

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 INFORMATIQUE
FILIÈRE : MATHÉMATIQUES



MÉMOIRE DE FIN D'ÉTUDES

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

Université de Mostaganem

Spécialité "Modélisation, Contrôle et Optimisation"

présentée par :

Khadra TLEMSANI

Intégration numérique et approximation au sens des moindres carrés

soutenu publiquement le 29 Juin 2022 devant le jury composé de :

Présidente :	Djemaia BENSIKADDOUR	MCB	UMAB
Examinatrice :	Maghnia HAMOU MAAMAR	MCA	UMAB
Encadreur :	Naima ABLAOUI-LAHMAR	MCB	UMAB

Année Universitaire : 2021 / 2022

M
A
S
T
E
R

Table des matières

Table des figures	2
Liste des tables	3
dédicace	i
Remerciements	ii
Résumé	iii
Introduction générale	1
1 L'intégration numérique	3
1.1 Introduction	3
1.2 Les formules de Newton-Côte simples	4
1.3 Les formules de Newton-Côtes composites	8
1.4 Les formules de cubature	11
2 Formules d'intégration basées sur l'approximation	12
2.1 Introduction	12
2.2 Rappels et définitions	12
2.3 Formules de quadrature basées sur l'approximation	14
2.3.1 Nouvelle formule de quadrature simple	14
2.3.2 Les formules de quadrature composites	17
2.4 Les formules de cubature basées sur l'approximation	18
2.4.1 Nouvelle formule cubique simple	18
2.4.2 Formules de cubature composites	21
3 Tests et résultats numériques	24
3.1 Introduction	24
3.2 Les fonctions test de Genz	24
3.3 Les tests en une dimension	25

3.4 Les tests en deux dimensions	26
Conclusion et perspective	34
Bibliographie	35
Annexe	36

Table des figures

1.1	Méthode de trapèze $n=1$ $f(x)=\exp(x)+x$	6
1.2	Méthode de simpson $n=2$ $f(x)=\exp(x)+x$	7
1.3	Fonction de Runge $g(x)=1/(1+25x^2)$ (<i>ligne bleue</i>).	8
1.4	Méthode de trapèze composite $n=5$ $f(x)=\exp(x)+x$	9
1.5	Méthode de Simpson composite $m=5$ $f(x)=\exp(x)+x$	10
3.1	Allure des différentes fonctions test, avec une valeur de $u=0.8$ et différentes valeurs de $a=1$ (ligne noire), $a=2$ (ligne verte), $a=9$ (ligne blue), $a=8$ (ligne rouge)	28
3.2	L'allure des erreurs des fonctions de Genz Trapèze généralisée (tiré plein vert), moindre carré généralisée d'ordre 1 (tiré plein noire), Simpson généralisée (tiré plein bleue) et moindre carré généralisée (tiré plein rouge)	32

Liste des tableaux

1.1 Erreurs d'intégration de la fonction $f(x)=\exp(x)+x$ par Trapèze et Simpson . . .	8
1.2 Comparaison entre formules simples et composites	11
2.1 Le tableau des erreurs associées à IMC1, IMC2 de la fonction $f(x)=\exp(x)+x$.	17
2.2 Le tableau des erreurs associé à IMCD1 et IMCD2 de la fonction $f(x,y)=x^2/(1+y^2)$	21
3.1 Tableau associé à la fonction oscillatoire	29
3.2 Tableau associé à la fonction continue	29
3.3 Tableau associé à la fonction de pic de produit	29
3.4 Tableau associé à la fonction de crête de coin	30
3.5 Tableau à la fonction gaussienne	30
3.6 Tableau la fonction discontinue	30
3.7 Tableau des erreurs associé aux fonctions de Genz	31
3.8 Tableau associé à la fonction continue en 2 D	33
3.9 Tableau associé à la fonction gaussienne en 2 D	33

Dédicace

Je dédie cet humble travail :

A mes parents *qui se sont sacrifiés pour ma réussite, de part leur amour immense, leur soutien, ainsi que leur confiance absolue et une éducation dont je suis fière.*

A mes frères *Souleyman et Mohammed qui étaient à mes cotes tout le temps.*

A mes soeurs *aussi, qui m'ont encouragé et m'ont soutenu.*

A l'esprit de ma grande sœur *Amina (Allah yarhamha), qui restera gravée dans nos mémoire à l'éternité.*

A toute ma famille et mes chères amies S.Amel, B.Fatima, Z.Rachida, B.Marwa, B.Nebila, B.Nour El Houda et B.Meriem.

Je vous dis merci.

Remerciements

Je remercie tout d'abord « ALLAH » de m'avoir donné la force, la volonté et le courage de mener à bout mon travail (El hemdou lillah).

*J'exprime toute ma gratitude à Mon Encadreuse madame **Naima Ablaoui-Lahmar**, pour l'effort fourni, les conseils prodigués, sa patience et son suivi du début à la fin de ce travail.*

Je remercie l'ensemble des enseignants du département de Mathématique et tous ceux qui m'ont aidé de près ou de loin par un geste, une parole ou un conseil.

*Je remercie très sincèrement, Mme **HAMOU MAAMAR Maghnia** et Mme **BENSIKADDOUR Djemaia** pour avoir accepté de faire partie du jury.*

*Je remercie aussi madame **Zineb Kaissarli**, pour les conseils et les remarques qu'elle nous a prodigué lors du séminaire.*

*Je remercie Mme **Leila BOUZID**, qui m'a aidé dans travail.*

Encore une fois, un énorme merci aux amis et à ma famille.

Enfin je remercie mes collègues de la promotion M2 (MCO) et (AF).

Merci à tous

Résumé

Des nouvelles méthodes de quadrature et cubature appelées (**IMC**) et (**IMCD**) sont établies, elles sont basées sur l'approximation au sens des moindres carrés. Ces méthodes ont été implémentées sur un code **MATLAB**, validées en utilisant la base de données des fonctions de Genz et comparées avec les méthodes de Newton-Côtes composites.

Mot-Clés : Intégration numérique, approximation, formule de quadrature, formule de cubature.

AMS Classification 65D30, 65D32.

Abstract

New methods of quadrature and cubature called (**IMC**) and (**IMCD**) are established, they are based on the approximation in the sense of the least squares. These methods were implemented on a **MATLAB** code, validated using the Genz function database and compared with composite Newton-Côtes methods.

Key words : numerical integration, approximation, , quadrature formula, formula of cubature.

AMS Classification 65D30, 65D32.

Introduction générale

Les intégrales simples et multidimensionnelles apparaissent fréquemment dans les sciences appliquées, comme cela est le cas en statistique (pour exprimer les lois de probabilité...), en acoustique sous marine (pour décrire les fréquences sonores issues d'une source [4]), en astrophysique (pour décrire l'intensité du transfert de rayonnement d'une étoile ou d'une planète [7]).

En une dimension, il s'agit de calculer

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

où f est une fonction donnée intégrable sur un domaine $[a, b]$ de \mathbb{R} .

Si une primitive F de f est connue, d'après la formule de Newton-Leibniz, ce calcul est donné par

$$I(f) = F(b) - F(a), \quad F' = f.$$

Mais, le plus souvent un des cas suivants se présente :

- Le calcul analytique est long, compliqué et rébarbatif
- Le résultat de l'intégrale est une fonction compliquée qui fait appel à d'autres fonctions elles-mêmes longues à évaluer
- Cette intégrale n'a pas d'expression analytique, comme pour les fonctions $f(x) = \cos(x^2)$, $f(x) = \frac{\sin(x)}{x}$.
- La fonction f est connue seulement en un certain nombre de points $x_i, i = 0, \dots, n$.

Dans tous ces cas, il est préférable de calculer numériquement la valeur de l'intégrale $I(f)$.

Il existe plusieurs méthodes (basées sur des principes différents) pour approcher numériquement $I(f)$. On peut citer les méthodes de Gauss, les méthodes de Newton-Côte ou alors les méthodes de Monté-Carlo.

Ce travail a pour but d'établir une nouvelle méthode basée sur l'approximation au sens des moindres carrés pour estimer les intégrales simples (unidimensionnelles) et doubles (bidimensionnelles).

Ce mémoire contient trois chapitres :

Le premier chapitre est un rappel des formules les plus utilisées qui approchent $I(f)$.

Dans le deuxième chapitre, nous allons formuler la nouvelle méthode d'intégration et ceci en approchant la fonction f par son polynôme d'approximation aux sens des moindres carrés dans le cas unidimensionnel et bidimensionnel.

Le troisième chapitre est réservé aux tests numériques afin de valider cette nouvelle formule. Nous nous appuyons sur la base de données des fonctions de Genz.

L'intégration numérique

Ce chapitre repose sur le livre [2].

1.1 Introduction

Le but de ce chapitre est de rappeler quelques formules des plus utilisées qui approchent $\int_a^b f(x) dx$ où f est une fonction intégrable donnée sur $[a, b]$ de \mathbb{R} .

Ces formules sont obtenues en approchant la fonction f par une fonction simple φ et en suite admettre que

$$\int_a^b f(x) dx \approx \int_a^b \varphi(x) dx.$$

Remarque 1.1.1 La fonction $\varphi(x)$ doit être choisie telle que le calcul de l'intégrale $\int_a^b \varphi(x) dx$ soit immédiat.

Définition 1.1.1 d'une formule de quadrature

Une formule de quadrature est une formule qui permet de calculer numériquement une intégrale simple (**quadrature mécanique**).

Elle prend généralement la forme suivante :

$$\int_a^b f(x) dx \approx \sum_{i=0}^n A_i f(x_i), \quad (1.1.1)$$

où les coefficients A_i , $i = 0, \dots, n$ sont à déterminer et sont indépendants de la fonction à intégrer f .

Nous appelons erreur de quadrature associée à la formule (1.1.1), la quantité notée $R(f)$ et définie par

$$R(f) = \int_a^b f(x)dx - \sum_{i=0}^n A_i f(x_i). \quad (1.1.2)$$

Définition 1.1.2 d'une formule exacte

La formule de quadrature (1.1.1) est dite exacte de degré n (ou bien le **degré de précision** n) si l'erreur $R(p_k) = 0$ pour tout polynôme p_k de degré $k \leq n$ et $R(p_{n+1}) \neq 0$ pour un certain polynôme de degré $(n + 1)$.

Théorème 1.1.1 Le degré de précision (de validité) est n si et seulement si $R(x^k) = 0 \forall k = 0, \dots, n$ et $R(x^{n+1}) \neq 0$.

Il existe plusieurs principes pour déterminer une formule de quadrature de type (1.1.1), parmi lesquels le principe de Gauss et le principe de Newton-Côtes. Nous nous intéressons au principe de Newton-Côte.

1.2 Les formules de Newton-Côte simples

Principe

Le principe général des méthodes de Newton-Côtes simples réside dans le choix de la fonction φ . Celle-ci est un polynôme d'interpolation.

Pour un entier n non nul choisi, nous considérons la subdivision équidistante du segment $[a, b]$

$$x_0 = a, \quad x_i = x_0 + ih \quad (i = 0, 1, \dots, n - 1), \quad x_n = b,$$

Nous rappelons que le polynôme d'interpolation sous forme de Lagrange, noté $L_n(x)$, est défini par

$$L_n(x) = \sum_{i=0}^n f(x_i)L_i(x),$$

où les polynômes élémentaires $L_i(x), i = 0, \dots, n$, sont donnés par :

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x - x_j)}{(x_i - x_j)}.$$

En approchant la fonction f par son polynôme d'interpolation L_n , et en admettant que

$$\int_{x_0}^{x_n} f(x)dx \approx \int_{x_0}^{x_n} L_n(x)dx,$$

nous obtenons

$$\int_{x_0}^{x_n} f(x)dx \approx \sum_{i=0}^n A_i f(x_i), \quad (1.2.1)$$

où les coefficients A_i sont donnés explicitement par :

$$A_i = \int_{x_0}^{x_n} \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x - x_j)}{(x_i - x_j)} dx.$$

Pour obtenir une formule plus pratique, nous posons

$$q = \frac{(x - x_0)}{h} \implies x = x_0 + qh \implies dx = hdq.$$

Avec cette notation nous avons :

$$\begin{aligned} \prod_{\substack{j=0 \\ j \neq i}}^n (x_i - x_j) &= \prod_{\substack{j=0 \\ j \neq i}}^n (x_0 + ih - x_j) = h^n i (i-1) (i-2) \dots 1 (-1) (-2) \dots (i-n) \\ &= h^n i! (-1)^{n-i} (n-1)!. \end{aligned}$$

L'expression de A_i devient

$$A_i = \int_0^n h \frac{h^n q (q-1) (q-2) \dots (q-(i-1)) (q-(i+1)) \dots (q-n)}{h^n i! (-1)^{n-i} (n-1)!} dq.$$

puisque $h = \frac{b-a}{n}$, alors $A_i = (b-a) H_i$, avec H_i donnée par :

$$H_i = \int_0^n \frac{q (q-1) (q-2) \dots (q-n)}{i! (-1)^{n-i} (n-1)! n (q-i)} dq \quad i = 0 \dots n. \quad (1.2.2)$$

Les constantes H_i sont appelées coefficients de Cotes et vérifient

$$\sum_{i=0}^n H_i = 1 \quad ; \quad H_i = H_{n-i} \text{ (propriété de symétrie).}$$

La formule (1.2.1) se met alors sous la forme :

$$\int_a^b f(x)dx = (b-a) \sum_{i=0}^n H_i f(x_i).$$

L'estimation de l'erreur de l'interpolation de Lagrange :

Théorème 1.2.1 Si la fonction f est $(n+1)$ fois dérivable sur $]a, b[$, l'erreur d'intégration E_I associée à la formule (1.2.1) est donnée par l'intégrale de l'erreur d'interpolation $E(x)$ c'est à dire $\forall x, \exists \xi \in]a, b[$:

$$E_I = \int_{x_0}^{x_n} E(x) dx$$

$$= \int_{x_0}^{x_n} \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i) dx \text{ telle que } \xi \in]a, b[.$$

Formule de trapèze

Pour trouver cette formule, la fonction f est approchée par le polynôme de Lagrange d'ordre 1, P_1 .

En appliquant la formule (1.2.2) pour $n = 1$, nous avons :

$$H_0 = - \int_0^1 \frac{q(q-1)}{q} dq = \frac{1}{2}, \quad H_1 = H_0 = \frac{1}{2}$$

d'où

$$\int_{x_0}^{x_1} f(x) dx \approx \frac{h}{2} (y_0 + y_1) \tag{1.2.3}$$

Théorème 1.2.2 Si la fonction f est de classe $C^2(]a, b[)$, le reste de la formule de quadrature (1.2.3) est égal à

$$Et = -\frac{(b-a)^3}{12} f''(\xi), \text{ avec } \xi \in]a, b[.$$

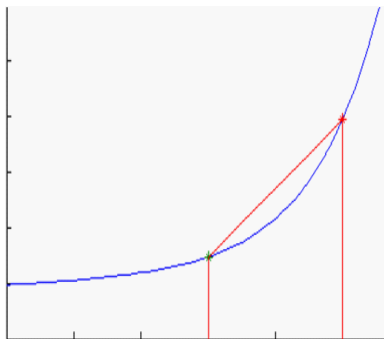


FIG. 1.1 – Méthode de trapèze $n=1$ $f(x)=\exp(x)+x$

Formule de Simpson

Cette formule est obtenue en approchant la fonction f par le polynôme d'interpolation d'ordre 2, P_2 .

En appliquant la formule (1.2.2) pour $n = 2$ nous avons :

$$H_0 = \frac{1}{2} \cdot \frac{1}{2} \int_0^2 (q-1)(q-2) dq = \frac{1}{4} \left(\frac{8}{3} - 6 + 4 \right) = \frac{1}{6}, \quad H_1 = \frac{1}{2} \cdot 1 \int_0^2 q(q-2) dq = \frac{2}{3} \text{ et } H_2 = H_0 =$$

$\frac{1}{6}$.
Donc

$$\int_{x_0}^{x_2} f(x) dx \approx \frac{h}{3} (y_0 + 4y_1 + y_2), \quad h = \frac{x_2 - x_0}{2}. \tag{1.2.4}$$

Théorème 1.2.3 Si la fonction f est de class $C^4(]a, b[)$ alors, le reste de la formule de quadrature (1.2.4) est égal à

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

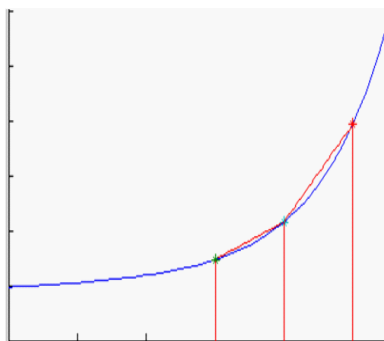


FIG. 1.2 – Méthode de simpson n=2 $f(x)=\exp(x)+x$

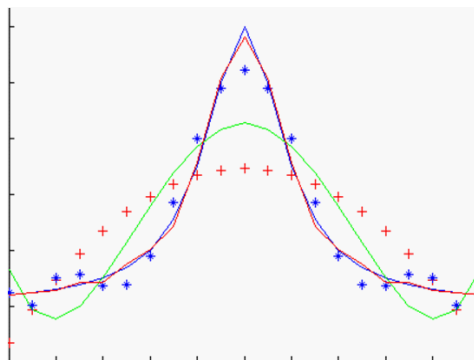
Exemple 1.2.1 Application des formules de Trapèze et de Simpson sur une fonction simple $f(x) = e^x + x$ sur deux intervalles de longueur 1 et 2.

Remarque 1.2.1 Nous remarquons que les formules de Trapèze et de Simpson ne donnent pas (toujours) des résultats satisfaisants. Pour y remédier, nous pouvons procéder de deux manières différentes :

1. Nous construisons des formules de quadrature de Newton-Côtes associées à des polynômes d'interpolation de degré plus élevé ($n \geq 3$). Mais pour des degrés élevés, le polynôme interpolant peut présenter de fortes oscillations c'est ce qui est appelé le phénomène de Runge [3] voir la figure (1.3).
2. Nous subdivisons le domaine total $[a, b]$ en un grand nombre de petits intervalles sur chacun, nous appliquons l'une des méthodes de Newton-Côtes simple, nous parlons alors des formules de Newton-Côtes composites.

h petit		h grand	
$[a, b] = [0, 1]$		$[a, b] = [2, 4]$	
$I_{EX} = 2.21828$		$I_{EX} = 53.20909$	
$E_{Trap} = 0.14086$	$E_{Simp} = 5.79323e - 4$	$E_{Trap} = 14.77811$	$E_{Simps} = 0.23402$

TAB. 1.1 – Erreurs d'intégration de la fonction $f(x)=\exp(x)+x$ par Trapèze et Simpson



Interpolation par un polynôme de degré 3 (+ rouge), degré 5 (ligne pleine verte), degré 9 (*bleue) et degré 15 (ligne pleine rouge).

1.3 Les formules de Newton-Côtes composites

Le principe général des méthodes de Newton-Côtes composites est basé sur

- La division de l'intervalle d'intégration $[a, b]$ en n parties égales $[x_i, x_{i+1}]$; $i = 0, \dots, n - 1$.
Par la relation de Chasles nous avons

$$\int_a^b f(x) dx = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x) dx. \quad (1.3.1)$$

- L'application de l'une des méthodes d'intégration simple sur chaque intervalle $[x_i, x_{i+1}]$; $i = 0, \dots, n - 1$.

L'expression de l'erreur d'intégration :

Si la fonction f est $(n + 1)$ fois dérivable sur $]a, b[$, l'erreur d'intégration E_I associée à la formule (1.3.1) est donnée par la somme des intégrales des erreurs d'interpolation $E_i(x)$ c'est-à-dire :

$$\begin{aligned}
 E_I &= \sum_{i=0}^n \int_{x_{i-1}}^{x_i} E_i(x) dx \\
 &= \sum_{i=0}^n \int_{x_{i-1}}^{x_i} \frac{f^{n+1}(\xi_i)}{(n+1)!} \prod_{i=0}^n (x - x_i) dx \text{ avec } \xi_i \in]x_{i-1}, x_i[.
 \end{aligned}$$

La formule de trapèze composite

Pour trouver cette formule, nous appliquons la formule de trapèze simple sur chaque partie de $[a, b]$.

En adoptant les notations : $h = \frac{b-a}{n}$ et $y_i = f(x_i)$; $i = 0 \dots n$, il vient

$$\int_a^b f(x) dx \approx \frac{h}{2} (y_0 + y_1) + \frac{h}{2} (y_1 + y_2) + \dots + \frac{h}{2} (y_{n-1} + y_n)$$

ou

$$\int_a^b f(x) dx \approx h \left(\frac{y_0}{2} + \frac{y_n}{2} + \sum_{i=1}^{n-1} y_i \right). \tag{1.3.2}$$

Théorème 1.3.1 Si la fonction f est de classe C^2 $]a, b[$, le reste de la formule de quadrature de Trapèze composite (1.3.2) est égal à :

$$E_{tg} = -\frac{h^3}{12} n \sum_{i=1}^n f''(\xi), \text{ avec } \xi \in]a, b[$$

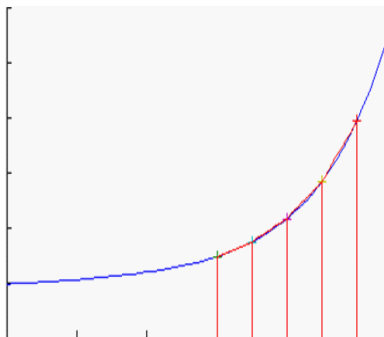


FIG. 1.4 – Méthode de trapèze composite $n=5$ $f(x)=\exp(x)+x$

La formule de Simpson composite

Pour trouver cette formule, nous appliquons la formule de Simpson simple sur chaque partie de $[a, b]$.

En adoptant les notations : $n = 2m$, $h = \frac{b-a}{2m}$ et $y_i = f(x_i)$; $i = 0, \dots, n$, il vient par le relation de Chasles

$$\int_a^b f(x) dx = \sum_{i=1}^m \int_{x_{2i-2}}^{x_{2i}} f(x) dx, \quad (3)$$

et par l'application de la formule de Simpson simple sur le sous intervalle $[x_{2i-2}, x_{2i}]$; $i = 0, \dots, m$,

$$\int_a^b f(x) dx \approx \sum_{i=1}^m \left(\frac{h}{3} (y_{2i-2} + 4y_{2i-1} + y_{2i}) \right).$$

Ou encore :

$$\int_a^b f(x) dx \approx \frac{h}{3} ((y_0 + y_{2m}) + 4 \sum_{i=1}^m y_{2i-1} + 2 \sum_{i=1}^m y_{2i}). \quad (1.3.3)$$

Théorème 1.3.2 Si la fonction f est de class $C^4([a, b])$, le reste de la formule de quadrature de Simpson composite (1.3.3) est donné par :

$$E_{sg} = -\frac{h^5}{90} m f^4(\xi); \text{ où } \xi \in]a, b[.$$

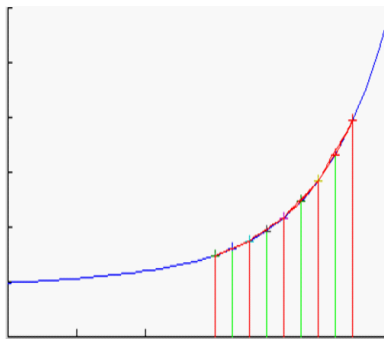


FIG. 1.5 – Méthode de Simpson composite $m=5$ $f(x)=\exp(x)+x$

Exemple 1.3.1 Pour constater l'apport des formules composites, nous allons appliquer ces quatre méthodes pour estimer l'intégrale $\int_2^4 (e^x + x) dx$. L'erreur associée à cette approximation est donnée dans le tableau suivant :

Trapèze simple	Trap composite	Simpson simple	Simp composite
14.7781	0.9794	0.2340	0.0010

TAB. 1.2 – Comparaison entre formules simples et composites

1.4 Les formules de cubature

Définition 1.4.1 *formule cubature black*

Une formule de cubature numérique est prévue pour le calcul numérique des intégrales doubles (*cubature mécanique*).

Nous voulons estimer l'intégrale double

$$\int_{a_1}^{a_2} \int_{b_1}^{b_2} f(x, y) dx dy. \quad (1.4.1)$$

Le théorème de Fubini ramène le calcul d'une intégrale double au calcul de deux itégrales simples successives, c'est à dire que (1.4.1) s'écrit comme suit

$$\int_{a_1}^{a_2} \int_{b_1}^{b_2} f(x, y) dx dy = \int_{a_1}^{a_2} \left(\int_{b_1}^{b_2} f(x, y) dy \right) dx. \quad (1.4.2)$$

Formule de cubature de Trapèze

Pour trouver cette formule, nous allons appliquer (3.1) deux fois de suite à (1.4.2). Une première application (par rapport à la variable y), donne

$$\int_{a_1}^{a_2} \int_{b_1}^{b_2} f(x, y) dx dy \approx \int_{a_1}^{a_2} \frac{b_2 - b_1}{2} (f(x, b_1) + f(x, b_2)) dx.$$

Après une seconde application de (3.1) (par rapport à la variable x), nous trouvons

$$\int_{a_1}^{a_2} \int_{b_1}^{b_2} f(x, y) dx dy \approx \frac{a_2 - a_1}{2} \frac{b_2 - b_1}{2} (f(a_1, b_1) + f(a_1, b_2) + f(a_2, b_1) + f(a_2, b_2)).$$

Formule de cubature de Simpson

Cette formule est obtenue, en appliquant (3.2) deux fois de suite à (1.4.2). Ce qui donne

$$\begin{aligned} \int_{a_1}^{a_2} \int_{b_1}^{b_2} f(x, y) dx dy \approx & \frac{a_2 - a_1}{6} \frac{b_2 - b_1}{6} (f(a_1, b_1) + 4f(\frac{a_1 + a_2}{2}, b_1) + f(a_2, b_1) + 4f(a_1, \frac{b_1 + b_2}{2}) + \\ & + 16f(\frac{a_1 + a_2}{2}, \frac{b_1 + b_2}{2}) + 4f(a_2, \frac{b_1 + b_2}{2}) + f(a_1, b_2) + \\ & + 4f(\frac{a_1 + a_2}{2}, b_2) + f(a_2, b_2)) \end{aligned}$$

Formules d'intégration basées sur l'approximation

2.1 Introduction

Dans ce chapitre, nous allons établir une nouvelle formule d'intégration numérique en se basant sur le principe de Newton-Côtes. Au lieu d'approcher la fonction f par son polynôme d'interpolation, nous allons l'approcher par son polynôme d'approximation au sens des moindres carrés P_n (de degré inférieur ou égal à n). Ce polynôme rend la quantité :

$$\|f - P_n\| = \sum_{i=0}^N |f(p_i) - P_n(p_i)|^2,$$

minimale. Où p_i ; $0 \leq i \leq N$ sont des points du domaine d'intégration et $N > n$.

Nous allons commencer par rappeler le cadre théorique de l'approximation aux sens des moindres carrés [6].

2.2 Rappels et définitions

Espaces vectoriels normés et espaces préhilbertiens

Définition 2.2.1 (*d'une distance*) Soit E un espace vectoriel sur \mathbb{R} . L'application d :

$$d : E \times E \longrightarrow \mathbb{R}_+$$

est une distance si elle vérifie les 3 propriétés :

- $d(x, x) = 0$, $\forall x \in E$ et $d(x, y) \geq 0$, $\forall x, y \in E$.
- $d(x, y) = d(y, x)$, $\forall x, y \in E$.
- $d(x, y) \leq d(x, z) + d(z, y)$, $\forall x, y, z \in E$.

(E, d) est dit espace métrique.

Définition 2.2.2 (d'une norme) L'application

$$\|\cdot\|_E : E \longrightarrow \mathbb{R}_+$$

est une norme sur E si elle vérifie les 3 propriétés :

- $\|x\|_E \geq 0$, $\forall x \in E$ et $\|x\|_E = 0 \iff x = 0_E$.
- $\|\lambda x\|_E = |\lambda| \|x\|_E$, $\forall x \in E$, $\forall \lambda \in \mathbb{R}_+$.
- $\|x + y\|_E \leq \|x\|_E + \|y\|_E$, $\forall x \in E, \forall y \in E$
 $(E, \|\cdot\|_E)$ est dit espace normé.

Exemple 2.2.1 La droite réelle \mathbb{R} , munie de la valeur absolue $|\cdot|$ est un espace vectoriel normé.

Remarque 2.2.1 Tout espace vectoriel normé $(E, \|\cdot\|_E)$ est en particulier un espace métrique pour la distance d , définie par :

$$d : E \times E \longrightarrow \mathbb{R}_+ \\ (x, y) \rightsquigarrow d(x, y) = \|x - y\|_E$$

Définition 2.2.3 (d'un produit scalaire)

Un produit scalaire sur le \mathbb{R} -espace vectoriel E est une application, notée $\langle \cdot, \cdot \rangle$, définie sur $E \times E$, à valeurs dans \mathbb{R} et qui vérifie :

- $\langle x, x \rangle \geq 0$, $\forall x \in E$ et $\langle x, x \rangle = 0 \iff x = 0$.
- $\langle x, y \rangle = \langle y, x \rangle$, $\forall x \in E, \forall y \in E$
- $\langle \lambda x_1 + \gamma x_2, y \rangle = \lambda \langle x_1, y \rangle + \gamma \langle x_2, y \rangle$, $\forall x_1, x_2, y \in E, \forall \lambda, \gamma \in \mathbb{R}$.

Remarque 2.2.2 - L'espace vectoriel E muni d'un produit scalaire $(E, \langle \cdot, \cdot \rangle)$ est appelé espace préhilbertien.

- Tout espace $(E, \langle \cdot, \cdot \rangle)$ muni d'un produit scalaire est un espace normé pour la norme "induite" et qui est définie par :

$$\|\cdot\|_E : E \longrightarrow \mathbb{R}_+ \\ x \rightsquigarrow \|x\| = \langle x, x \rangle^{\frac{1}{2}}$$

Exemple 2.2.2 La norme induite sur \mathbb{R}^n est :

$$\|x\|_{\mathbb{R}^n} = \langle x, x \rangle^{\frac{1}{2}} = \left(\sum_{i=1}^N x_i^2 \right)^{1/2}, \forall x \in \mathbb{R}^n.$$

Meilleure approximation dans un espace vectoriel normé

Soit $(E, \|\cdot\|)$ un espace vectoriel normé.

Définition 2.2.4 Soient f un élément de E et B un sous-espace de E . L'élément $y \in B$ qui réalise :

$$\|f - y\| = \min_{z \in B} \|f - z\|$$

est appelé la meilleure approximation ou (le meilleur approximant) de f dans B . Nous disons aussi que y réalise la meilleure approximation de f dans B .

Caractérisation du meilleur approximant dans un espace préhilbertien

La caractérisation du meilleur approximant est basée sur le théorème suivant

Théorème 2.2.1 [6] Soit B un sous-espace vectoriel d'un espace préhilbertien $(E, \langle \cdot, \cdot \rangle)$. Soit $f \in E$. L'élément $y \in B$ est le meilleur approximant de f si et seulement si $(f - y)$ est orthogonal à tout élément $z \in B$.

Nous supposons que le sous-espace $B \subset E$ est engendré par la famille $\{g_0, g_1, \dots, g_N\} \subset E$. Le meilleur approximant y de f s'écrit alors :

$$y = \sum_{i=0}^N a_i g_i \text{ où } a_i \in \mathbb{R}.$$

D'après le théorème précédent, $(f - y)$ est orthogonal à B , ce qui revient à dire que

$$\left\langle f - \sum_{i=0}^N a_i g_i, z \right\rangle = 0, \forall z \in B.$$

En particulier, pour tout $g_j \in B$, $j = 0, 1, 2, \dots, N$, nous avons :

$$\left\langle f - \sum_{i=0}^N a_i g_i, g_j \right\rangle = 0 \iff \sum_{i=0}^N a_i \langle g_i, g_j \rangle = \langle f, g_j \rangle ; j = 0, 1, 2, \dots, N. \quad (2.2.1)$$

L'ensemble des relations (2.2.1) prend la forme matricielle :

$$\begin{bmatrix} \langle g_0, g_0 \rangle & \langle g_1, g_0 \rangle & \dots & \langle g_N, g_0 \rangle \\ \langle g_0, g_1 \rangle & \langle g_1, g_1 \rangle & \dots & \langle g_N, g_1 \rangle \\ \dots & \dots & \dots & \dots \\ \langle g_0, g_N \rangle & \langle g_1, g_N \rangle & \dots & \langle g_N, g_N \rangle \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \cdot \\ a_N \end{bmatrix} = \begin{bmatrix} \langle f, g_0 \rangle \\ \langle f, g_1 \rangle \\ \cdot \\ \langle f, g_N \rangle \end{bmatrix} \quad (2.2.2)$$

La matrice de ce système est appelée **matrice de Gram** de l'ensemble $\{g_0, g_1, \dots, g_N\}$. Si les éléments g_i ; $i = 0, 1, \dots, N$ de cet ensemble sont linéairement indépendants, le déterminant de Gram est différent de zéro. Ceci entraîne que le système (2.2.2) possède une solution unique $(a_0, a_1, \dots, a_N)^T$, d'où l'existence et l'unicité du meilleur approximant $y = \sum_{i=0}^N a_i g_i$.

Si $\{g_0, g_1, \dots, g_N\}$ sont liés, le déterminant de Gram est nul et par conséquent $(a_0, a_1, \dots, a_N)^T$ n'est pas la seule solution du système (2.2.2). Dans ce cas le meilleur approximant ne peut pas être écrit d'une manière unique comme une combinaison linéaire des éléments g_i ; $i = 0, 1, \dots, N$.

2.3 Formules de quadrature basées sur l'approximation

2.3.1 Nouvelle formule de quadrature simple

Soit $E = C^0([a, b])$ l'ensemble des fonctions continues sur un intervalle $[a, b]$ de \mathbb{R} . Soit f une fonction dans E . Pour établir la nouvelle formule de quadrature permettant d'estimer

$\int_a^b f(x)dx$, nous allons approcher la fonction f par son polynôme d'approximation au sens des moindres carrés discrète. Dans ce cas, nous considérons

- Le sous espace $B = P_n$ (l'ensemble des polynômes de degré inférieur ou égal à n).
- La base considérée $g_i = x^i$; $i = 0, 1, \dots, n$.
- $n + 2$ points x_k , $k = 0, 1, \dots, n + 1$ choisis dans l'intervalle $[a, b]$.
- Le produit scalaire $\langle g_i, g_j \rangle = \sum_{k=0}^{n+1} g_i(x_k)g_j(x_k)$; $i, j = 0, 1, \dots, n$.

Le meilleur approximant de f est alors l'élément $y = \sum_{i=0}^n \alpha_i x^i$, où les coefficients α_i ; ($i = 0, \dots, n$) sont solutions du système

$$\begin{bmatrix} n+2 & \sum_{i=0}^{n+1} x_i & \cdots & \sum_{i=0}^{n+1} x_i^n \\ \sum_{i=0}^{n+1} x_i & \sum_{i=0}^{n+1} x_i^2 & \cdots & \sum_{i=0}^{n+1} x_i^{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{n+1} x_i^n & \sum_{i=0}^{n+1} x_i^{n+1} & \cdots & \sum_{i=0}^{n+1} x_i^{2n} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n+1} f(x_i) \\ \sum_{i=0}^{n+1} x_i f(x_i) \\ \vdots \\ \sum_{i=0}^{n+1} x_i^n f(x_i) \end{bmatrix} \quad (2.3.1)$$

Théorème 2.3.1 *La formule*

$$IMC := \sum_{i=0}^n \frac{\alpha_i}{i+1} (b^{i+1} - a^{i+1})$$

est une approximation de $\int_a^b f(x)dx$ où les coefficients α_i ; ($i = 0, \dots, n$) sont solutions du système (2.3.1).

Preuve. Comme

$$\begin{aligned} f(x) &\approx \sum_{i=0}^n \alpha_i x^i \\ \Rightarrow \int_a^b f(x)dx &\approx \int_a^b \sum_{i=0}^n \alpha_i x^i dx \\ &\approx \sum_{i=0}^n \alpha_i \int_a^b x^i dx \\ &\approx \sum_{i=0}^n \alpha_i \left(\frac{x^{i+1}}{i+1} \right)_{x=a}^{x=b}. \end{aligned}$$

□

Nous allons considérer, essentiellement, deux cas particuliers.

Approximation par un polynôme de degré 1

Pour $n = 1$, le meilleur approximant y dans B (Le sous-espace engendré par $\{1, x\}$), s'écrit sous la forme

$$y = \alpha_0 + \alpha_1 x,$$

où les coefficients α_0 et α_1 sont solutions du système suivant :

$$\begin{bmatrix} 3 & \sum_{i=0}^2 x_i \\ \sum_{i=0}^2 x_i & \sum_{i=0}^2 x_i^2 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^2 f(x_i) \\ \sum_{i=0}^2 x_i f(x_i) \end{bmatrix}.$$

La formule d'intégration (IMC) prend la forme :

$$\int_a^b f(x) dx \approx \alpha_0(b-a) + \frac{\alpha_1}{2}(b^2 - a^2).$$

Nous adoptons la notation suivante

$$IMC1 := \alpha_0(b-a) + \frac{\alpha_1}{2}(b^2 - a^2).$$

Approximation par un polynôme de degré 2

Pour $n = 2$, le meilleur approximant y dans B (Le sous-espace engendré par $\{1, x, x^2\}$), s'écrit sous la forme

$$y = \alpha_0 + \alpha_1 x + \alpha_2 x^2,$$

où les coefficients α_0, α_1 et α_2 sont donnés par le système suivant :

$$\begin{bmatrix} 4 & \sum_{i=0}^3 x_i & \sum_{i=0}^3 x_i^2 \\ \sum_{i=0}^3 x_i & \sum_{i=0}^3 x_i^2 & \sum_{i=1}^3 x_i^3 \\ \sum_{i=0}^3 x_i^2 & \sum_{i=0}^3 x_i^3 & \sum_{i=0}^3 x_i^4 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^3 f(x_i) \\ \sum_{i=0}^3 x_i f(x_i) \\ \sum_{i=0}^3 x_i^2 f(x_i) \end{bmatrix}.$$

La formule d'intégration (IMC) prend la forme :

$$\int_a^b f(x) dx \approx IMC2,$$

$$IMC2 := \alpha_0(b-a) + \frac{\alpha_1}{2}(b^2 - a^2) + \frac{\alpha_2}{3}(b^3 - a^3).$$

Exemple 2.3.1 Application des formules de quadrature (IMC1) et (IMC2), pour estimer l'intégrale de la fonction $f(x) = e^x + x$.

Remarque 2.3.1 Les raisons pour lesquelles les formules de Newton-Côtes composites ont été introduites, sont encore observées dans cette nouvelle méthode. Ce qui nous contraint à considérer les formules composites associées à cette nouvelle méthode.

$[a, b] = [0, 1]$		$[a, b] = [2, 4]$	
$I_{EX} = 2.21828$		$I_{EX} = 53.20909$	
$E_{IMC1} = 0.07072$	$E_{IMC2} = 2.58325e - 004$	$E_{IMC1} = 7.50607$	$E_{IMC2} = 0.10536$

TAB. 2.1 – Le tableau des erreurs associées à IMC1, IMC2 de la fonction $f(x)=\exp(x)+x$

2.3.2 Les formules de quadrature composites

Le principe général des méthodes composites est basé sur

- La division de l'intervalle d'intégration $[a, b]$ en N parties égales $[x_j, x_{j+1}] ; j = 0 \dots N - 1$.

Par la relation de Chasles nous avons

$$\int_a^b f(x) dx = \sum_{j=0}^{N-1} \int_{x_j}^{x_{j+1}} f(x) dx.$$

- L'application de l'une des nouvelles méthodes d'intégration simple sur chaque intervalle $[x_j, x_{j+1}] ; j = 0 \dots N - 1$. Ce qui donne

$$\int_a^b f(x) dx \approx \sum_{j=0}^{N-1} \sum_{i=0}^n \frac{a_{i,j}}{i+1} (x_{j+1}^{i+1} - x_j^{i+1}). \tag{2.3.2}$$

Les réels $a_{i,j}$ sont les coefficients du polynôme d'approximation ($y = \sum_{i=0}^n a_{i,j} x^i$) sur l'intervalle $[x_j, x_{j+1}] , j = 0 \dots N - 1$.

Formule des moindres carrés d'ordre 1 composite (IMCC1)

La formule des moindres carrés d'ordre 1 composite est obtenue en remplaçant $n = 1$ dans (2.3.2).

$$\int_a^b f(x) dx \approx \sum_{j=0}^{N-1} \alpha_{0,j} (x_{j+1} - x_j) + \alpha_{1,j} \frac{x_{j+1}^2 - x_j^2}{2},$$

avec les coefficients $\alpha_{0,j}, \alpha_{1,j}$ sont solutions du système :

$$\begin{bmatrix} 3 & \sum_{k=0}^2 x_{k,j} \\ \sum_{k=0}^2 x_{k,j} & \sum_{k=0}^2 x_{k,j}^2 \end{bmatrix} \begin{bmatrix} \alpha_{0,j} \\ \alpha_{1,j} \end{bmatrix} = \begin{bmatrix} \sum_{k=0}^2 f(x_{k,j}) \\ \sum_{k=0}^2 x_{k,j} f(x_{k,j}) \end{bmatrix}.$$

Les points $x_{k,j}$ sont définis par $x_{k,j} = x_j + k \frac{x_{j+1} - x_j}{3}$ avec $k = 0, 1, 2$.

Formule des moindres carrés d'ordre 2 composite (IMCC2)

La formule des moindres carrés d'ordre 2 composite est obtenue en remplaçant $n = 2$ dans (2.3.2).

$$\int_a^b f(x)dx \approx \sum_{j=0}^{N-1} a_{0,j}(x_{j+1} - x_j) + \alpha_{1,j} \frac{x_{j+1}^2 - x_j^2}{2} + \alpha_{2,j} \frac{x_{j+1}^3 - x_j^3}{3},$$

avec les coefficients $\alpha_{0,j}$, $\alpha_{1,j}$ et $\alpha_{2,j}$ sont solutions du système :

$$\begin{bmatrix} 4 & \sum_{k=0}^3 x_{k,j} & \sum_{k=0}^3 x_{k,j}^2 \\ \sum_{k=0}^3 x_{k,j} & \sum_{k=0}^3 x_{k,j}^2 & \sum_{k=0}^3 x_{k,j}^3 \\ \sum_{k=0}^3 x_{k,j}^2 & \sum_{k=0}^3 x_{k,j}^3 & \sum_{k=0}^3 x_{k,j}^4 \end{bmatrix} \begin{bmatrix} \alpha_{0,j} \\ \alpha_{1,j} \\ \alpha_{2,j} \end{bmatrix} = \begin{bmatrix} \sum_{k=0}^3 f(x_{k,j}) \\ \sum_{k=0}^3 x_{k,j} f(x_{k,j}) \\ \sum_{k=0}^3 x_{k,j}^2 f(x_{k,j}) \end{bmatrix}.$$

Les points $x_{k,j}$ sont définis par $x_{k,j} = x_j + k \frac{x_{j+1} - x_j}{4}$ avec $k = 0, 1, 2, 3$.

2.4 Les formules de cubature basées sur l'approximation

2.4.1 Nouvelle formule cubique simple

Soit $E = C^0([a_1, a_2] \times [b_1, b_2])$ l'ensemble des fonctions continues sur un intervalle $[a_1, a_2] \times [b_1, b_2]$ de \mathbb{R}^2 . Soit f une fonction dans E . Pour établir la nouvelle formule de cubature permettant d'estimer $\int_{a_1}^{a_2} \int_{b_1}^{b_2} f(x, y) dx dy$, nous allons approcher la fonction f par son polynôme d'approximation au sens des moindres carrés discrète. Dans ce cas, nous prenons

- Le sous espace $B = P_n$ (l'ensemble des polynômes de degré inférieur ou égal à n).
- La base considérée est $g_{i,j} = x^i y^j$; $i, j = 0, 1, \dots, n$.
- $(n+2)^2$ points $(x_k, y_{k'})$; $k, k' = 0, 1, \dots, n+1$ choisis dans l'intervalle $[a_1, a_2] \times [b_1, b_2]$.
- Le produit scalaire $\langle g_{i,j}, g_{l,m} \rangle = \sum_{k,k'=0}^{n+1} g_{i,j}(x_k, y_{k'}) g_{l,m}(x_k, y_{k'})$; $i, j, l, m = 0, 1, \dots, n$.

Le meilleur approximant de f est alors l'élément $y = \sum_{j=0}^n \sum_{i=0}^n \alpha_{i,j} x^i y^j$, où les coefficients $\alpha_{i,j}$; ($i, j = 0, \dots, n$) sont solutions du système

$$\begin{bmatrix} N & \sum_{i=0}^{N-1} x_i & \cdots & \sum_{i=0}^{N-1} x_i^n y_i^n \\ \sum_{i=0}^{N-1} x_i & \sum_{i=0}^{N-1} x_i^2 & \cdots & \sum_{i=0}^{N-1} x_i^{n+1} y_i^n \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{N-1} x_i^n & \sum_{i=0}^{N-1} x_i^{n+1} & \cdots & \sum_{i=0}^{N-1} x_i^{2n} y_i^{2n} \end{bmatrix} \begin{bmatrix} \alpha_{0,0} \\ \alpha_{1,0} \\ \vdots \\ \alpha_{n-1,n} \\ \alpha_{n,n} \end{bmatrix} = \begin{bmatrix} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} f(x_i, y_j) \\ \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} x_i f(x_i, y_j) \\ \vdots \\ \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} x_i^n y_j^n f(x_i, y_j) \end{bmatrix}, \quad (2.4.1)$$

avec $N = (n+2)^2$

Proposition 2.4.1 *La formule*

$$IMCD := \sum_{j=0}^n \sum_{i=0}^n \frac{\alpha_{i,j}}{(j+1)(i+1)} (a_2^{i+1} - a_1^{i+1})(b_2^{j+1} - b_1^{j+1}), \quad (2.4.2)$$

est une approximation de $\int_{a_1}^{a_2} \int_{b_1}^{b_2} f(x,y) dx dy$ où les coefficients $\alpha_{i,j}$; ($i, j = 0, \dots, n$) sont solutions du système (2.4.1).

Nouvelle formule cubique d'ordre 1 Pour $n = 1$, le meilleur approximant y dans B (Le sous-espace polynômiale engendré par $\{1, x, y, xy\}$), s'écrit sous la forme

$$f(x, y) dx dy \approx \alpha_{0,0} + \alpha_{1,0}x + \alpha_{0,1}y + \alpha_{1,1}xy,$$

avec les coefficients $\alpha_{0,0}, \alpha_{1,0}, \alpha_{0,1}, \alpha_{1,1}$ sont solutions du système suivant :

$$\begin{bmatrix} 9 & 3X & 3Y & XY \\ 3X & 3 \sum_{i=0}^2 x_i^2 & XY & \left(\sum_{i=0}^2 x_i^2 \right) Y \\ 3Y & XY & 3 \sum_{i=0}^2 y_i^2 & X \left(\sum_{i=0}^2 y_i^2 \right) \\ XY & \left(\sum_{i=0}^2 x_i^2 \right) Y & X \left(\sum_{i=0}^2 y_i^2 \right) & \left(\sum_{i=0}^2 x_i^2 \right) \left(\sum_{i=0}^2 y_i^2 \right) \end{bmatrix} \begin{bmatrix} \alpha_{0,0} \\ \alpha_{1,0} \\ \alpha_{0,1} \\ \alpha_{1,1} \end{bmatrix} = \begin{bmatrix} \sum_{j=0}^2 \sum_{i=0}^2 f(x_i, y_j) \\ \sum_{j=0}^2 \sum_{i=0}^2 x_i f(x_i, y_j) \\ \sum_{j=0}^2 \sum_{i=0}^2 y_j f(x_i, y_j) \\ \sum_{j=0}^2 \sum_{i=0}^2 x_i y_j f(x_i, y_j) \end{bmatrix},$$

avec $X = \sum_{i=0}^2 x_i$ et $Y = \sum_{i=0}^2 y_i$.

La formule d'intégration (IMCD) prend la forme :

$$\int_{a_1}^{a_2} \int_{b_1}^{b_2} f(x, y) dx dy \approx IMCD1$$

$$IMCD1 := a_{0,0}(b_2 - b_1)(a_2 - a_1) + a_{1,0} \left(\frac{a_2^2 - a_1^2}{2} \right) (b_2 - b_1) + a_{0,1} (a_2 - a_1) \left(\frac{b_2^2 - b_1^2}{2} \right) + \dots \\ \dots + a_{1,1} \left(\frac{a_2^2 - a_1^2}{2} \right) \left(\frac{b_2^2 - b_1^2}{2} \right).$$

Nouvelle formule cubique d'ordre 2 Pour $n = 2$, le meilleur approximant y dans B (Le sous-espace polynômiale engendré par $\{1, x, x^2, y, xy, x^2y, y^2, xy^2, x^2y^2\}$), s'écrit sous la forme

$$f(x, y) dx dy \approx \alpha_{0,0} + \alpha_{1,0}x + \alpha_{2,0}x^2 + \alpha_{0,1}y + \alpha_{1,1}xy + \alpha_{2,1}x^2y + \alpha_{0,2}y^2 + \alpha_{1,2}xy^2 + \alpha_{2,2}x^2y^2,$$

avec les coefficients $\alpha_{0,0}, \alpha_{1,0}, \alpha_{2,0}, \alpha_{0,1}, \alpha_{1,1}, \alpha_{2,1}, \alpha_{0,2}, \alpha_{1,2}, \alpha_{2,2}$ sont solutions du système $MA = B$:

$$M = \begin{bmatrix} 16 & 4X_1 & 4X_2 & 4Y_1 & X_1Y_1 & Y_1X_2 & 4Y_2 & X_1Y_2 & X_2Y_2 \\ 4X_1 & 4X_2 & 4X_3 & X_1Y_1 & Y_1X_2 & Y_1X_3 & X_1Y_2 & X_2Y_2 & X_3Y_2 \\ 4X_2 & 4X_3 & 4X_4 & Y_1X_2 & Y_1X_3 & Y_1X_4 & X_2Y_2 & X_3Y_2 & X_4Y_2 \\ 4Y_1 & X_1Y_1 & Y_1X_2 & 4Y_2 & X_1Y_2 & X_2Y_2 & 4Y_3 & X_1Y_3 & X_2Y_3 \\ X_1Y_1 & Y_1X_2 & Y_1X_3 & X_1Y_2 & X_2Y_2 & X_3Y_2 & X_1Y_3 & X_2Y_3 & X_3Y_3 \\ Y_1X_2 & Y_1X_3 & Y_1X_4 & Y_2X_2 & X_3Y_2 & X_4Y_2 & X_2Y_3 & X_3Y_3 & X_4Y_3 \\ 4Y_2 & X_1Y_2 & X_2Y_2 & 4Y_3 & X_1Y_3 & X_2Y_3 & 4Y_4 & X_1Y_4 & X_2Y_4 \\ X_1Y_2 & X_2Y_2 & X_3Y_2 & X_1Y_3 & X_2Y_3 & X_3Y_3 & X_1Y_4 & X_2Y_4 & X_3Y_4 \\ X_2Y_2 & X_3Y_2 & X_4Y_2 & X_2Y_3 & X_3Y_3 & X_4Y_3 & X_2Y_4 & X_3Y_4 & X_4Y_4 \end{bmatrix},$$

$$A = [\alpha_{0,0}, \alpha_{1,0}, \alpha_{2,0}, \alpha_{0,1}, \alpha_{1,1}, \alpha_{2,1}, \alpha_{0,2}, \alpha_{1,2}, \alpha_{2,2}]^T$$

et

$$B = \begin{bmatrix} \sum_{j=0}^3 \sum_{i=0}^3 f(x_i, y_j), \sum_{j=0}^3 \sum_{i=0}^3 x_i f(x_i, y_j), \sum_{j=0}^3 \sum_{i=0}^3 x_i^2 f(x_i, y_i), \sum_{j=0}^3 \sum_{i=0}^3 y_j f(x_i, y_i), \dots \\ , \dots, \sum_{j=0}^3 \sum_{i=0}^3 x_i y_j f(x_i, y_i), \sum_{j=0}^3 \sum_{i=0}^3 x_i^2 y_j f(x_i, y_i), \dots \\ , \dots, \sum_{j=0}^3 \sum_{i=0}^3 y_j^2 f(x_i, y_i), \sum_{j=0}^3 \sum_{i=0}^3 x_i y_j^2 f(x_i, y_i), \sum_{j=0}^3 \sum_{i=0}^3 x_i^2 y_j^2 f(x_i, y_i) \end{bmatrix}^T.$$

Avec

$$X_k = \sum_{i=0}^3 x_i^k \text{ et } Y_k = \sum_{i=0}^3 y_i^k.$$

La formule d'intégration (IMCD) prend la forme :

$$\int_{a_1}^{a_2} \int_{b_1}^{b_2} f(x, y) dx dy \approx IMCD2$$

avec

$$\begin{aligned} IMCD2 := & \alpha_{0,0}(b_2 - b_1)(a_2 - a_1) + \alpha_{1,0}(b_2 - b_1)\left(\frac{a_2^2 - a_1^2}{2}\right) + \alpha_{2,0}\left(\frac{a_2^3 - a_1^3}{3}\right)(b_2 - b_1) + \\ & + \alpha_{0,1}(a_2 - a_1)\left(\frac{b_2^2 - b_1^2}{2}\right) + \alpha_{1,1}\left(\frac{a_2^2 - a_1^2}{2}\right)\left(\frac{b_2^2 - b_1^2}{2}\right) + \\ & + \alpha_{2,1}\left(\frac{a_2^3 - a_1^3}{3}\right)\left(\frac{b_2^2 - b_1^2}{2}\right) + \alpha_{0,2}(a_2 - a_1)\left(\frac{b_2^3 - b_1^3}{3}\right) + \alpha_{1,2}\left(\frac{a_2^2 - a_1^2}{2}\right)\left(\frac{b_2^3 - b_1^3}{3}\right) + \\ & + \alpha_{2,2}\left(\frac{a_2^3 - a_1^3}{3}\right)\left(\frac{b_2^3 - b_1^3}{3}\right). \end{aligned}$$

Exemple 2.4.1 Estimation $\int_{a_1}^{a_2} \int_{b_1}^{b_2} \frac{x^2}{1+y^2} dx dy$ par les formules IMCD1 et IMCD2.

$[a_1, a_2] \times [b_1, b_2] = [0, 1] \times [0, 1]$		$[2, 5] \times [2, 4]$	
$I_{EX} = 0.2617$		$I_{EX} = 8.52808$	
$E_{IMC1} = 0.22019$	$\mathbf{E}_{IMC2} = \mathbf{9.9666e - 4}$	$E_{IMC1} = 0.15708$	$\mathbf{E}_{IMC2} = \mathbf{0.00196}$

Remarque 2.4.1 Pour les mêmes problèmes que nous avons rencontré lors de l'application des formules de quadrature ; c'est à dire que nous perdons de la précision quand l'intervalle est grand, nous allons associer les formules composites associées aux formules cubiques basées sur l'approximation des moindres carrés, notées (IMCDC1) pour le degré 1 et (IMCDC2) pour le degré 2.

2.4.2 Formules de cubature composites

Cette partie, lourde à lire, exige de la concentration vu le nombre élevé des indices.

Soient N_x, N_y deux entiers non nuls donnés. Soient $x_l, l = 0 \dots N_x - 1$ et $y_m, m = 0 \dots N_y - 1$ deux subdivisions des intervalles $[a_1, a_2]$ et $[b_1, b_2]$ respectivement, telles que $[a_1, a_2] \times [b_1, b_2] = \bigcup_{l,m} [x_l, x_{l+1}] \times [y_m, y_{m+1}]$.

Par la relation de Chasles nous avons

$$\int_{a_1}^{a_2} \int_{b_1}^{b_2} f(x, y) dx dy = \sum_{l=0}^{N_x-1} \sum_{k=0}^{N_y-1} \int_{x_l}^{x_{l+1}} \int_{y_m}^{y_{m+1}} f(x, y) dx dy.$$

La formule cubique composite associée à (2.4.2) s'écrit alors comme

$$\int_{a_1}^{a_2} \int_{b_1}^{b_2} f(x, y) dx dy \approx \sum_{l=0}^{N_x-1} \sum_{m=0}^{N_y-1} \sum_{j=0}^n \sum_{i=0}^n \frac{\alpha_{i,j}}{(j+1)(i+1)} (x_{l+1}^{i+1} - x_l^{i+1})(y_{m+1}^{j+1} - y_m^{j+1}). \quad (2.4.3)$$

Les réels $\alpha_{i,j}$ sont les coefficients du polynôme d'approximation ($y = \sum_{j=0}^n \sum_{i=0}^n a_{i,j} x^i y^j$) sur l'intervalle $[x_l, x_{l+1}] \times [y_m, y_{m+1}]$; $l = 0 \dots N_x - 1, m = 0, \dots, N_y - 1$.

Formule de cubique d'ordre 1 composite (IMCDC1)

La formule cubique d'ordre 1 composite est obtenue en remplaçant $n = 1$ dans (2.4.3).

$$\begin{aligned} \int_{a_1}^{a_2} \int_{b_1}^{b_2} f(x, y) dx dy &\approx IMCDC1 \\ IMCDC1 &:= \sum_{l=0}^{N_x-1} \sum_{m=0}^{N_y-1} \alpha_{0,0,l,m} (x_{l+1} - x_l)(y_{m+1} - y_m) + \frac{\alpha_{1,0,l,m}}{2} (x_{l+1}^2 - x_l^2) \\ &\quad (y_{m+1} - y_m) + \frac{\alpha_{0,1,l,m}}{2} (x_{l+1} - x_l)(y_{m+1}^2 - y_m^2) + \\ &\quad + \frac{\alpha_{1,1,l,m}}{4} (x_{l+1}^2 - x_l^2)(y_{m+1}^2 - y_m^2). \end{aligned}$$

avec les coefficients $\alpha_{0,0,l,m}$, $\alpha_{1,0,l,m}$, $\alpha_{0,1,l,m}$, $\alpha_{1,1,l,m}$ sont solutions du système :

$$\begin{bmatrix} 9 & 3X_1 & 3Y_1 & X_1Y_1 \\ 3X_1 & 3X_2 & X_1Y_1 & X_2Y_1 \\ 3Y_1 & X_1Y_1 & 3Y_2 & X_1Y_2 \\ X_1Y_1 & X_2Y_1 & X_1Y_2 & X_2Y_2 \end{bmatrix} \begin{bmatrix} \alpha_{0,0,l,m} \\ \alpha_{0,1,l,m} \\ \alpha_{1,0,l,m} \\ \alpha_{1,1,l,m} \end{bmatrix} = \begin{bmatrix} \sum_{j=0}^2 \sum_{i=0}^2 f(x_{i,l}, y_{j,m}) \\ \sum_{j=0}^2 \sum_{i=0}^2 x_{i,l} f(x_{i,l}, y_{j,m}) \\ \sum_{j=0}^2 \sum_{i=0}^2 y_{j,m} f(x_{i,l}, y_{j,m}) \\ \sum_{j=0}^2 \sum_{i=0}^2 x_{i,l} y_{j,m} f(x_{i,l}, y_{j,m}) \end{bmatrix},$$

avec $X_i = \sum_{k=0}^2 x_{k,l}^i$ et $Y_i = \sum_{k=0}^2 y_{k,m}^i$.

Les points $x_{k,l}$ et $y_{k,m}$ sont définis par $x_{k,l} = x_l + k \frac{x_{l+1} - x_l}{3}$ et $y_{k,m} = y_m + k \frac{y_{m+1} - y_m}{3}$, avec $k = 0, 1, 2$.

Formule de cubique d'ordre 2 composite (IMCDC2)

La formule des moindres carrés d'ordre 2 composite est obtenue en remplaçant $n = 2$ dans (2.4.3).

$$\int_{a_1}^{a_2} \int_{b_1}^{b_2} f(x, y) dx dy \approx IMCDC2$$

$$\begin{aligned} IMCDC2 := & \sum_{l=0}^{N_x-1} \sum_{m=0}^{N_y-1} \alpha_{0,0,l,m} (x_{l+1} - x_l)(y_{m+1} - y_m) + \frac{\alpha_{1,0,l,m}}{2} (x_{l+1}^2 - x_l^2)(y_{m+1} - y_m) + \\ & + \frac{\alpha_{2,0,l,m}}{3} (x_{l+1}^3 - x_l^3)(y_{m+1} - y_m) + \frac{\alpha_{0,1,l,m}}{2} (x_{l+1} - x_l)(y_{m+1}^2 - y_m^2) + \\ & + \frac{\alpha_{1,1,l,m}}{4} (x_{l+1}^2 - x_l^2)(y_{m+1}^2 - y_m^2) + \frac{\alpha_{2,1,l,m}}{6} (x_{l+1}^3 - x_l^3)(y_{m+1}^2 - y_m^2) + \\ & + \frac{\alpha_{0,2,l,m}}{3} (x_{l+1} - x_l)(y_{m+1}^3 - y_m^3) + \frac{\alpha_{1,2,l,m}}{6} (x_{l+1}^2 - x_l^2)(y_{m+1}^3 - y_m^3) + \\ & + \frac{\alpha_{2,2,l,m}}{9} (x_{l+1}^3 - x_l^3)(y_{m+1}^3 - y_m^3), \end{aligned}$$

avec les coefficients $\alpha_{0,0,l,m}$, $\alpha_{1,0,l,m}$, $\alpha_{2,0,l,m}$, $\alpha_{0,1,l,m}$, $\alpha_{1,1,l,m}$, $\alpha_{2,1,l,m}$, $\alpha_{0,2,l,m}$, $\alpha_{1,2,l,m}$ et $\alpha_{2,2,l,m}$ sont solutions du système $MA = B$:

$$M = \begin{bmatrix} 16 & 4X_1 & 4X_2 & 4Y_1 & X_1Y_1 & Y_1X_2 & 4Y_2 & X_1Y_2 & X_2Y_2 \\ 4X_1 & 4X_2 & 4X_3 & X_1Y_1 & Y_1X_2 & Y_1X_3 & X_1Y_2 & X_2Y_2 & X_3Y_2 \\ 4X_2 & 4X_3 & 4X_4 & Y_1X_2 & Y_1X_3 & Y_1X_4 & X_2Y_2 & X_3Y_2 & X_4Y_2 \\ 4Y_1 & X_1Y_1 & Y_1X_2 & 4Y_2 & X_1Y_2 & X_2Y_2 & 4Y_3 & X_1Y_3 & X_2Y_3 \\ X_1Y_1 & Y_1X_2 & Y_1X_3 & X_1Y_2 & X_2Y_2 & X_3Y_2 & X_1Y_3 & X_2Y_3 & X_3Y_3 \\ Y_1X_2 & Y_1X_3 & Y_1X_4 & Y_2X_2 & X_3Y_2 & X_4Y_2 & X_2Y_3 & X_3Y_3 & X_4Y_3 \\ 4Y_2 & X_1Y_2 & X_2Y_2 & 4Y_3 & X_1Y_3 & X_2Y_3 & 4Y_4 & X_1Y_4 & X_2Y_4 \\ X_1Y_2 & X_2Y_2 & X_3Y_2 & X_1Y_3 & X_2Y_3 & X_3Y_3 & X_1Y_4 & X_2Y_4 & X_3Y_4 \\ X_2Y_2 & X_3Y_2 & X_4Y_2 & X_2Y_3 & X_3Y_3 & X_4Y_3 & X_2Y_4 & X_3Y_4 & X_4Y_4 \end{bmatrix},$$

$$A = [\alpha_{0,0,l,m}, \alpha_{1,0,l,m}, \alpha_{2,0,l,m}, \alpha_{0,1,l,m}, \alpha_{1,1,l,m}, \alpha_{2,1,l,m}, \alpha_{0,2,l,m}, \alpha_{1,2,l,m}, \alpha_{2,2,l,m}]^T$$

et

$$B = \begin{bmatrix} \sum_{j,i=0}^3 f(x_{i,l,m}, y_{j,l,m}), \sum_{j,i=0}^3 x_{i,l,m} f(x_{i,l,m}, y_{j,l,m}), \sum_{j,i=0}^3 x_{i,l,m}^2 f(x_{i,l,m}, y_{j,l,m}), \\ , \sum_{j,i=0}^3 y_{j,l,m} f(x_{i,l,m}, y_{j,l,m}), \sum_{j,i=0}^3 x_{i,l,m} y_{j,l,m} f(x_{i,l,m}, y_{j,l,m}), \\ , \sum_{j,i=0}^3 x_{i,l,m}^2 y_{j,l,m} f(x_{i,l,m}, y_{j,l,m}), \sum_{j,i=0}^3 y_{j,l,m}^2 f(x_{i,l,m}, y_{j,l,m}), \\ , \sum_{j,i=0}^3 x_{i,l,m} y_{j,l,m}^2 f(x_{i,l,m}, y_{j,l,m}), \sum_{j,i=0}^3 x_{i,l,m}^2 y_{j,l,m}^2 f(x_{i,l,m}, y_{j,l,m}) \end{bmatrix}^T.$$

Avec $X_k = \sum_{k=0}^3 x_{k,l}^k$ et $Y_k = \sum_{k=0}^3 y_{k,m}^k$.

Les points $x_{k,l}$, $y_{k,m}$ sont définis par $x_{k,l} = x_l + k \frac{x_{l+1} - x_l}{4}$ et $y_{k,m} = x_m + k \frac{x_{m+1} - x_m}{4}$ avec $k = 0, 1, 2, 3$.

Tests et résultats numériques

3.1 Introduction

Pour valider les méthodes d'intégration basées sur l'approximation au sens des moindres carrés (IMC) et (IMCD), nous allons les tester sur une base de fonctions (appelées fonctions de Genz [1] [5]), dont nous connaissons leurs intégrales exacte sur $[0, 1]^n$; $n = 1, 2$.

Puis nous allons comparer les valeurs approchées I_{app} , obtenues à l'aide de **code MATLAB**, avec l'intégrale exacte I_{ex} . Nous allons aussi comparer ces résultats avec ceux donnés par les deux méthodes de Newton-Côtes composites (**Trapèze et Simpson**), et ceci en considérant les erreurs relatives.

Nous rappelons que l'erreur relative, notée E , est définie par

$$E = \frac{|I_{ex} - I_{app}|}{|I_{ex}|}.$$

3.2 Les fonctions test de Genz

Les fonctions de Genz de n variables sont des fonctions définies sur $[0, 1]^n$, avec $n = 1, 2$ et dépendent toutes de deux vecteurs de paramètres, à l'exception de la fonction de type "Corner Peak" :

- un vecteur de paramètres a à composantes positives, traduisant la difficulté de l'intégration
- un vecteur de paramètres u à composantes dans $[0, 1]$, permettant de déterminer un aspect de la fonction (déphasage, localisation du minimum, localisation du maximum ou localisation d'une irrégularité)

Chacune des fonctions possède une allure différente, et le nom choisi par Genz en traduit l'allure générale.

En une dimension ($n = 1$), les fonctions de Genz sont

$$\begin{aligned}
 g_1(x) &= \cos(2\pi u + ax) \quad ; \text{ fonction oscillatoire} \\
 g_2(x) &= \exp(-a|x - u|) \quad ; \text{ fonction continue} \\
 g_3(x) &= \frac{1}{a^{-2} + (x - u)^{-2}} \quad ; \text{ fonction de pic de produit} \\
 g_4(x) &= (1 + ax)^{-2} \quad ; \text{ fonction de crête de coin} \\
 g_5(x) &= \exp(-a^2(x - u)^2) \quad ; \text{ fonction gaussienne} \\
 g_6(x) &:= \begin{cases} g(x) = 0 & \text{si } x > u \\ g(x) = e^{ax} & \text{sinon.} \end{cases} \quad ; \text{ fonction discontinue}
 \end{aligned}$$

Une représentation graphique de ces fonctions est donnée par la figure (3.1).
 En deux dimensions ($n = 2$), les fonctions de Genz sont

$$\begin{aligned}
 h_1(x, y) &= \exp(-a_1|x - u_1| - a_2|y - u_2|); \text{ fonction continue} \\
 h_2(x, y) &= \exp(-a_1^2(x - u_1)^2 - a_2^2(y - u_2)^2); \text{ fonction gaussienne} \\
 h_3(x, y) &= \left(\frac{1}{a_1^{-2} + (x - u_1)^{-2}}\right)\left(\frac{1}{a_2^{-2} + (y - u_2)^{-2}}\right); \text{ fonction de pic de produit} \\
 h_4(x, y) &= \begin{cases} h_4(x, y) = 0 & ; \text{ si } x > u_1 \text{ ou } y > u_2 \\ h_4(x, y) = \exp(a_1x + a_2y); & \text{ sinon} \end{cases} \quad ; \text{ fonction discontinue}
 \end{aligned}$$

3.3 Les tests en une dimension

Les quatres formules de quadrature composites (Trapèze, Simpson, IMC1, IMC2) sont testées pour les différentes fonctions de Genz, pour un nombre de subdivision de l'intervalle $[0, 1]$ $n = 4$ et pour différentes valeurs des paramètres a et u . Les erreurs relatives commises par les formules de Trapèze composite, Simpson composite, IMCC1 et IMCC2 sont notées respectivement \mathbf{E}_{TrG} , \mathbf{E}_{SiG} , \mathbf{E}_{IMCC1} , \mathbf{E}_{IMCC2} . Pour chaque fonction test, les résultats, les erreurs et le temp d'exécution, sont résumés dans les tableaux (3.1)-(3.6).

Exemple 3.3.1 $g_1(x) = \cos(2\pi u + ax)$, avec $\int_0^1 g_1(x)dx = \frac{\sin(2\pi u + a) - \sin(2\pi u)}{a}$

Exemple 3.3.2 $g_2(x) = \exp(-a|x - u|)$, avec $\int_0^1 g_2(x)dx = \frac{2 - \exp(-au) - \exp(a(u-1))}{a}$

Exemple 3.3.3 $g_3(x) = \frac{1}{a^{-2} + (x-u)^{-2}}$, avec $\int_0^1 g_3(x)dx = a^2(1 - a \arctan(\frac{1-u}{a})) + \arctan(\frac{-u}{au})$

Exemple 3.3.4 $g_4(x) = (1 + ax)^{-2}$, avec $\int_0^1 g_4(x)dx = \frac{1}{a}(1 - \frac{1}{1+a})$

Exemple 3.3.5 $g_5(x) = \exp(-a^2(x - u)^2)$, avec $\int_0^1 g_5(x)dx = \frac{\sqrt{\pi}}{2a}(\text{erf}(a(1 - u)) + \text{erf}(au))$

Exemple 3.3.6 $g_6(x) := \begin{cases} g(x) = 0 & \text{si } x > u \\ g(x) = e^{ax} & \text{sinon.} \end{cases}$, avec $\int_0^1 g_6(x)dx = \frac{\exp(au) - 1}{a}$

Remarque 3.3.1 *A partir des résultats observés sur les tableaux (3.1)-(3.6) et comme cela est prévisible, les résultats de l'intégration numérique sont meilleurs quand la fonction à intégrer est approchée par un polynôme de degré 2.*

En comparant les formules de quadrature issues de l'approximation par un polynôme de degré 1; E_{Trg} et E_{IMCC1} , nous remarquons que la formule (IMCC1) donne de meilleurs résultats que ceux donnés par la formule de Trapèze et ceci pour tous les couples de paramètre (a, u) choisis. Pour certaines fonctions de base, nous constatons un gain en précision de l'ordre de 10^{-2} comme cela est visible sur les tableaux (3.1), (3.2) et (3.5).

Par contre les formules de quadrature issues de l'approximation par un polynôme de degré 2, c'est à dire (IMCC2) et Simpson sont généralement du même ordre de précision. Cependant, pour certaines fonctions et pour certains jeu des paramètres $(a = 1, u = 1)$, l'erreur commise par la méthode (IMCC2) est améliorée d'une décimale voir les tableaux (3.1), (3.3) et (3.6) et peut même s'améliorer de 3 décimales en se référant au tableau (3.5).

Pour constater la convergence numérique des méthodes (IMCC1) et (IMCC2), nous allons mettre en évidence la variation des erreurs relatives en fonction du nombre de subdivision de l'intervalle $[0, 1]$.

Pour le couple de paramètre $(a = 10, u = \frac{1}{4})$, les résultats des tests effectués sont résumés dans le tableau (3.7). Les graphes du logarithme des erreurs relatives pour les différentes fonctions et les différentes méthodes sont représentés par la figure (3.2).

Remarque 3.3.2 *Nous constatons, à partir des résultats du tableau (3.7) et des graphes de la figure (3.2), que les erreurs décroissent quand le nombre de subdivision de l'intervalle $[0, 1]$ croit; c'est ce qui traduit la convergence numérique. Cependant, nous remarquons une instabilité numérique de la méthode (IMCC2), particulièrement pour les fonctions de base Discontinuous function et Product Peak function. Pour les autres types de fonctions de base, les méthodes (IMCC) sont plus rapides que le méthode de Newton-Côte composite.*

3.4 Les tests en deux dimensions

Comme dans le cas des fonctions d'une variable, les quatres formules de cubature composites (Trapèze, Simpson, IMCDC1, IMCDC2) sont testées pour certaines fonctions de Genz, pour un nombre de subdivision de l'intervalle $[0, 1]^2$ $n = 8$ et pour différentes valeurs des vecteurs de paramètres (a_1, a_2) et (u_1, u_2) . Les erreurs relatives commises par ces formules sont notées respectivement \mathbf{E}_{TrDG} , \mathbf{E}_{SiDG} , \mathbf{E}_{IMCDC1} , \mathbf{E}_{IMCDC2} . Pour chaque fonction test, les résultats des erreurs et le temp d'exécution, sont résumés dans les tableaux (3.8) et (3.9).

Exemple 3.4.1 $h_1(x, y) = \exp(-(a_1|x - u_1| + a_2|y - u_2|))$, avec

$$\int_0^1 \int_0^1 h_1(x, y) dx dy = \prod_{i=1}^2 \frac{2 - \exp(-a_i u_i) - \exp(a_i(u_i - 1))}{a_i}$$

Exemple 3.4.2 $h_2(x, y) = \exp(-a_1^2(x - u_1)^2 - a_2^2(y - u_2)^2)$, avec $\int_0^1 \int_0^1 h_2(x, y) dx dy = \frac{\sqrt{\pi}}{2a_1} (\operatorname{erf}(a_1(1 - u_1)) + \operatorname{erf}(a_1 u_1)) * \frac{\sqrt{\pi}}{2a_2} (\operatorname{erf}(a_2(1 - u_2)) + \operatorname{erf}(a_2 u_2))$

Remarque 3.4.1 *Nous constatons que les mêmes remarques faites en dimension 1 sont toujours valables en dimension 2. Mais nous observons une perte de précision qui est due à la dimension. Ce phénomène est connu en analyse numérique par le "fléau de la dimension".*

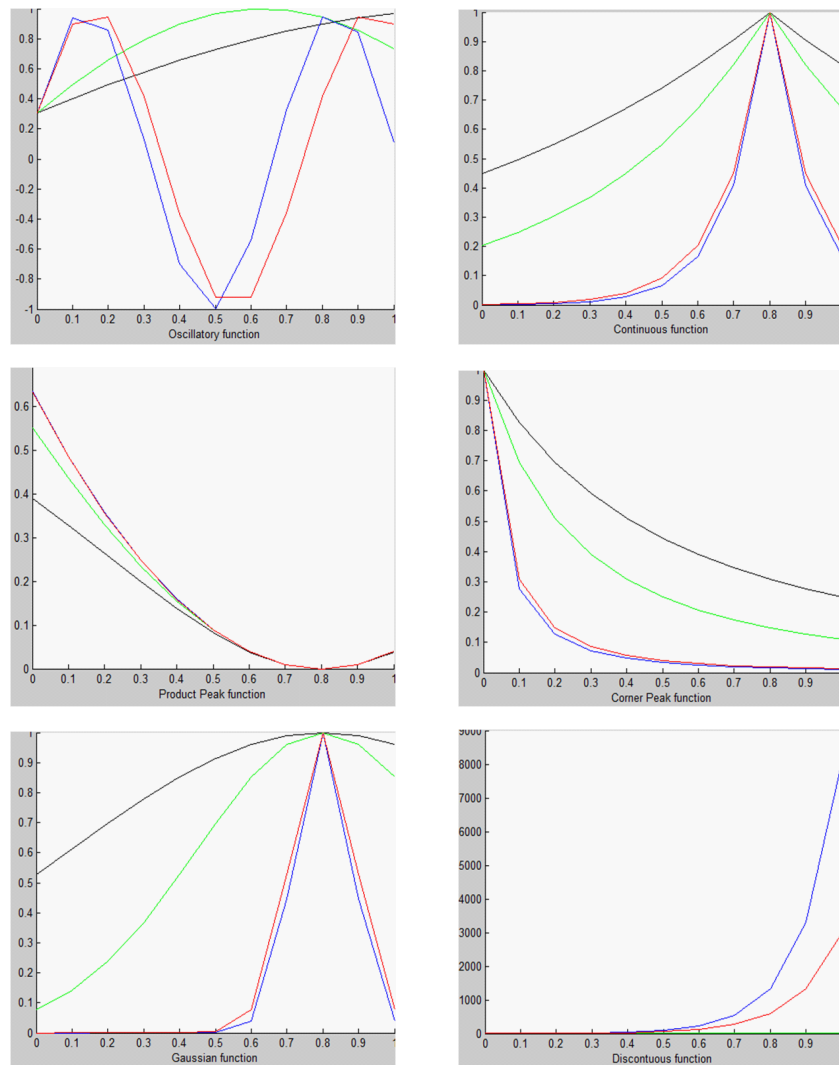


FIG. 3.1 – Allure des différentes fonctions test, avec une valeur de $u=0.8$ et différentes valeurs de $a=1$ (ligne noire), $a=2$ (ligne verte), $a=9$ (ligne blue), $a=8$ (ligne rouge)

(a, u)	$(3, 0)$	$(10, \frac{1}{4})$	$(\frac{1}{2}, \frac{1}{2})$	$(\sqrt{2}, \frac{3}{4})$	$(1, 1)$	$(\frac{5}{6}, \frac{5}{4})$
\mathbf{I}_{EX}	0.0470	-0.1839	-0.9588	0.5968	0.8414	-0.3931
\mathbf{E}_{TrG}	0.0473	0.58465	0.0013	0.0104	0.0052	0.0036
\mathbf{T}_{TrG}	0.0115 <i>sec</i>	0.0106 <i>sec</i>	0.0103 <i>sec</i>	0.0101 <i>sec</i>	0.0115 <i>sec</i>	0.0107 <i>sec</i>
\mathbf{E}_{IMCC1}	0.0236	0.28403	$6.5116e-4$	0.0052	0.0026	0.0018
\mathbf{T}_{IMCC1}	0.0289 <i>sec</i>	0.0287 <i>sec</i>	0.0249 <i>sec</i>	0.0287 <i>sec</i>	0.0297 <i>sec</i>	0.0274 <i>sec</i>
\mathbf{E}_{SiG}	$1.1173e-4$	0.01657	$8.4810e-8$	$5.4455e-6$	$1.3588e-6$	$6.5494e-7$
\mathbf{T}_{SiG}	0.0084 <i>sec</i>	0.0092 <i>sec</i>	0.0096 <i>sec</i>	0.0091 <i>sec</i>	0.0097 <i>sec</i>	0.0087 <i>sec</i>
\mathbf{E}_{IMCC2}	$2.0255e-5$	0.00295	$1.5398e-8$	$9.8851e-7$	$2.4671e-7$	$1.1891e-7$
\mathbf{T}_{IMCC2}	0.4352 <i>sec</i>	0.0299 <i>sec</i>	0.0298 <i>sec</i>	0.0311 <i>sec</i>	0.0290 <i>sec</i>	0.0689 <i>sec</i>

TAB. 3.1 – Tableau associé à la fonction oscillatoire

(a, u)	$(3, 0)$	$(10, \frac{1}{4})$	$(\frac{1}{2}, \frac{1}{2})$	$(\sqrt{2}, \frac{3}{4})$	$(1, 1)$	$(\frac{5}{6}, \frac{5}{4})$
\mathbf{I}_{EX}	0.31674	0.19174	0.88479	0.6728	0.6321	0.4986
\mathbf{E}_{TrG}	0.0464	0.4735	0.0013	0.0103	0.0052	0.1088
\mathbf{T}_{TrG}	0.0110 <i>sec</i>	0.0117 <i>sec</i>	0.0111 <i>sec</i>	0.0110 <i>sec</i>	0.01143 <i>sec</i>	0.0100 <i>sec</i>
\mathbf{E}_{IMCC1}	0.0232	0.2424	$6.5091e-4$	0.0052	0.0026	0.1068
\mathbf{T}_{IMCC1}	0.0282 <i>sec</i>	0.0288 <i>sec</i>	0.0293 <i>sec</i>	0.0292 <i>sec</i>	0.0290 <i>sec</i>	0.02662 <i>sec</i>
\mathbf{E}_{SiG}	$1.0805e-4$	0.0113	$8.4731e-8$	$5.4052e-6$	$1.3538e-6$	0.1048
\mathbf{T}_{SiG}	0.0086 <i>sec</i>	0.0082 <i>sec</i>	0.0091 <i>sec</i>	0.0085 <i>sec</i>	0.0091 <i>sec</i>	0.0084 <i>sec</i>
\mathbf{E}_{IMCC2}	$1.9653e-5$	0.0021	$1.5385e-8$	$9.8189e-7$	$2.4588e-7$	0.1048
\mathbf{T}_{IMCC2}	0.0287 <i>sec</i>	0.0277 <i>sec</i>	0.03042 <i>sec</i>	0.0286 <i>sec</i>	0.0298 <i>sec</i>	0.02603 <i>sec</i>

TAB. 3.2 – Tableau associé à la fonction continue

(a, u)	$(3, 0)$	$(10, \frac{1}{4})$	$(\frac{1}{2}, \frac{1}{2})$	$(\sqrt{2}, \frac{3}{4})$	$(1, 1)$	$(\frac{5}{6}, \frac{5}{4})$
\mathbf{I}_{EX}	0.3127	0.1453	0.0536	0.1259	0.2146	0.2943
\mathbf{E}_{TrG}	0.0270	0.0710	0.0484	0.0574	0.0121	0.0033
\mathbf{T}_{TrG}	0.0108 <i>sec</i>	0.0112 <i>sec</i>	0.0112 <i>sec</i>	0.0100 <i>sec</i>	0.0107 <i>sec</i>	0.0498 <i>sec</i>
\mathbf{E}_{IMCC1}	0.0135	0.0355	0.0242	0.0287	0.0060	0.0016
\mathbf{T}_{IMCC1}	0.0464 <i>sec</i>	0.0283 <i>sec</i>	0.0283 <i>sec</i>	0.0287 <i>sec</i>	0.0288 <i>sec</i>	0.0243 <i>sec</i>
\mathbf{E}_{SiG}	$6.7694e-6$	$2.1923e-6$	$2.7989e-5$	$5.4310e-5$	$1.7606e-7$	$2.9354e-5$
\mathbf{T}_{SiG}	0.0096 <i>sec</i>	0.0095 <i>sec</i>	0.0098 <i>sec</i>	0.0059 <i>sec</i>	0.0047 <i>sec</i>	0.0714 <i>sec</i>
\mathbf{E}_{IMCC2}	$3.0074e-6$	$9.7433e-7$	$1.0722e-5$	$2.4099e-5$	$6.9559e-8$	$1.2973e-5$
\mathbf{T}_{IMCC2}	0.0296 <i>sec</i>	0.0300 <i>sec</i>	0.0298 <i>sec</i>	0.0299 <i>sec</i>	0.0254 <i>sec</i>	0.2327 <i>sec</i>

TAB. 3.3 – Tableau associé à la fonction de pic de produit

a	3	10	$\frac{1}{2}$	$\sqrt{2}$	1	$(\frac{5}{6}, \frac{5}{4})$
\mathbf{I}_{EX}	0.2500	0.0909	0.6666	0.4142	0.5000	0.5454
\mathbf{E}_{TrG}	0.1124	0.7253	0.0054	0.0322	0.0179	0.0132
\mathbf{T}_{TrG}	0.0102 <i>sec</i>	0.0105 <i>sec</i>	0.0103 <i>sec</i>	0.0101 <i>sec</i>	0.0104 <i>sec</i>	0.0605 <i>sec</i>
\mathbf{E}_{IMCC1}	0.0574	0.3989	0.0027	0.0162	0.0090	0.0066
\mathbf{T}_{IMCC1}	0.0282 <i>sec</i>	0.0280 <i>sec</i>	0.0280 <i>sec</i>	0.0277 <i>sec</i>	0.0249 <i>sec</i>	0.0282 <i>sec</i>
\mathbf{E}_{SiG}	0.0024	0.0726	$5.2214e - 6$	$1.9842e - 4$	$5.9758e - 5$	$3.16272e - 5$
\mathbf{T}_{SiG}	0.0044 <i>sec</i>	0.0040 <i>sec</i>	0.0048 <i>sec</i>	0.0047 <i>sec</i>	0.0097 <i>sec</i>	0.0105 <i>sec</i>
\mathbf{E}_{IMCC2}	0.0011	0.0381	$2.3244e - 6$	$8.9182e - 5$	$2.6718e - 5$	$1.41167e - 5$
\mathbf{T}_{IMCC2}	0.0364 <i>sec</i>	0.0289 <i>sec</i>	0.0291 <i>sec</i>	0.0269 <i>sec</i>	0.0289 <i>sec</i>	0.3713 <i>sec</i>

TAB. 3.4 – Tableau associé à la fonction de crête de coin

(a, u)	$(3, 0)$	$(10, \frac{1}{4})$	$(\frac{1}{2}, \frac{1}{2})$	$(\sqrt{2}, \frac{3}{4})$	$(1, 1)$	$(\frac{5}{6}, \frac{5}{4})$
\mathbf{I}_{EX}	0.2954	0.1772	0.9795	0.7828	0.7468	0.6674
\mathbf{E}_{TrG}	$3.0744e - 5$	0.4148	0.0025	0.0124	0.0051	0.0019
\mathbf{T}_{TrG}	0.0627 <i>sec</i>	0.0108 <i>sec</i>	0.0116 <i>sec</i>	0.0093 <i>sec</i>	0.0113 <i>sec</i>	0.0115 <i>sec</i>
\mathbf{E}_{IMCC1}	$1.6348e - 5$	0.1403	0.0012	0.0062	0.00257	$9.8948e - 4$
\mathbf{T}_{IMCC1}	0.0280 <i>sec</i>	0.0284 <i>sec</i>	0.0263 <i>sec</i>	0.0283 <i>sec</i>	0.0292 <i>sec</i>	0.02608 <i>sec</i>
\mathbf{E}_{SiG}	$1.9518e - 6$	0.1341	$9.3910e - 7$	0.0062	0.0025	$1.40353e - 6$
\mathbf{T}_{SiG}	0.0015 <i>sec</i>	0.0090 <i>sec</i>	0.0054 <i>sec</i>	0.0285 <i>sec</i>	0.0278 <i>sec</i>	0.0087 <i>sec</i>
\mathbf{E}_{IMCC2}	$9.0441e - 7$	0.0521	$4.1717e - 7$	$9.9066e - 6$	$1.1834e - 6$	$6.2162e - 7$
\mathbf{T}_{IMCC2}	0.0475 <i>sec</i>	0.0284 <i>sec</i>	0.0263 <i>sec</i>	0.0284 <i>sec</i>	0.0273 <i>sec</i>	0.0285 <i>sec</i>

TAB. 3.5 – Tableau à la fonction gaussienne

(a, u)	$(10, \frac{1}{4})$	$(\frac{1}{2}, \frac{1}{2})$	$(\sqrt{2}, \frac{3}{4})$	$(1, 1)$	$(\frac{5}{6}, \frac{5}{4})$
\mathbf{I}_{EX}	1.11825	0.56805	1.33521	1.71828	2.20072
\mathbf{E}_{TrG}	0.03234	$3.25410e - 4$	0.00585	0.00520	0.00565
\mathbf{T}_{TrG}	0.01073 <i>sec</i>	0.01071 <i>sec</i>	0.01046 <i>sec</i>	0.01157 <i>sec</i>	0.06450 <i>sec</i>
\mathbf{E}_{IMCC1}	0.01620	$1.62752e - 4$	0.00293	0.00260	0.00282
\mathbf{T}_{IMCC1}	0.02737 <i>sec</i>	0.02638 <i>sec</i>	0.03283 <i>sec</i>	0.02769 <i>sec</i>	0.02700 <i>sec</i>
\mathbf{E}_{SiG}	$5.23722e - 5$	$5.29758e - 9$	$1.71303e - 6$	$1.35382e - 6$	$1.59370e - 6$
\mathbf{T}_{SiG}	0.00405 <i>sec</i>	0.00142 <i>sec</i>	0.00405 <i>sec</i>	0.00632 <i>sec</i>	0.03920 <i>sec</i>
\mathbf{E}_{IMCC2}	$2.33066e - 5$	$2.35464e - 9$	$7.61520e - 7$	$6.01821e - 7$	$7.084693e - 7$
\mathbf{T}_{IMCC2}	0.02774 <i>sec</i>	0.02532 <i>sec</i>	0.02765 <i>sec</i>	0.02731 <i>sec</i>	0.33056 <i>sec</i>

TAB. 3.6 – Tableau la fonction discontinue

<i>nombre des points</i>		$n = 4$	$n = 8$	$n = 16$	$n = 32$	$n = 64$
$g_1(x) = \cos(2\pi u + ax)$	E_{TrG}	0.5846	0.1337	0.03276	0.0081	0.0020
	E_{IMCC1}	0.2840	0.06642	0.01635	0.0040	0.0010
	E_{Si_mG}	0.01657	8.8882e-4	5.36042e-5	3.3210e-6	2.0711e-7
	E_{IMCC2}	0.00295	1.6066e-4	9.72204e-6	5.9589e-7	1.3371e-7
$g_2(x) = \exp(-a x - u)$	E_{TrG}	0.47356	0.1269	0.03234	0.0081	0.0020
	E_{IMCC1}	0.24248	0.06387	0.01619	0.0040	0.0010
	E_{Si_mG}	0.01139	8.09844e-4	5.23722e-5	3.3017e-6	2.0681e-7
	E_{IMCC2}	0.00210	1.4772e-4	9.5211e-6	6.0002e-7	3.6664e-8
$g_3(x) = \frac{1}{a^{-2} + (x-u)^{-2}}$	E_{TrG}	0.07104	0.01776	0.00444	0.00110	2.7749e-4
	E_{IMCC1}	0.03552	0.00888	0.00222	5.5499e-4	1.3874e-4
	E_{Si_mG}	2.19238e-6	1.36967e-7	8.55963e-9	5.3499e-10	3.3470e-11
	E_{IMCC2}	3.98099e-7	2.48982e-8	1.6012e-9	8.8563e-9	9.4718e-8
$g_4(x) = (1 + ax)^{-2}$	E_{TrG}	0.72534	0.23582	0.067036	0.0175	0.0044
	E_{IMCC1}	0.3989	0.12329	0.03405	0.0088	0.0022
	E_{Si_mG}	0.0726	0.01077	0.00107	8.0702e-5	5.3486e-6
	E_{IMCC2}	0.0157	0.00211	1.9972e-4	1.4771e-5	9.6982e-7
$g_5(x) = \exp(-a^2(x - u)^2)$	E_{TrG}	0.41484	0.00313	1.57427e-4	4.2986e-5	1.0996e-5
	E_{IMCC1}	0.14037	9.4011e-4	8.1133e-5	2.16599e-5	5.5088e-6
	E_{Si_mG}	0.13410	0.00125	4.8402e-6	3.3283e-7	2.12662e-8
	E_{IMCC2}	0.02166	2.0031e-4	8.9049e-7	6.0717e-8	2.2277e-9
$g_6(x) := \begin{cases} 0 & \text{si } x > u \\ e^{ax} & \text{sinon.} \end{cases}$	E_{TrG}	0.03234	0.00812	0.00203	5.0857e-4	1.2715e-4
	E_{IMCC1}	0.01619	0.00406	0.00101	2.5429e-4	6.3577e-5
	E_{Si_mG}	5.2372e-5	3.30176e-6	2.0681e-7	1.2932e-8	8.0840e-10
	E_{IMCC2}	9.5211e-6	5.996e-7	3.778e-8	3.7785e-8	1.5959e-7

TAB. 3.7 – Tableau des erreurs associé aux fonctions de Genz

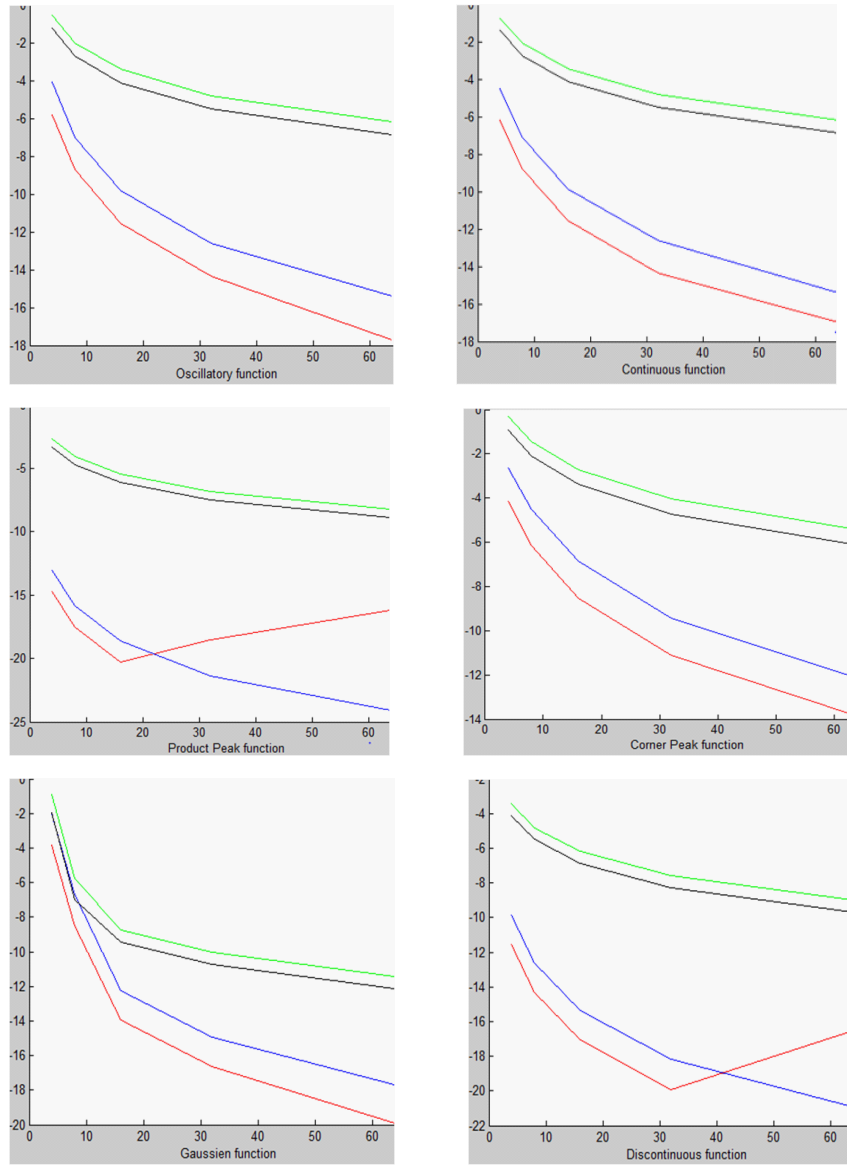


FIG. 3.2 – L’allure des erreurs des fonctions de Genz Trapèze généralisée (tiré plein vert), moindre carré généralisée d’ordre 1 (tiré plein noire), Simpson généralisée (tiré plein bleue) et moindre carré généralisée (tiré plein rouge)

(\mathbf{a}_1, u_1)	(2, 0.8)	$(\frac{1}{2}, \frac{1}{2})$	(0.1, 0.8)	(3, 0.9)
(\mathbf{a}_2, u_2)	(1, 0.4)	(2, 0.6)	(0.2, 0.7)	(1, 0.5)
\mathbf{I}_{EX}	0.44033	0.55277	0.91308	0.31267
\mathbf{E}_{TrDG}	0.02838	0.02577	0.00289	0.060697
\mathbf{T}_{TrDG}	0.021351 <i>sec</i>	0.02020 <i>sec</i>	0.02132 <i>sec</i>	0.01853 <i>sec</i>
\mathbf{E}_{IMCDC1}	0.01612	0.01023	0.00170	0.02409
\mathbf{T}_{IMCDC1}	0.13275 <i>sec</i>	0.04900 <i>sec</i>	0.05114 <i>sec</i>	0.04782 <i>sec</i>
\mathbf{E}_{SiDG}	0.00379	0.00529	$5.25292e - 004$	0.01232
\mathbf{T}_{SiDG}	0.02254 <i>sec</i>	0.02388 <i>sec</i>	0.02340 <i>sec</i>	0.02216 <i>sec</i>
\mathbf{E}_{IMCDC2}	0.00145	0.00199	$1.97122e - 004$	0.00467
\mathbf{T}_{IMCDC2}	0.11466 <i>sec</i>	0.08439 <i>sec</i>	0.08302 <i>sec</i>	0.08774 <i>sec</i>

TAB. 3.8 – Tableau associé à la fonction continue en 2 D

(\mathbf{a}_1, u_1)	(2, 0.8)	$(\frac{1}{2}, 0.5)$	(0.1, 0.8)	(3, 0.9)
(\mathbf{a}_2, u_2)	(1, 0.4)	(2, 0.6)	(0.2, 0.7)	(1, 0.5)
\mathbf{I}_{EX}	0.56943	0.71723	0.99337	0.36206
\mathbf{E}_{TrDG}	0.02430	0.02272	$5.16453e - 004$	0.03192
\mathbf{T}_{TrDG}	0.49194 <i>sec</i>	0.02152 <i>sec</i>	0.02072 <i>sec</i>	0.02045 <i>sec</i>
\mathbf{E}_{IMCDC1}	0.01215	0.01135	$2.58224e - 004$	0.01584
\mathbf{T}_{IMCDC1}	0.04964 <i>sec</i>	0.05351 <i>sec</i>	0.04927 <i>sec</i>	0.05311 <i>sec</i>
\mathbf{E}_{SiDG}	$5.99484e - 005$	$4.58513e - 005$	$2.71683e - 008$	$3.51308e - 004$
\mathbf{T}_{SiDG}	0.02565 <i>sec</i>	0.02236 <i>sec</i>	0.02245 <i>sec</i>	0.02397 <i>sec</i>
\mathbf{E}_{IMCDC2}	$3.01873e - 007$	$2.08666e - 005$	$1.27345e - 008$	$1.05342e - 004$
\mathbf{T}_{IMCDC2}	0.08009 <i>sec</i>	0.07828 <i>sec</i>	0.08528 <i>sec</i>	0.82269 <i>sec</i>

TAB. 3.9 – Tableau associé à la fonction gaussienne en 2 D

Conclusion et perspective

Dans ce travail, nous nous sommes appuyés sur l'approximation au sens des moindres carrés pour élaborer une nouvelle formule de quadrature (**IMC**) et une autre de cubature (**IMCD**).

Ces deux formules numériques ont été validées en les testant sur la base de données des fonctions de Genz.

Numériquement, ces méthodes ont été comparées avec les méthodes des Newton-Côtes composites (Trapèze et Simpson). Il a été constaté que les formules (**IMC**) et (**IMCD**) sont aussi précises que les méthodes de Newton-Côtes voir meilleures dans beaucoup de cas.

Comme perspectives, nous proposons

- d'établir une estimation de l'erreur
- comparer cette méthode avec la méthode de Monté-Carlo notamment en dimension 2.
- voir que devient la formule (**IMC**) et (**IMCD**) quand l'approximation aux sens des moindres carrés est basée sur les fonctions e^{nx} et les fonctions circulaires $\cos(nx)$ et $\sin(nx)$.

Bibliographie

- [1] A. Genz, Testing multidimensional integration routines. In Proc. of international conference on Tools, methods and languages for scientific and engineering computation, pages 81-94. Elsevier North-Holland, Inc., 1984.
- [2] B. Demidovich.I.Maron, Eléments de calcul numérique, Editions MIR Moscou, 1967.
- [3] J. F. Epperson, On the Runge exemple. Amer. Math.Monthly, 94(4) : 329-341, 1987.
- [4] J. Berntsen, On the numerical calculation of multidimensional integrals appearing in the theorie of underwater acoustics, Series C : mathematical and physical sciences. Vol. 357, 1992.
- [5] N. Dumont, Méthodes numériques et modèles réduit de chimie tabulée pour la propagation d'incertitudes de cintique chimique. Thèse de Doctorat. Universite Paris-Saclay. 2019.
- [6] S. Derradji, Analyse Numérique 1, polycopé réimprimé par l'office des publications universitaires (OPU) Alger, 1990.
- [7] V.Sobolev, Cours d'astrophysique théorique, Editions MIR Moscou, traduction française par V. Polonski, 1990.

Annexe

Cette partie concerne les codes Matlab utilisés.

Les programme en une dimension

– Méthode de Trapèze

```
function I = trpz( a ,b,f )  
y(1)=f(a);  
y(2)=f(b);  
h=b-a;  
I=(h/2)*(y(1)+y(2))  
end
```

– Méthode de Trapèze généralisée

```
function [ It ] =trpz_general(a,b,j,f )  
M=2^(j-1)  
h=(b-a)/M;  
x0=a;  
x(1)=a;  
y(1)=f(x(1));  
It(k)=0  
for i=2 :M+1  
x(i)=x0+(i-1)*h;  
y(i)=f(x(i));  
%%rappel la méthode du Trapèze simple  
I = trpz( x(i-1) ,x(i),f );  
It=It(k)+I;  
end  
end
```

– Méthode de Simpson

```

function [ I ] =simpson ( a ,b ,h ,f )
y0=f(a);
y1=f((a+b)/2);
y2=f(b);
I=h/3*(y0+4*y1+y2);
end

```

– **Méthode de Simpson généralisée**

```

function [ Is ] = simpson_general( a,b,f,j )
M=2^(j-1)
h=(b-a)/M;
x=[a :h :b];
hh=h/2;
for i=1 :M
%%rappel la méthode du Simpson simple
[ I(i) ] =simpson ( x(i) ,x(i+1) ,hh ,f );
endI
Is=sum(I);
end

```

– **Méthode IMC1(formule de quadrature par une approximation d'ordre 1)**

```

function [X,I] =moindre_carre_dordre1(a,b,f)
%%les points pour déterminer le polynôme d'ordre 1
x=[a,a+(b-a)/2,b]
for i=1 :length(x)
y(i)=f(x(i));
end
y=[y(1),y(2),y(3)]
%%polynôme de degré 1
N=length(x);
A=[N,sum(x);sum(x),sum(x.^2)]
B=[sum(y);sum(x.*y)]
X=inv(A)*B
%%intégrale du polynôme de degré 1
I=X(1)*(b-a)+X(2)/2*(b^2-a^2)
end

```

– **Méthode de IMC2(formule de quadrature par une approximation d'ordre 2)**

```

function [X,I] =moindre_carre_dordre2(a,b,f)
%%les points pour déterminer le polynôme de degré 2
%x=[a,a+(b-a)/3,a+2*(b-a)/3,a+3*(b-a)/3];
x=[a,a+(b-a)/7,a+2*(b-a)/7,a+3*(b-a)/7,a+4*(b-a)/7,a+5*(b-a)/7,a+6*(b-a)/7,a+7*(b-a)/7];
for i=1 :length(x)
y(i)=f(x(i));
end

```

```
% y=[y(1),y(2),y(3),y(4)];
y=[y(1),y(2),y(3),y(4),y(5),y(6),y(7),y(8)];
%%polynôme de degré 2
N=length(x);
A=[N,sum(x),sum(x.^2);sum(x),sum(x.^2),sum(x.^3);sum(x.^2),sum(x.^3),sum(x.^4)];
B=[sum(y);sum(x.*y);sum(x.^2.*y)];
X=inv(A)*B
%%sol de degré 2
I=X(1)*(b-a)+X(2)/2*(b^2-a^2)+X(3)/3*(b^3-a^3);
end
```

– **Méthode de IMCC1(formule de quadrature composite par une approximation d'ordre 1)**

```
function [ Imd1 ] =m_car_genr_dordre1 ( a,b,j,f )
%j fixée
a(1)=a;
%%decomposition lintervalle
%M=2^(j-1)
h=(b-a)/M;
a=[a :h :b];
for i=1 :M
%%rappel la méthode IMC1
[X,I] =moindre_carre_dordre1(a(i),a(i+1),f);
end
Im=sum(I);
end
```

– **Méthode de IMCC2(formule de quadrature composite par une approximation d'ordre 2)**

```
function [ Im ] =m_car_genr_ordre2 ( a,b,j,f )
%j fixée
a(1)=a;
M=2^(j-1)
h=(b-a)/M;
a=[a :h :b];
for i=1 :M
%%rappel la méthode IMC2
[X,I(i)] =moindre_carre_dordre2(a(i),a(i+1),f);
end
Im=sum(I);
end
```

Les programme en deux dimensions**– Méthode de Trapèze double**

```
function [ Itd ] = trap_double( f,a1,a2,b1,b2 )
Itd=((a2-a1)/2)*((b2-b1)/2)*(f(a1,b1)+f(a1,b2)+f(a2,b1)+f(a2,b2))
end
```

– Méthode de Simpson double

```
function [Isd ] = symp_double( f,a1,a2,b1,b2)
Isd=(((a2-a1)/6)*((b2-b1)/6))*(f(a1,b1)+f(a2,b1)+f(a1,b2)+f(a2,b2)+4*f((a1+a2)/2,b1)
+4*f(a1,(b1+b2)/2)+4*f(a2,(b1+b2)/2)+4*f((a1+a2)/2,b2)+16*f((a1+a2)/2,(b1+b2)/2))
end
```

– Méthode de Trapèze double généralisée

```
function [Itgd ] = trp_double_gnrl(f,a1,a2,b1,b2,j,l)
%j l fixée
Nx=2^(j-1)
Ny=2^(l-1)
hx=(a2-a1)/Nx
hy=(b2-b1)/Ny
Itgd=0
Px(1)=a1
Py(1)=b1
for i=2 :Nx+1
for k=2 :Ny+1
Px(i)=a1+(i-1)*hx
Py(k)=b1+(k-1)*hy
%%rappel la méthode Trapèze double
[ Itd ] = trap_double( f,Px(i-1),Px(i),Py(k-1),Py(k))
Itgd=Itgd+Itd
end
end
end
```

– Méthode de Simpson double généralisée

```
function [ Isgd ] = sumpson_double_gnrl( a1,a2,b1,b2,f,j,l)
%j j' fixée
Nx=2^(j-1)
Ny=2^(l-1)
hx=(a2-a1)/Nx
hy=(b2-b1)/Ny
Isgd=0
for i=2 :Nx+1
for k=2 :Ny+1
Px(i)=a1+(i-1)*hx
```

```

Py(k)=b1+(k-1)*hy
%% rappel la méthode Simpson double
[Isd ] = symp_double( f,Px(i-1),Px(i),Py(k-1),Py(k))
Isgd=Isgd+Isd
end
end
end

```

– **Méthode de IMCD1(formule de cubature par une approximation de degré 1)**

```

function [ I ] = moind_carre_double_dordre1(a1,a2,b1,b2,f)
%%les points de polynôme de degré 1
x=[a1,(a2+a1)/2,a2,a1,(a2+a1)/2,a2,a1,(a2+a1)/2,a2]
y=[b1,b1,b1,(b2+b1)/2,(b2+b1)/2,(b2+b1)/2,b2,b2,b2]
for i=1 :length(x)
T(i)=f(x(i),y(i));
end
T
N=length(x)
%%polynome degré1
A=[N,sum(x),sum(y),sum(x.*y);sum(x),sum(x.^2),sum(x.*y),sum(x.^2.*y);
sum(y),sum(x.*y),sum(y.^2),sum(x.*y.^2);sum(x.*y),sum(x.^2.*y),sum(x.*y.^2),sum(x.^2.*y.^2)];
B=[sum(T);sum(x.*T);sum(y.*T);sum(x.*y.*T)];
X=inv(A)*B
%%intégration du polynome degré 1
I=((a2-a1)/2)*((b2-b1)/2)*(4*X(1)+2*X(2)*(a1+a2)+2*X(3)*(b1+b2)+X(4)*(a1+a2)*(b1+b2))
end

```

– **Méthode de IMCD2(formule de cubature par une approximation de degré 2)**

```

function [ I ] = moind_carre_double_dordre2(a1,a2,b1,b2,f)
%%les points de degré 2
x=[a1,(a2+2*a1)/3,(a1+2*a2)/3,a2,a1,(a2+2*a1)/3,(a1+2*a2)/3,a2,a1,(a2+2*a1)/3,
(a1+2*a2)/3,a2,a1,(a2+2*a1)/3,(a1+2*a2)/3,a2]
y=[b1,b1,b1,b1,(2*b1+b2)/3,(2*b1+b2)/3,(2*b1+b2)/3,(2*b1+b2)/3,(b1+2*b2)/3,(b1+2*b2)/3,
(b1+2*b2)/3,b2,b2,b2,b2]
for i=1 :length(x)
T(i)=f(x(i),y(i));
end
T
N=length(x)
%%polynome degré2
A=[N,sum(x),sum(y),sum(x.*y),sum(x.^2),sum(y.^2),sum(x.^2.*y.^2),sum(x.^2.*y),
sum(x.*y.^2);sum(x),sum(x.^2),sum(x.*y),sum(x.^2.*y),sum(x.^3),sum(x.*y.^2),
sum(x.^3.*y.^2),sum(x.^3.*y),sum(x.^2.*y.^2);sum(y),sum(x.*y),sum(y.^2),sum(x.*y.^2),
sum(x.^2.*y),sum(y.^3),sum(x.^2.*y.^3),sum(x.^2.*y.^2),sum(x.*y.^3);sum(x.*y),
sum(x.^2.*y),sum(x.*y.^2),sum(x.^2.*y.^2),sum(x.^3.*y),sum(x.*y.^3),sum(x.^3.*y.^3),

```



```

sum(x.^3.*y.^2),sum(x.^2.*y.^3);sum(x.^2),sum(x.^3),sum(x.^2.*y),sum(x.^3.*y),
sum(x.^4),sum(x.^2.*y.^2),sum(x.^4.*y.^2),sum(x.^4.*y),sum(x.^3.*y.^2);sum(y.^2),
sum(x.*y.^2),sum(y.^3),sum(x.*y.^3),sum(x.^2.*y.^2),sum(y.^4),sum(x.^2.*y.^4),
sum(x.^2.*y.^3),sum(x.*y.^3);sum(x.^2.*y.^2),sum(x.^3.*y.^2),sum(x.^2.*y.^3),
sum(x.^3.*y.^3),sum(x.^4.*y.^2),sum(x.^2.*y.^4),sum(x.^4.*y.^4),sum(x.^4.*y.^3),
sum(x.^3.*y.^4);sum(x.^2.*y),sum(x.^3.*y),sum(x.^2.*y.^2),sum(x.^3.*y.^2),
sum(x.^4.*y),sum(x.^2.*y.^3),sum(x.^4.*y.^3),sum(x.^4.*y.^2),sum(x.^3.*y.^3);
sum(x.*y.^2),sum(x.^2.*y.^2),sum(x.*y.^3),sum(x.^2.*y.^3),sum(x.^3.*y.^2),
sum(x.*y.^4),sum(x.^3.*y.^4),sum(x.^3.*y.^3),sum(x.^2.*y.^4)];
B=[sum(T);sum(x.*T);sum(y.*T);sum(x.*y.*T);sum(x.^2.*T);sum(y.^2.*T);
sum(x.^2.*y.^2.*T);sum(x.^2.*y.*T);sum(x.*y.^2.*T)];
X=inv(A)*B
%%Intégration le polynôme de degré 2
I=X(1)*((b2-b1)*(a2-a1))+X(2)*((b2-b1)*((a2^2-a1^2)/2))+X(3)*((a2-a1)
*((b2^2-b1^2)/2))+X(4)*(((a2^2-a1^2)/2)*((b2^2-b1^2)/2))+X(5)*(((a2^3-a1^3)/3)*(b2-b1))
+X(6)*(((b2^3-b1^3)/3)*(a2-a1))+X(7)*(((a2^3-a1^3)/3)*((b2^3-b1^3)/3))
+X(8)*(((a2^3-a1^3)/3)*((b2^2-b1^2)/2))+X(9)*(((a2^2-a1^2)/2)*((b2^3-b1^3)/3))
end

```

Méthode de IMCDC1(formule de cubature composite par une approximation d'ordre1)

```

function [Imcgd ] =moindr_carre_double_genrl_dordre1( a1,a2,b1,b2,f,j,l)
%j j' fixée
Nx=2^(j-1)
Ny=2^(l-1)
hx=(a2-a1)/Nx
hy=(b2-b1)/Ny
Imcgd=0
Px(1)=a1
Py(1)=b1
for i=2 :Nx+1
for k=2 :Ny+1
Px(i)=a1+(i-1)*hx
Py(k)=b1+(k-1)*hy
%%rappel la méthode IMCD2
[ Imcd ] = moindr_carre_double_dordre1(Px(i-1),Px(i),Py(k-1),Py(k),f)
Imcgd=Imcgd+Imcd
end
end
end

```

Méthode de IMCDC2(formule de cubature composite par une approximation d'ordre2)

```

function [Imcgd ] =moindr_carre_double_genrl_dordre2( a1,a2,b1,b2,f,j,l)
%j j' fixée

```

```
Nx=2^(j-1)
Ny=2^(l-1)
hx=(a2-a1)/Nx
hy=(b2-b1)/Ny
Imcgd=0
Px(1)=a1
Py(1)=b1
for i=2 :Nx+1
for k=2 :Ny+1
Px(i)=a1+(i-1)*hx
Py(k)=b1+(k-1)*hy
%%rappel la méthode IMCD2
[ Imcd ] = moind_carre_double_dordre2(Px(i-1),Px(i),Py(k-1),Py(k),f)
Imcgd=Imcgd+Imcd
end
end
end
```