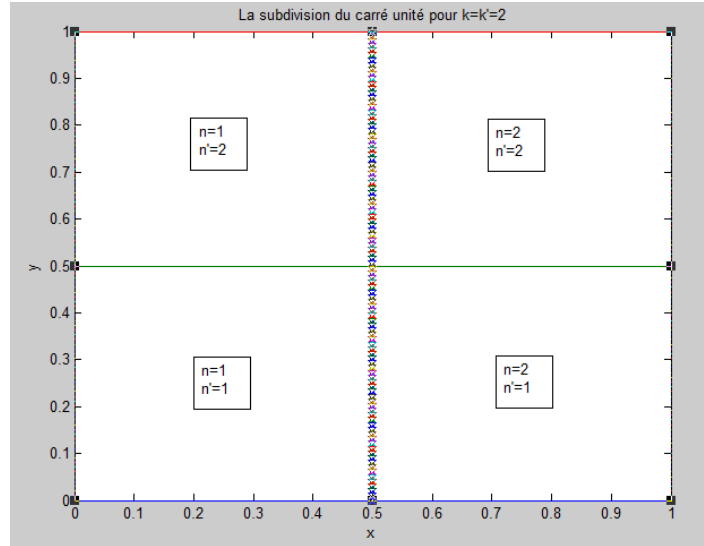


FIGURE 2.1 – représentation de la subdivision du carré unité pour $k=k'=2$

La taille du vecteur Ψ est $2^{k-1}2^{k'-1}MM' = 36$. Ce vecteur est donnée par

$$\Psi(x, y) \quad (2.9)$$

$$= \begin{bmatrix} \Psi_{1010}(x, y), \Psi_{1011}(x, y), \Psi_{1012}(x, y), \Psi_{1020}(x, y), \dots, \\ \Psi_{1110}(x, y), \dots, \Psi_{2220}(x, y), \Psi_{2221}(x, y), \Psi_{2222}(x, y) \end{bmatrix}.$$

On sait que pour obtenir la matrice P_x , on doit exprimer $\int_0^x \Psi_{nmn'm'}(\tau, y) d\tau$, en fonction de Ψ où $1 \leq n, n' \leq 2$ et $0 \leq m, m' \leq 2$ avec

$$\Psi_{nmn'm'}(x, y) = \begin{cases} 2^{\frac{k}{2}} 2^{\frac{k'}{2}} \sqrt{m + \frac{1}{2}} \sqrt{m' + \frac{1}{2}} L_m(2^k x - 2n + 1) L_{m'}(2^{k'} y - 2n' + 1) \\ \text{si } \frac{n-1}{2^{k-1}} \leq x \leq \frac{n}{2^{k-1}} \text{ et } \frac{n'-1}{2^{k'-1}} \leq y \leq \frac{n'}{2^{k'-1}} \\ 0 \text{ ailleurs} \end{cases}.$$

Par exemple l'expression de l'intégrale de la première composante $\int_0^x \Psi_{1010}(\tau, y) d\tau$ fournie la première ligne de la matrice P_x .

On note $I_{nn'} = \left\{ \frac{n-1}{2^{k-1}} \leq x \leq \frac{n}{2^{k-1}} \text{ et } \frac{n'-1}{2^{k'-1}} \leq y \leq \frac{n'}{2^{k'-1}} \right\}$

On a besoin pour ce calcul, des trois polynômes de Legendre L_0 , L_1 et L_2 .

$$\begin{aligned} L_0(4x - 2n + 1) &= 1; & L_1(4x - 2n + 1) &= 4x - 2n + 1; \\ L_2(4x - 2n + 1) &= \frac{3}{2}(4x - 2n + 1)^2 - \frac{1}{2}. \end{aligned}$$

L'expression de quelques ondelettes :

$$\begin{aligned} \Psi_{n0n'0}(x, y) &= \begin{cases} 2 \times 2 \sqrt{\frac{1}{2}} \sqrt{\frac{1}{2}} L_0(4x - 2n + 1) L_0(4y - 2n' + 1) \\ \text{si } (x, y) \in I_{nn'} \\ 0 \text{ ailleurs} \end{cases} \\ \Psi_{n1n'0}(x, y) &= \begin{cases} 2 \times 2 \sqrt{\frac{3}{2}} \sqrt{\frac{1}{2}} (4x - 2n + 1) L_0(4y - 2n' + 1) \\ \text{si } (x, y) \in I_{nn'} \\ 0 \text{ ailleurs} \end{cases} \\ \Psi_{n2n'0}(x, y) &= \begin{cases} 2 \times 2 \sqrt{\frac{5}{2}} \sqrt{\frac{1}{2}} [\frac{3}{2}(4x - 2n + 1)^2 - \frac{1}{2}] L_0(4y - 2n' + 1) \\ \text{si } (x, y) \in I_{nn'} \\ 0 \text{ ailleurs} \end{cases}. \end{aligned}$$

On s'intéresse au calcul de :

$$\begin{aligned} \int_0^x \Psi_{11n'0}(\tau, y) d\tau &= \begin{cases} 2 \times 2 \sqrt{\frac{3}{2}} \sqrt{\frac{1}{2}} \int_0^x (4\tau - 1) L_0(4y - 2n' + 1) d\tau \\ \text{si } (x, y) \in I_{1n'} \\ 0 \text{ ailleurs} \end{cases}. \\ \int_0^x \Psi_{11n'0}(\tau, y) d\tau &= \begin{cases} 2 \times 2 \sqrt{\frac{3}{2}} \sqrt{\frac{1}{2}} (2x^2 - x) L_0(4y - 2n' + 1) & \text{si } x < \frac{1}{2} \\ 0 & \text{si } \frac{1}{2} \leq x \end{cases} \\ \text{car} & \int_0^{\frac{1}{2}} (4\tau - 1) d\tau = 0. \end{aligned}$$

On vérifie facilement que

$$2x^2 - x = \frac{1}{12} (L_2(4x - 1) - L_0(4x - 1)).$$

D'où

$$\int_0^x \Psi_{11n'0}(\tau, y) d\tau = \begin{cases} 2 \times 2 \sqrt{\frac{3}{2}} \sqrt{\frac{1}{2}} [\frac{1}{12} (L_2(4x - 1) - L_0(4x - 1))] L_0(4y - 2n' + 1), & \text{si } x < \frac{1}{2} \\ 0 & \text{si } \frac{1}{2} \leq x \end{cases}.$$

En revenant à la définition des ondelettes, on peut exprimer cette dernière égalité comme suit

$$\int_0^x \Psi_{11n'0}(\tau, y) d\tau = \frac{1}{4} \left[\frac{\sqrt{3}}{3\sqrt{5}} \Psi_{12n'0}(x, y) - \frac{\sqrt{3}}{3} \Psi_{10n'0}(x, y) \right].$$

Pour $n' = 1$, $\int_0^x \Psi_{1110}(\tau, y) d\tau$ représente l'intégrale de la septième composante et donc

elle fournit la septième ligne de la matrice P_x . C'est à dire

$$\begin{aligned} & \int_0^x \Psi_{1110}(\tau, y) d\tau \\ &= \frac{1}{4} \left[\frac{\sqrt{3}}{3\sqrt{5}} \Psi_{1210}(x, y) - \frac{\sqrt{3}}{3} \Psi_{1010}(x, y) \right] \\ &= \frac{1}{2^2} \left[-\frac{1}{\sqrt{3}}, \underbrace{0, \dots, 0}_{11 \text{ zéros}}, \frac{1}{\sqrt{3}\sqrt{5}}, \underbrace{0, \dots, 0}_{23 \text{ zéros}} \right] \Psi(x, y). \end{aligned}$$

Donc la septième ligne de la matrice P_x est

$$\frac{1}{2^2} \left[-\frac{1}{\sqrt{3}}, \underbrace{0, \dots, 0}_{11 \text{ zéros}}, \frac{1}{\sqrt{3}\sqrt{5}}, \underbrace{0, \dots, 0}_{23 \text{ zéros}} \right].$$

3.2 Matrice opérationnelle d'intégration par rapport à y

Théorème 2.3 La matrices opérationnelle d'intégration P_y [6] est entièrement déterminée par la formule :

$$\int_0^y \Psi(x, \tau) d\tau = P_y \Psi(x, y). \quad (2.10)$$

La matrice opérationnelle d'intégration P_y qui est d'ordre $2^{k-1}2^{k'-1}MM'$ est donnée par :

$$P_y = \frac{1}{2^{k'}} \begin{bmatrix} P & O & O & \dots & O \\ O & P & O & \dots & O \\ O & O & P & \dots & O \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ O & O & O & \dots & P \end{bmatrix},$$

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 sont données par

$$P = \begin{bmatrix} L & F & F & \dots & F \\ 0 & L & F & \dots & F \\ 0 & 0 & L & \dots & F \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & L \end{bmatrix}$$

F , L et 0 sont des matrices de taille $M' \times M'$ avec 0 la matrice nulle, L est la même matrice donnée par (2.8) et F est définie par :

$$F = \begin{bmatrix} 2 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}.$$

Proposition 2.4 Les matrices opérationnelles d'intégration P_x et P_y , définie par les sous matrices précédentes, sont inversibles.

4 La formule d'intégration des ondelettes de Legendre

La formule d'intégration des ondelettes de Legendre qu'on veut établir dans cette section s'exprime en fonction du vecteur d'ondelettes $\Psi(x, y)$ donnée par (2.9) et des matrices opérationnelles d'intégration P_x et P_y . On dispose donc de tous les outils dont on a besoin. Cette formule, notée LWF (Formula Wavelets Legendre), est énoncée dans le théorème suivant :

Théorème 2.4 Soit $f(x, y)$ une fonction de $L^2(]a, A[\times]b, B[)$ où a, A, b, B sont des réels donnés. On a alors l'approximation

$$\int_a^A \int_b^B f(x, y) dx dy \simeq (A - a)(B - b) C^T P_x P_y \Psi(1, 1), \quad (2.11)$$

où le vecteur C est le vecteur de décomposition suivant la base d'ondelettes de la fonction :

$$(x, y) \longmapsto f((A - a)x + a, (B - b)y + b).$$

Preuve. On considère le changement de variables qui transforme le pavé $]a, A[\times]b, B[$ en carré unité $I =]0, 1[\times]0, 1[$ et qui est donné par

$$\begin{aligned} X &= (A - a)x + a \\ Y &= (B - b)y + b, \end{aligned}$$

de tel sorte que

$$\int_a^A \int_b^B f(x, y) dx dy = (A - a)(B - b) \int_0^1 \int_0^1 f((A - a)x + a, (B - b)y + b) dx dy.$$

Comme la fonction $(x, y) \longmapsto f(x, y) \in L^2(]a, A[\times]b, B[)$, alors la fonction $(x, y) \longmapsto f((A - a)x + a, (B - b)y + b) \in L^2(I)$.

D'après la formule (2.4), on peut avoir l'approximation :

$$\begin{aligned} f((A - a)x + a, (B - b)y + b) &\simeq \sum_{n=1}^{2^{k-1}M-1} \sum_{m=0}^{2^{k'-1}M'-1} C_{nmn'm'} \Psi_{nmn'm'}(x, y) \\ &\simeq C^T \Psi(x, y), \end{aligned} \quad (2.12)$$

où

$$C_{nmn'm'} = \int_0^1 \int_0^1 f((A - a)x + a, (B - b)y + b) \Psi_{nmn'm'}(x, y) dx dy.$$

L'intégrale $\int_a^A \int_b^B f(x, y) dx dy$ est alors approchée par

$$\int_a^A \int_b^B f(x, y) dx dy \simeq (A - a)(B - b) \int_0^1 \int_0^1 C^T \Psi(x, y) dx dy.$$

En appliquant le théorème de Fubini, on a

$$\int_a^A \int_b^B f(x, y) dx dy \simeq (A - a)(B - b) C^T \int_0^1 \left(\int_0^1 \Psi(x, y) dx \right) dy.$$

En tenant compte de la relation (2.7) avec $x = 1$, on arrive à

$$\int_a^A \int_b^B f(x, y) dx dy \simeq (A - a)(B - b) C^T \int_0^1 P_x \Psi(1, y) dy.$$

Finalement, la relation (2.10) pour $y = 1$, nous conduit à la formule d'intégration des ondelettes de Legendre

$$\int_a^A \int_b^B f(x, y) dx dy \simeq (A - a)(B - b) C^T P_x P_y \Psi(1, 1).$$

■

Remarque 2.2 Comme les deux matrices d'intégrations P_x, P_y commutent c'est à dire $P_x P_y = P_y P_x$, alors cette formule d'intégration peut aussi s'écrire

$$\int_a^A \int_b^B f(x, y) dx dy \simeq (A - a)(B - b) C^T P_y P_x \Psi(1, 1).$$

Chapitre 3

Les tests numériques

Pour valider la nouvelle formule d'intégration LWF, on va la tester numériquement et on va la comparer avec les deux méthodes d'intégration des trapèzes composite et de Simpson composite et ceci sur plusieurs exemples. Les codes permettant cette étude ont été réalisés sur le logiciel Matlab

On va considérer différentes valeurs de M, M' (degré du polynôme de Legendre) et différentes valeurs de k, k' (niveau de décomposition du carré unité).

Le vecteur des coefficients C apparaissant dans la formule d'intégration LWF (2.11), est obtenu de manière approximative et ceci en discrétisant la relation (2.12). Il est donc solution du système linéaire d'ordre $2^{k-1}2^{k'-1}MM'$

$$f((A-a)x_i + a, (B-b)y_j + b) \simeq C^T \Psi(x_i, y_j), \quad (3.1)$$

$$\text{où } 1 \leq i \leq 2^{k-1}M \text{ et } 1 \leq j \leq 2^{k'-1}M'. \quad (3.2)$$

Les points (x_i, y_j) de cette discrétisation sont donnés dans [7] par

$$x_i = \frac{i-0.5}{2^{k-1}M}, y_j = \frac{j-0.5}{2^{k'-1}M'}; \leq i \leq 2^{k-1}M \text{ et } 1 \leq j \leq 2^{k'-1}M'. \quad (3.3)$$

L'erreur d'intégration Er est définie par : $Er = |I_{ex} - I_{app}|$,

où

I_{ex} : représente l'intégrale exacte .

I_{app} : représente l'approximation de l'intégrale obtenue par l'une des trois formules.

On note

E_{LWF} : l'erreur associée à la formule des ondelettes de Legendre.

E_{trap} : l'erreur associée à la formule des trapèzes composite.

E_{Simp} : l'erreur associée à la formule de Simpson composite.

1 Exemple 1

On prend la fonction $f(x, y) = x^4 + y^3$ sur l'intervalle $[0, 1] \times [0, 1]$.

Pour $k = k' = 2$:

1. EXEMPLE 1

$M = M'$	N	E_{LWF}	E_{trap}	E_{Simp}
2	16	$8.9730e-02$	$3.6328e-02$	$5.2083e-04$
4	64	$9.9763e-04$	$9.1064e-03$	$3.2552e-05$
6	144	0	$4.0493e-03$	$6.4300e-06$
8	256	0	$2.2781e-03$	$2.0345e-06$
10	400	0	$1.4581e-03$	$8.3333e-07$

TABLEAU 3.1 – Le tableau des erreurs pour $f(x, y) = x^4 + y^3$ et $k = k' = 2$:

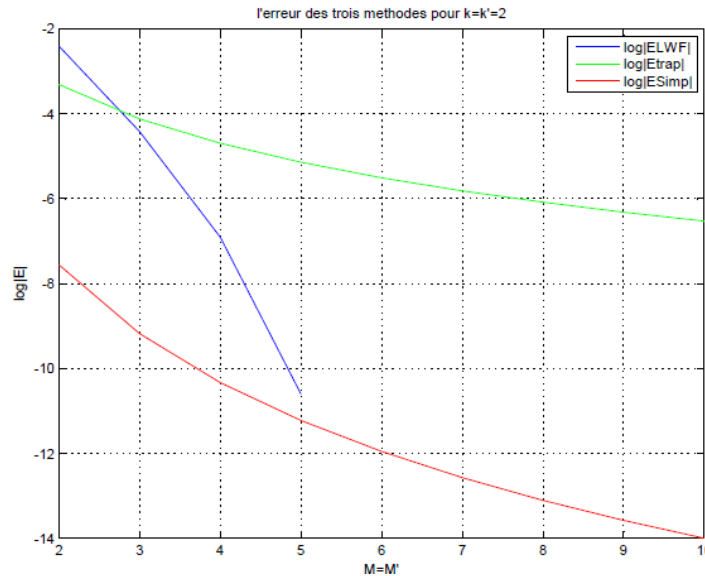


FIGURE 3.1 – Logarithme de l'erreur absolue pour $f(x, y) = x^4 + y^3$

Pour $k = k' = 3$:

$M = M'$	N	E_{LWF}	E_{trap}	E_{Simp}
2	64	$3.0561e-02$	$9.1064e-03$	$3.2552e-05$
4	256	$6.9328e-05$	$2.2781e-03$	$2.0345e-06$
6	576	0	$1.0126e-03$	$4.0188e-07$
8	1024	0	$5.6963e-04$	$1.2716e-07$
10	1600	0	$3.6457e-04$	$5.2083e-08$

TABLEAU 3.2 – Le tableau des erreurs pour $f(x, y) = x^4 + y^3$ et $k = k' = 3$:

D'après les tableaux précédents, on remarque que toutes les méthodes convergent numériquement car l'erreur diminue quand le nombre des niveaux (k, k') et le degré des polynômes de Legendre (M, M') augmentent, (c'est à dire quand le nombre des points $N = 2^{k-1}2^{k'-1}MM'$ augmente). cette décroissance de l'erreur est visible dans le graphe [2].

On remarque que la méthode LWF est sensible au degré du polynôme de Legendre. En effet, pour un nombre de points N correspondant à des degrés M, M' strictement inférieur à 6, la méthode de Simpson donne de meilleurs résultats mais quand les degrés M, M' sont supérieur ou égale à 6, c'est la méthode LWF qui devient plus précise. Elle est exacte pour cet exemple (polynôme de degré inférieur ou égal à 4) alors que l'erreur par la méthode de Simpson reste inférieur à 10^{-8} .

2 Exemple 2

On prend la fonction $f(x, y) = x^2 e^y$ sur l'intervalle $[-1, 1] \times [0, 1]$.
Pour $k = k' = 2$:

$M = M'$	N	E_{LWF}	E_{trap}	E_{Simp}
2	16	$2.4116e-01$	$1.4990e-01$	$2.4676e-05$
4	64	$5.7575e-05$	$3.7335e-02$	$1.5508e-06$
6	144	$3.7127e-08$	$1.6582e-02$	$3.0665e-07$
8	256	$1.2454e-11$	$9.3252e-03$	$9.7062e-08$
10	400	$3.1086e-15$	$5.9674e-03$	$3.9763e-08$

TABLEAU 3.3 – Le tableau des erreurs pour $f(x, y) = x^2 e^y$ et $k = k' = 2$:

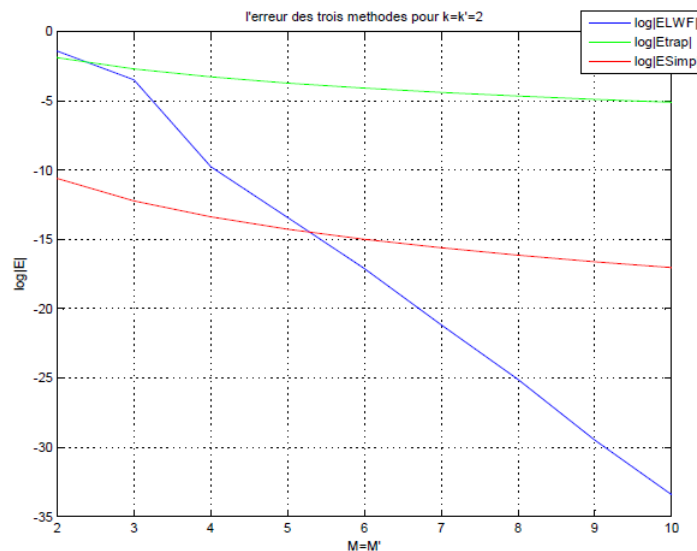


FIGURE 3.2 – Logarithme de l'erreur absolue pour $f(x, y) = x^2 e^y$

Pour $k = k' = 3$:

$M = M'$	N	E_{LWF}	E_{trap}	E_{Simp}
2	64	$8.0108e-02$	$3.7335e-02$	$1.5508e-06$
4	256	$4.0323e-06$	$9.3252e-03$	$9.7062e-08$
6	576	$6.4891e-10$	$4.1438e-03$	$1.9178e-08$
8	1024	$5.3291e-14$	$2.3307e-03$	$6.0685e-09$
10	1600	$6.6613e-16$	$1.4916e-03$	$2.4858e-09$

TABLEAU 3.4 – Le tableau des erreurs pour $f(x, y) = x^2 e^y$ et $k = k' = 3$:

Pour cet exemple d'une fonction polynômiale de degré infinie, on peut faire les mêmes remarques que pour le précédent exemple. La décroissance de l'erreur est visible dans le graphe.

La méthode de Simpson donne de meilleurs résultats pour un nombre de points N correspondant à des degrés M, M' strictement inférieur à 6, mais quand les degrés M, M'

3. EXEMPLE 3

sont supérieurs ou égaux à 6, c'est la méthode LWF qui devient plus précise. Elle atteint des erreurs de l'ordre 10^{-16} alors que l'erreur par la méthode de Simpson reste inférieure à 10^{-9} .

3 Exemple 3

On prend la fonction $f(x, y) = \frac{x^2}{1+y^2}$ sur l'intervalle $[0, 1] \times [0, 2]$.

Pour $k = k' = 2$:

$M = M'$	N	E_{LWF}	E_{trap}	E_{Simp}
2	16	$3.2749e-02$	$1.0398e-02$	$6.7350e-04$
4	64	$1.9809e-04$	$2.6037e-03$	$2.8645e-06$
6	144	$1.8651e-05$	$1.1576e-03$	$3.2984e-07$
8	256	$1.1605e-06$	$6.5125e-04$	$1.0388e-07$
10	400	$6.5143e-10$	$4.1682e-04$	$4.2591e-08$

TABLEAU 3.5 – Le tableau des erreurs pour $f(x, y) = \frac{x^2}{1+y^2}$ et $k = k' = 2$:

Pour $k = k' = 3$:

$M = M'$	N	E_{LWF}	E_{trap}	E_{Simp}
2	64	$3.2749e-02$	$1.0398e-02$	$6.7350e-04$
4	256	$1.9809e-04$	$2.6037e-03$	$2.8645e-06$
6	576	$1.8651e-05$	$1.1576e-03$	$3.2984e-07$
8	1024	$1.1605e-06$	$6.5125e-04$	$1.0388e-07$
10	1600	$6.5143e-10$	$4.1682e-04$	$4.2591e-08$

TABLEAU 3.6 – Le tableau des erreurs pour $f(x, y) = \frac{x^2}{1+y^2}$ et $k = k' = 3$:

Dans cet exemple, d'une fonction rationnelle aussi on remarque la décroissance de l'erreur qui est visible dans le graphe.

On a les mêmes remarques que pour les exemples précédents, la seule différence réside dans la vitesse de convergence de la méthode LWF tel que jusqu'à des degrés M, M' strictement inférieur à 8, la méthode de Simpson donne de meilleurs résultats mais quand les degrés M, M' sont supérieur ou égale à 10, c'est la méthode LWF qui devient plus précise.

3. EXEMPLE 3

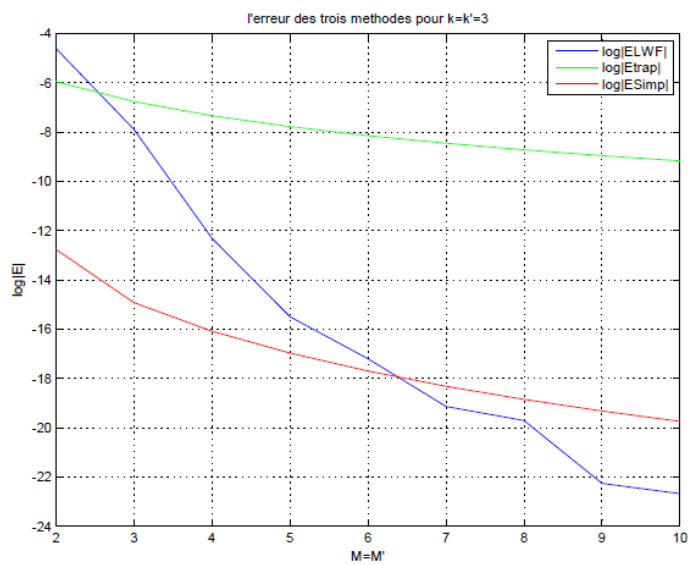


FIGURE 3.3 – Logarithme de l'erreur absolue pour $f(x, y) = \frac{x^2}{1+y^2}$

Conclusion et Perspectives

Dans ce travail, on a proposé une nouvelle méthode pour estimer les intégrales doubles. Cette méthode est basée sur la décomposition, de la fonction à intégrer, en ondelettes de Legendre en deux dimensions. Les tests réalisés, montrent que cette méthode est plus efficace par rapport aux autres méthodes de cubature telles que la méthode des trapèzes et celle de Simpson composites. L'approximation est de plus en plus précise que l'on augmente l'ordre du polynôme de Legendre.

On rappelle que la formule d'intégration numérique LWF s'exprime en fonction du vecteur C et en fonction des matrices opérationnelles P_x et P_y .

Le vecteur des coefficients C apparaissant dans la formule d'intégration LWF (2.11), est obtenu de manière approximative comme étant la solution d'un système linéaire d'ordre $2^{k-1}2^{k'-1}MM'$ et ceci après discrétisation de la relation (2.12).

Les matrices opérationnelles P_x et P_y sont elles aussi calculées de manière approximative voir [6].

Comme perspective pour améliorer la précision des résultats de la méthode LWF, on propose de

1. Calculer les matrices opérationnelles P_x et P_y de manière exacte (ceci est possible grâce aux propriétés des polynômes de Legendre).
2. Améliorer le calcul du vecteur C et ceci en trouvant un autre système de points différent de celui donné par [7].

Comme autre perspective, on espère pouvoir trouver une estimation de l'erreur d'intégration de la formule LWF qui n'est que l'intégrale de l'erreur de l'approximation (2.3). Il est donc très intéressant de se pencher sur ce domaine.

Bibliographie

- [1] B. Démidovith, I. Maron : Eléments de calcul numérique, 677-679. [9](#), [15](#), [16](#), [17](#)
- [2] BENKAHLA Fatiha et CHABOUNI Acia, Résolution des EDPs par la Méthode des Ondelettes de Legendre en 2D. MEMOIRE, 2016. [9](#), [28](#)
- [3] Chapitre 5-méthode d'intégration numérique, Université Paul Sabatier (2014-2015). 33-35. 36-43. [8](#), [10](#), [11](#), [12](#), [14](#), [15](#)
- [4] C. Portenier, Analyse fonctionnelle; Fachbereich Mathematik und Informatik, Philipps-Universität Marburg (2005). 21 [18](#)
- [5] F. Yin, J. Song, Solving linear PDEs with the Aid of Two-Dimensional Legendre Wavelets, National Conference on Information Technology and Computer Science (CITCS 2012) [7](#)
- [6] H. Parsian, two dimension Legendre wavelets and operational matrices of integration, Acta Mathematica Academiae Paedagogicae 21 (2005) 101-106. [33](#), [38](#), [44](#), [70](#) [7](#), [19](#), [20](#), [24](#), [32](#)
- [7] I. Aziz, S. Ul-Islam and B. Sarler, Wavelets collocation method for the numerical solution of elliptic BV problems, Applied Mathematical Modelling, 37 (2013), 676-694. [27](#), [32](#)
- [8] L. Guillopé, Analyse fonctionnelle. Approximations Hilbertiennes et développements en série. Ecole polytechnique de l'université de Nantes (2008). 21, 22 [18](#)
- [9] M. Abramowitz and I.A. Stegun, Handbook of Mathematical Functions With Formulas, Graphs and Mathematical Tables; National Bureau of Standards Applied Mathematics, Serie 55, (1972). 21 [18](#)
- [10] M. Razzaghi, S. Yousefi : Legendre wavelets direct method for variational problems. Mathematics and Computers in Simulation 53, 2000, 185-192.
- [11] Naima ABLAOUI-LAHMAR, Sur l'analyse multi-résolution en ondelettes bidimensionnelles et application pour la simulation des problèmes biologiques. THÈSE, 2016.

Résumé :

Dans ce mémoire, on propose une nouvelle méthode pour estimer les intégrales doubles. Cette méthode est basée sur la décomposition, de la fonction à intégrer, en ondelettes de Legendre en deux dimensions.

Pour introduire cette nouvelle formule, on a rappelé certains outils mathématiques ; les polynômes de Legendre, les ondelettes de Legendre en deux dimensions, l'approximation et la décomposition dans certains espace de Hilbert $L^2([0, 1[\times]0, 1[)$, les matrices opérationnelles d'intégration P_x et P_y .

Numériquement, cette méthode a été testée et comparée avec d'autres méthodes.

Les tests numériques réalisés, montrent que cette méthode est plus efficace par rapport aux autres méthodes de cubature telles que la méthode des trapèzes et celle de Simpson composites.

Mots-Clés : Intégration numérique, les polynômes de Legendre, les Ondelettes de Legendre bidimensionnelles, les matrices opérationnelles d'intégration P_x et P_y .

Abstract :

In this thesis, we propose a new method to estimate double integrals. This method is based on the decomposition of the function to be integrated into Legendre wavelets in two dimensions.

To introduce this new formula, we recalled certain mathematical tools ; Legendre polynomials, two-dimensional Legendre wavelets, approximation and decomposition in certain Hilbert spaces $L^2([0, 1[\times]0, 1[)$, operational matrices d'intégration P_x and P_y .

Numerically, this method has been tested and compared with other methods.

The numerical tests carried out, show that this method is more efficient compared to other cubature methods such as the trapezoidal method and that of composite Simpson.

Key words : Numerical integration, Legendre polynomials, two-dimensional Legendre wavelets, operational matrices d'intégration P_x and P_y .

ملخص

في هذه المذكرة، نقترح طريقة جديدة لتقدير التكاملات المزدوجة. تعتمد هذه الطريقة على تحلل الوظيفة المراد دمجها في موجات Legendre ذات البعدين.

لتقديم هذه الصيغة الجديدة، استدعينا بعض الأدوات الرياضية، متعدد الحدود Legendre، موجات Legendre ثنائية الأبعاد، التقريب والتحلل في بعض فضاءات هيلبرت $L^2([0,1] \times [0,1])$ ، مصفوفات التكامل التشغيلي P_x و P_y .

عددياً، تم اختيار هذه الطريقة ومقارنتها بالطرق الأخرى.

أظهرت الاختبارات العددية التي تم إجراؤها أن هذه الطريقة أكثر كفاءة مقارنة بطرق التكعيب الأخرى مثل طريقة شبه المنحرف وطريقة سيمبسون المركبة.

Annexe

Dans cette partie, on présente les différents codes utilisés.

4 Les sous programmes

4.1 Polynôme de Legendre px et py

```
function [px] = polyx_legendre(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] = polyy_legendre(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
```

4.2 L'ondelette de Legendre

```
function [psi] = OndLeg(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
```

4.3 Le vecteur d'ondelette de Legendre

```
function [psiv] = VectOndLeg(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 = polyx_legendre(2^(kx)*x-2*nx+1,Mx);
            py = polyy_legendre(2^(ky)*y-2*ny+1,My);
            for my = 1 :My
                i = i+1;
                psiv(i) = OndLeg(x,y,kx,ky,nx,ny,mx,my,px(mx),py(my));
            end
        end
    end
end
end
end
```

4.4 La matrice d'ondelette de Legendre

```
function [ psim ] = MatOndLeg(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
```

4.5 Les matrices d'intégrations Px; Py

```
function [Px] = matrice_integrationsx(Mx,My,kx,ky)
%D la matrice identité
D=eye(2^(ky-1)*My);
L = tri_diag_matricx(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_integrationsy(My,ky,Mx,kx)
L = tri_diag_matricy(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
May = eye(2^(kx-1)*My);
Py = (1/2^ky) * kron(May,repeat_matrixy(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
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_matrixy(F,L,kx,ky,Mx,My));
end
end

```

4.5.1 Les matrices Lx et Ly

```

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
```

4.5.2 les sous programmes repeatMtrix 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
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
```

4.6 Les sous programmes des données, la solution exacte, le vecteur second membre, trapèze et Simpson :

4.6.1 Les données

```
function [ x,y ] = donnees( kx,ky,Mx,My )
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;
end
```

4.6.2 La solution exacte

```
function [ fff ] = solex( f,a,A,b,B )
syms x y
J=int(f(x,y));
g(x)=(J);
h(y)=g(A)-g(a);
k=int(h(y));
r(y)=(k);
fff=r(B)-r(b);
end
```

4.6.3 Le vecteur second membre

```
function [ ff ] = vectff( a,b,A,B,x,y,f,kx,ky,Mx,My )
ff=zeros(1,2^(kx-1)*2^(ky-1)*Mx*My);
j=1;
for ii=1 :My*2^(ky-1)
    for i=1 :Mx*2^(kx-1)
```

```
    ff(j)=f((A-a)*x(i)+a,(B-b)*y(ii)+b);
    j=j+1;
end
end
end
```

4.6.4 Trapèze

```
function [ II ] = trapeze(a,A,b,B,kx,ky,Mx,My,f)
hx=(A-a)/(2^(kx-1)*Mx);
hy=(B-b)/(2^(ky-1)*My);
for i=1 :2^(kx-1)*Mx-1
    x(i)=a+(i)*hx;
end
for ii=1 :2^(ky-1)*My-1
    y(ii)=b+(ii)*hy;
end
S=0;
S1=0;
S2=0;
S3=0;
S4=0;
for i=1 :My*2^(ky-1)-1
    S=S+f(a,y(i));
    S1=S1+f(A,y(i));
    for ii=1 :Mx*2^(kx-1)-1
        S4=S4+f(x(ii),y(i));
    end
end
for ii=1 :Mx*2^(kx-1)-1
    S2=S2+f(x(ii),b);
    S3=S3+f(x(ii),B);
end
II=(hx*hy/4)*(f(a,b)+f(a,B)+2*S+f(A,b)+f(A,B)+2*S1+2*(S2+S3+2*S4));
end
```

4.6.5 Simpson

```
function [ I ] = simpson(a,A,b,B,kx,ky,Mx,My,f)
hx=(A-a)/(2^(kx-1)*Mx);
hy=(B-b)/(2^(ky-1)*My);
for i=1 :2^(kx-1)*Mx-1
    x(i)=a+(i)*hx;
end
for ii=1 :2^(ky-1)*My-1
    y(ii)=b+(ii)*hy;
```

```
end
S1=0;
S2=0;
S3=0;
S4=0;
S5=0;
S6=0;
for i=1 :(My*2^(ky-1))/2-1
    S1=S1+f(a,y(2*i))+f(A,y(2*i));
end
for i=1 :(My*2^(ky-1))/2
    S2=S2+f(a,y(2*i-1))+f(A,y(2*i-1));
end
for i=1 :(Mx*2^(kx-1))/2-1
    S3=S3+f(x(2*i),b)+f(x(2*i),B);
end
for i=1 :(Mx*2^(kx-1))/2
    S4=S4+f(x(2*i-1),b)+f(x(2*i-1),B);
end
for i=1 :(Mx*2^(kx-1))/2-1
    for j=1 :(My*2^(ky-1))/2-1
        S5=S5+f(x(2*i),y(2*j))+2*f(x(2*i),y(2*j-1));
    end
    S5=S5+2*f(x(2*i),y(My*2^(ky-1)-1));
end
for i=1 :(Mx*2^(kx-1))/2
    for j=1 :(My*2^(ky-1))/2-1
        S6=S6+f(x(2*i-1),y(2*j))+2*f(x(2*i-1),y(2*j-1));
    end
    S6=S6+2*f(x(2*i-1),y(My*2^(ky-1)-1));
end
I=(hx*hy/9)*(f(a,b)+f(a,B)+f(A,b)+f(A,B)+2*S1+4*S2+2*S3+4*S4+4*S5+8*S6);
end
```

5 Le programme principale

```
clear all
close all
clc
%%%%%% TestLWF.m
format long
%%le pavé d'intégration
a=-1;
A=1;
b=0;
B=1;
f=@(x,y)(x^2*exp(y));%%La fonction qui est intégrer
```

```

[ fff ] = solex( f,a,A,b,B );%%L'intégrale exacte
kx=2;%%le nombre des niveaux par rapport x
ky=2;%%le nombre des niveaux par rapport y
Mx=2;%%le degré des polynômes de Legendre px
My=2;%%le degré des polynômes de Legendre py
[ x,y ] = donnees( kx,ky,Mx,My );%%La subdivision du pavé d'intégration [-1,1]*[0,1]
[ psim ] = MatOndLeg(x,y,Mx,My,kx,ky);%%La matrice psi
[ ff ] = vectff( a,b,A,B,x,y,f,kx,ky,Mx,My );%%Le vecteur second membre
C=(pinv((psim)'))*((ff)');%%Le vecteur des coefficients
[Px] = matrice_integracion(Mx,My,kx,ky);%%La matrice opérationnelle Px
[Py] = matrice_integraciony(My,ky,Mx,kx);%%La matrice opérationnelle Py
[ psiv1 ] = VectOndLeg(1,1,Mx,My,kx,ky);%%Le vecteur psi(1,1)
fff=(A-a)*(B-b)*((C)')*Px*Py*((psiv1)');%%L'intégrale approche de la formule LWF
%%L' erreur absolu de la formule LWF
Eo2=abs(fff-ffff)
[ II ] = trapeze(a,A,b,B,kx,ky,Mx,My,f );%%L'intégrale approche de la formule de tra-
pèze composite
%%L' erreur absolu de la formule de trapèze composite
Et2=abs(fff-II)
[ I ] = simpson(a,A,b,B,kx,ky,Mx,My,f );%%L'intégrale approche de la formule de tra-
pèze composite
%%L' erreur absolu de la formule de Simpson composite
Es2=abs(fff-I)
Mx=3;
My=3;
[ x,y ] = donnees( kx,ky,Mx,My );
[ psim ] = MatOndLeg(x,y,Mx,My,kx,ky);
[ ff ] = vectff( a,b,A,B,x,y,f,kx,ky,Mx,My );
C=(pinv((psim)'))*((ff)');
[Px] = matrice_integracion(Mx,My,kx,ky);
[Py] = matrice_integraciony(My,ky,Mx,kx);
[ psiv1 ] = VectOndLeg(1,1,Mx,My,kx,ky);
ffff=(A-a)*(B-b)*((C)')*Px*Py*((psiv1)');
Eo3=abs(fff-ffff);
[ II ] = trapeze(a,A,b,B,kx,ky,Mx,My,f );
Et3=abs(fff-II);
[ I ] = simpson(a,A,b,B,kx,ky,Mx,My,f );
Es3=abs(fff-I);
Mx=4;
My=4;
[ x,y ] = donnees( kx,ky,Mx,My );
[ psim ] = MatOndLeg(x,y,Mx,My,kx,ky);
[ ff ] = vectff( a,b,A,B,x,y,f,kx,ky,Mx,My );
C=(pinv((psim)'))*((ff)');
[Px] = matrice_integracion(Mx,My,kx,ky);
[Py] = matrice_integraciony(My,ky,Mx,kx);
[ psiv1 ] = VectOndLeg(1,1,Mx,My,kx,ky);
ffff=(A-a)*(B-b)*((C)')*Px*Py*((psiv1)');
Eo4=abs(fff-ffff)

```

```

[ II ] = trapeze(a,A,b,B,kx,ky,Mx,My,f) ;
Et4=abs(fff-II)
[ I ] = simpson(a,A,b,B,kx,ky,Mx,My,f) ;
Es4=abs(fff-I)
Mx=5;
My=5;
[ x,y ] = donnees( kx,ky,Mx,My ) ;
[ psim ] = MatOndLeg(x,y,Mx,My,kx,ky) ;
[ ff ] = vectff( a,b,A,B,x,y,f,kx,ky,Mx,My ) ;
C=(pinv((psim)'))*((ff)') ;
[Px] = matrice_integracion(Mx,My,kx,ky) ;
[Py] = matrice_integraciony(My,ky,Mx,kx) ;
[ psiv1 ] = VectOndLeg(1,1,Mx,My,kx,ky) ;
ffff=(A-a)*(B-b)*((C)')*Px*Py*((psiv1)') ;
Eo5=abs(fff-ffff) ;
[ II ] = trapeze(a,A,b,B,kx,ky,Mx,My,f) ;
Et5=abs(fff-II) ;
[ I ] = simpson(a,A,b,B,kx,ky,Mx,My,f) ;
Es5=abs(fff-I) ;
Mx=6;
My=6;
[ x,y ] = donnees( kx,ky,Mx,My ) ;
[ psim ] = MatOndLeg(x,y,Mx,My,kx,ky) ;
[ ff ] = vectff( a,b,A,B,x,y,f,kx,ky,Mx,My ) ;
C=(pinv((psim)'))*((ff)') ;
[Px] = matrice_integracion(Mx,My,kx,ky) ;
[Py] = matrice_integraciony(My,ky,Mx,kx) ;
[ psiv1 ] = VectOndLeg(1,1,Mx,My,kx,ky) ;
ffff=(A-a)*(B-b)*((C)')*Px*Py*((psiv1)') ;
Eo6=abs(fff-ffff) ;
[ II ] = trapeze(a,A,b,B,kx,ky,Mx,My,f) ;
Et6=abs(fff-II) ;
[ I ] = simpson(a,A,b,B,kx,ky,Mx,My,f) ;
Es6=abs(fff-I) ;
Mx=7;
My=7;
[ x,y ] = donnees( kx,ky,Mx,My ) ;
[ psim ] = MatOndLeg(x,y,Mx,My,kx,ky) ;
[ ff ] = vectff( a,b,A,B,x,y,f,kx,ky,Mx,My ) ;
C=(pinv((psim)'))*((ff)') ;
[Px] = matrice_integracion(Mx,My,kx,ky) ;
[Py] = matrice_integraciony(My,ky,Mx,kx) ;
[ psiv1 ] = VectOndLeg(1,1,Mx,My,kx,ky) ;
ffff=(A-a)*(B-b)*((C)')*Px*Py*((psiv1)') ;
Eo7=abs(fff-ffff) ;
[ II ] = trapeze(a,A,b,B,kx,ky,Mx,My,f) ;
Et7=abs(fff-II) ;
[ I ] = simpson(a,A,b,B,kx,ky,Mx,My,f) ;
Es7=abs(fff-I) ;

```

```
Mx=8;
My=8;
[ x,y ] = donnees( kx,ky,Mx,My );
[ psim ] = MatOndLeg(x,y,Mx,My,kx,ky);
[ ff ] = vectff( a,b,A,B,x,y,f,kx,ky,Mx,My );
C=(pinv((psim)'))*((ff)');
[Px] = matrice_integracion(Mx,My,kx,ky);
[Py] = matrice_integraciony(My,ky,Mx,kx);
[ psiv1 ] = VectOndLeg(1,1,Mx,My,kx,ky);
ffff=(A-a)*(B-b)*((C)')*Px*Py*((psiv1)');
Eo8=abs(fff-ffff)
[ II ] = trapeze(a,A,b,B,kx,ky,Mx,My,f);
Et8=abs(fff-II)
[ I ] = simpson(a,A,b,B,kx,ky,Mx,My,f);
Es8=abs(fff-I)
Mx=9;
My=9;
[ x,y ] = donnees( kx,ky,Mx,My );
[ psim ] = MatOndLeg(x,y,Mx,My,kx,ky);
[ ff ] = vectff( a,b,A,B,x,y,f,kx,ky,Mx,My );
C=(pinv((psim)'))*((ff)');
[Px] = matrice_integracion(Mx,My,kx,ky);
[Py] = matrice_integraciony(My,ky,Mx,kx);
[ psiv1 ] = VectOndLeg(1,1,Mx,My,kx,ky);
ffff=(A-a)*(B-b)*((C)')*Px*Py*((psiv1)');
Eo9=abs(fff-ffff);
[ II ] = trapeze(a,A,b,B,kx,ky,Mx,My,f);
Et9=abs(fff-II);
[ I ] = simpson(a,A,b,B,kx,ky,Mx,My,f);
Es9=abs(fff-I);
Mx=10;
My=10;
[ x,y ] = donnees( kx,ky,Mx,My );
[ psim ] = MatOndLeg(x,y,Mx,My,kx,ky);
[ ff ] = vectff( a,b,A,B,x,y,f,kx,ky,Mx,My );
C=(pinv((psim)'))*((ff)');
[Px] = matrice_integracion(Mx,My,kx,ky);
[Py] = matrice_integraciony(My,ky,Mx,kx);
[ psiv1 ] = VectOndLeg(1,1,Mx,My,kx,ky);
ffff=(A-a)*(B-b)*((C)')*Px*Py*((psiv1)');
Eo10=abs(fff-ffff)
[ II ] = trapeze(a,A,b,B,kx,ky,Mx,My,f);
Et10=abs(fff-II)
[ I ] = simpson(a,A,b,B,kx,ky,Mx,My,f);
Es10=abs(fff-I)
M=[2,3,4,5,6,7,8,9,10];
E=[log(Eo2),log(Eo3),log(Eo4),log(Eo5),log(Eo6),log(Eo7),log(Eo8),log(Eo9),log(Eo10)];
e=[log(Et2),log(Et3),log(Et4),log(Et5),log(Et6),log(Et7),log(Et8),log(Et9),log(Et10)];
ee=[log(Es2),log(Es3),log(Es4),log(Es5),log(Es6),log(Es7),log(Es8),log(Es9),log(Es10)];
```

```
plot(M,E,M,e,'g',M,ee,'r')
grid
xlabel('M=M''')
ylabel('log|E|')
legend('log|ELWF|','log|Etrap|','log|ESimp|')
title('l''erreur des trois methodes pour k=k''=2')
```
