

RÉPUBLIQUE DÉMOCRATIQUE POPULAIRE D'ALGÉRIE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed Khider – BISKRA

Faculté des Sciences Exactes, de la Nature et de la Vie

DEPARTEMENT DE MATHÉMATIQUES



Mémoire présenté pour obtenir le diplôme de

Master in “**Mathématiques**”

Option : Équations différentielles partielles et analyse numérique

par Mlle : **KABOUYA AICHA MARAM**

Titre :

**Les intégrales et les Dérivées de Bernstein : Application
pour la solutions d'équations différentielles.**

Devant le Jury :

Pr. KHELIL Naceur	Prof.	U.Biskra	Président
Dr. KACI Fatma	M.C.B	U.Biskra	Rapportuse
Dr. HOUAS Amrane	M.C.B	U.Biskra	Examineur

25/05/2025

Dédicace

*Al*hamdoulillah, qui m'a donné la capacité d'écrire et de réfléchir, la force de croire, la patience de poursuivre mon rêve jusqu'à son aboutissement, et la joie de lever les mains vers le ciel en disant : « Alhamdoulillah. »

Je dédie ce mémoire à :

Mes chers parents, **Mabrouka** et **Tahar**, dont l'amour, le dévouement et les sacrifices ont été le fondement de ma réussite académique. Votre soutien inconditionnel et vos encouragements constants m'ont donné la force de surmonter les obstacles. Je vous suis profondément reconnaissante ; ce succès est le fruit de vos efforts et de votre présence constante à mes côtés. et cette réussite est la vôtre, entre les mains de votre fille dévouée.

À mes frères et sœurs, **Imen**, **Sara** et **Ibrahim**, votre soutien moral a été ma bouée de sauvetage dans les moments difficiles. Je vous suis infiniment reconnaissante pour tout ce que vous m'avez apporté. Je vous souhaite plein de succès et de bonheur, et j'espère vous voir atteindre les plus hauts sommets.

À ma sœur **Ghofrane**, compagne de mes jours difficiles à l'université, ma bien-aimée, mon pilier, ma confidente et mon soutien le plus fidèle à chaque étape. Merci pour ta présence constante. Que cette réussite soit notre joie partagée.

À mes chères amies **Noura**, **Sana**, **Hadil** et **Nardjes**, merci du fond du cœur pour tout ce que vous avez fait pour moi. Votre rôle dans ma vie a été précieux, et je vous en suis profondément reconnaissante.

À mon fiancé bien-aimé **Badis**, mon pilier fort, mon soutien moral et pratique. Il a grandement contribué à l'accomplissement de ce mémoire. Je lui souhaite tout le succès et l'épanouissement dans sa vie.

Remerciement

Tout d'abord, je remercie Dieu tout-puissant pour son aide et ses faveurs qui m'ont permis de mener à bien ce travail. Je tiens à exprimer mes sincères remerciements et ma reconnaissance à mon superviseur, Mme **KACI Fatma**, pour m'avoir suggéré l'un des thèmes les plus importants et pour ses précieux conseils, son soutien continu et sa gentillesse, ainsi que pour le temps précieux qu'elle a consacré à cette recherche. Son enthousiasme et son intérêt ont joué un rôle majeur dans la réalisation de ce travail.

Je voudrais également exprimer ma sincère gratitude aux membres du comité de discussion, **M. KHELIL Naceur** et **M. HOUAS Amrane**, pour leur gentillesse dans l'évaluation de ce travail et pour l'honneur de leur présence et de leur précieuse contribution scientifique.

Je tiens également à exprimer ma sincère gratitude et mon appréciation à tous **les professeurs du département de mathématiques** pour tous les efforts qu'ils ont déployés pour nous enseigner et nous guider. Sans eux, nous ne serions pas là où nous sommes aujourd'hui.

Table des matières

Dédicace	i
Remerciement	ii
Table des matières	iii
Table des figures	v
Introduction	2
1 Polynômes de Bernstein	3
1.1 Introduction	3
1.2 Définition des polynômes de Bernstein	3
1.3 Propriétés des polynômes de Bernstein	4
1.4 Polynômes de Bernstein généralisés	5
1.5 Dérivées des polynômes de Bernstein	6
1.6 Intégrales des polynômes de Bernstein	8
1.6.1 Propriétés intégrales des polynômes de Bernstein	8
1.7 Théorème d'approximation de Weierstrass	9
2 Méthodes numériques basées sur les polynômes de Bernstein pour les équations différentielles	11
2.1 Introduction	11
2.2 Équations Différentielles	11
2.2.1 Types d'équations différentielles	12
2.2.2 Équation différentielle ordinaire linéaire	12
2.3 Méthode de collocation de Bernstein	13
2.3.1 Formulation du problème	13
2.3.2 Les étapes de la méthode collocation de Bernstein	14
2.4 Méthode de Bernstein–Petrov–Galerkin	15
2.4.1 Formulation du problème	15
2.4.2 Les étapes de la méthode de BPG	15
3 Applications Numériques	18
3.1 Introduction	18
3.2 Erreur d'approximation	18
3.3 Applications de la Méthode de collocation de Bernstein	19
3.3.1 Exemple 1	19
3.3.2 Exemple 2 : Étude comparative	22

3.4	Application de la Méthode de Bernstein–Petrov–Galerkin	24
3.4.1	Problème aux limites homogènes	24
3.4.2	Comparaison avec le solveur <code>bvp4c</code>	27
3.5	Étude comparative entre la méthode de collocation de Bernstein et la méthode BPG	29
	Conclusion	33
	Bibliographie	34

Table des figures

1.1	Polynômes de base de Bernstein pour $n = 4$	5
3.1	Solution exacte, solution approchée pour $N = 5$ et erreur absolue de l'exemple 1.	22
3.2	Comparaison des méthodes numériques	24
3.3	Solution exacte, approximation BPG ($N = 6$) et erreur absolue.	27
3.4	Solution exacte, approximation par bpv4c et approximation par BPG ($N = 8$).	29
3.5	Solution exacte, approximation par collocation de Bernstein et approximation par BPG ($N = 8$).	32

Notation

- $b_i^n(x)$: n-ième i-ième -degré du polynôme de Bernstein .
- $B_n(f; x)$: Une suite de polynômes de Bernstein .
- $\binom{n}{i}$: le coefficient binomial .
- E_{\max} :l'erreur maximale .
- E_{mean} :l'erreur moyenne .

Introduction

Le développement de méthodes numériques pour résoudre les équations différentielles est devenu de plus en plus crucial, en raison de l'absence d'une méthode universelle capable de s'adapter à tous les types de problèmes. Cette situation a suscité un intérêt croissant pour l'utilisation des polynômes, qui offrent une base numérique à la fois souple et performante pour l'approximation des solutions.

L'utilisation des polynômes en analyse numérique remonte à plusieurs siècles, avec les premières contributions notables de Newton et Lagrange aux 17 et 18 siècles. Ces derniers ont posé les fondations de certaines des méthodes d'approximation les plus connues basées sur les polynômes. Avec le développement de l'informatique et la nécessité de techniques numériques plus stables et flexibles, les chercheurs ont commencé à explorer des alternatives dotées de meilleures propriétés numériques, ouvrant ainsi la voie à l'utilisation de nouveaux types de polynômes, comme les polynômes de Bernstein.

Les polynômes de Bernstein ont été introduits pour la première fois en 1911 par le mathématicien Russe Sergueï Natanovitch Bernstein, dans le cadre de sa tentative de fournir une démonstration constructive du théorème d'approximation de Weierstrass. Ces polynômes sont ensuite devenus une pierre angulaire des techniques numériques modernes, en particulier dans des domaines comme l'analyse numérique et la modélisation graphique.

Dans ce contexte, les polynômes de Bernstein constituent des outils mathématiques fondamentaux ayant prouvé leur efficacité en analyse numérique, notamment pour l'approximation des fonctions et la résolution numérique des équations différentielles. Leur importance découle de leurs propriétés particulières, qui en font une alternative puissante aux polynômes classiques tels que ceux de Lagrange.

Dans ce mémoire, nous nous intéressons à l'étude des polynômes de Bernstein, en mettant en évidence certaines de leurs propriétés fondamentales, notamment leurs dérivées et leurs intégrales, qui jouent un rôle central dans la construction de méthodes numériques. Nous présentons deux approches numériques reposant sur ces polynômes. Le mémoire est divisé en trois chapitres.

Le premier chapitre aborde la définition mathématique formelle des polynômes de Bernstein et présente leurs propriétés fondamentales telles que la positivité, la symétrie, la partition de l'unité et la convergence vers les fonctions continues selon le théorème de Weierstrass. Il

comprend également une présentation de leurs dérivées et intégrales, tout en traitant de leur généralisation à différents domaines.

Le deuxième chapitre est consacré à l'utilisation des polynômes de Bernstein pour la construction d'outils numériques efficaces destinés à la résolution des équations différentielles. Il présente en détail deux méthodes principales : la méthode de collocation de Bernstein et la méthode de Bernstein–Petrov–Galerkin (BPG). Le chapitre comprend une présentation théorique de la méthodologie de résolution ainsi qu'une analyse approfondie des étapes de mise en œuvre.

Le dernier chapitre est dédié à l'application des modèles théoriques précédents dans l'environnement MATLAB à travers des exemples numériques pratiques pour la résolution d'équations différentielles. Les résultats sont analysés en termes de précision de la solution et de rapidité de convergence, avec une comparaison aux autres méthodes numériques, mettant ainsi en évidence l'efficacité des approches adoptées et la comparaison entre la méthode de collocation et la méthode de Bernstein–Petrov–Galerkin (BPG).

Polynômes de Bernstein

1.1 Introduction

Ce chapitre présente les définitions fondamentales et les propriétés mathématiques des polynômes de Bernstein. Il couvre leur construction de base, ainsi que leurs principales caractéristiques telles que la positivité, la symétrie, etc. Le rôle de ces polynômes dans l'approximation des fonctions est également discuté à travers le théorème de Weierstrass. En outre, les formes généralisées, les dérivées et les propriétés intégrales des polynômes de Bernstein sont introduites, préparant ainsi le terrain pour leur application dans les méthodes numériques de résolution des équations différentielles.

1.2 Définition des polynômes de Bernstein

Bernstein a introduit le concept de polynômes de Bernstein en 1911 dans son article " Une preuve du Théorème de Weierstrass basée sur le calcul des probabilités "[3].

Polynômes de base de Bernstein : Pour $0 \leq i \leq n$, les $n+1$ polynômes de base de Bernstein de degré n sur $x \in [0, 1]$ sont définis comme suit :

$$b_i^n(x) = \binom{n}{i} (1-x)^{n-i} x^i \tag{1.1}$$

Où $\binom{n}{i} = \frac{n!}{i!(n-i)!}$ et $n, i \in \mathbb{N}$.

Pour $i < 0$ et $i > n$ il est défini que $b_i^n(x) = 0$.

Généralisation du domaine de définition des polynômes de base de Bernstein : Dans leur forme classique, les polynômes de Bernstein sont définis sur l'intervalle standard $[0, 1]$. Toutefois, pour traiter des problèmes posés sur un intervalle quelconque $[a, b]$, il est nécessaire

d'effectuer un changement de variable approprié. Ce changement consiste à remplacer la variable $x \in [a, b]$ par la variable $t \in [0, 1]$, à l'aide de la transformation affine suivante :

$$t = \frac{x - a}{b - a}, \quad \text{ou inversement} \quad x = a + (b - a)t.$$

Cette substitution permet de généraliser les polynômes de Bernstein à l'intervalle $[a, b]$, en redéfinissant la base comme suit :

$$b_i^n(x) = \binom{n}{i} \left(\frac{x - a}{b - a}\right)^i \left(1 - \frac{x - a}{b - a}\right)^{n-i}, \quad \text{pour } x \in [a, b].$$

Exemple 1.2.1. Polynômes de base de Bernstein. *Considérons les polynômes de base de Bernstein sur $x \in [0, 1]$.*

► $n = 0$. *Le polynôme de base de Bernstein de degré 0 est donné par :*

$$b_0^0(x) = 1$$

► $n = 1$. *Les deux polynômes de base de Bernstein de degré 1 sont donnés par :*

$$b_0^1(x) = 1 - x, \quad b_1^1(x) = x$$

► $n = 2$. *Les trois polynômes de base de Bernstein de degré 2 sont donnés par :*

$$b_0^2(x) = (1 - x)^2, \quad b_1^2(x) = 2(1 - x)x, \quad b_2^2(x) = x^2$$

► $n = 3$. *Les quatre polynômes de base de Bernstein de degré 3 sont donnés par :*

$$b_0^3(x) = (1 - x)^3, \quad b_1^3(x) = 3(1 - x)^2x, \quad b_2^3(x) = 3(1 - x)x^2, \quad b_3^3(x) = x^3$$

► $n = 4$. *Les cinq polynômes de base de Bernstein de degré 4 sont donnés par :*

$$b_0^4(x) = (1 - x)^4, \quad b_1^4(x) = 4(1 - x)^3x, \quad b_2^4(x) = 6(1 - x)^2x^2, \quad b_3^4(x) = 4(1 - x)x^3, \quad b_4^4(x) = x^4$$

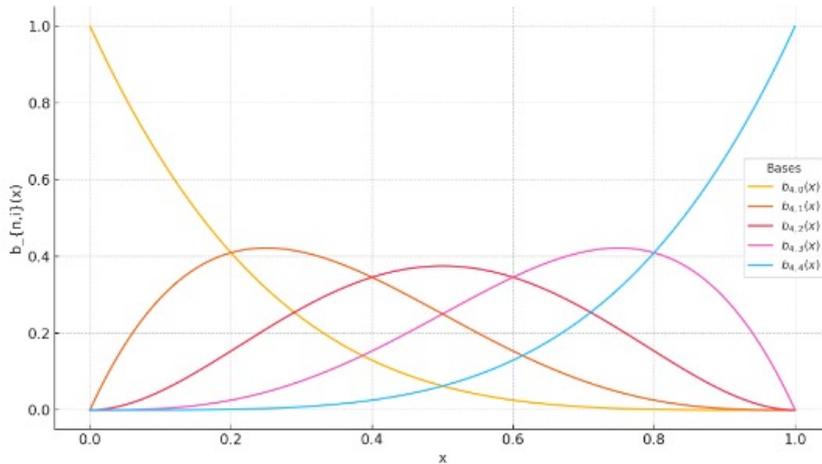
La Figure 1 présente les cinq polynômes de Bernstein de degré 4 dans la région $[0, 5]$

1.3 Propriétés des polynômes de Bernstein

Positivité :

Les polynômes de base de Bernstein de degré n sont positifs sur l'intervalle $[0, 1]$

$$\forall x \in [0, 1], \quad b_i^n(x) \geq 0$$


 FIGURE 1.1 – Polynômes de base de Bernstein pour $n = 4$

Recursive :

Les polynômes de Bernstein de degré n se construisent par combinaison linéaire de ceux de degré $n - 1$. Ainsi, chacun d'eux peut s'exprimer selon la relation suivante :

$$b_i^n(x) = \frac{n+1-i}{n+1} b_i^{n+1}(x) + \frac{i+1}{n+1} b_{i+1}^{n+1}(x) \quad (1.2)$$

$$b_i^n(x) = (1-x) b_i^{n-1}(x) + x b_{i-1}^{n-1}(x)$$

Partition de l'unité :

La somme des polynômes de Bernstein de degré n est toujours égale à 1 pour tout x peut s'écrire comme :

$$\sum_{i=0}^n b_i^n(x) = 1$$

Symétrie :

Les polynômes de base de Bernstein $b_i^n(x)$ et $b_{n-i}^n(x)$ sont symétriques par rapport à $x = \frac{1}{2}$, c'est-à-dire :

$$b_i^n(1-x) = b_{n-i}^n(x)$$

1.4 Polynômes de Bernstein généralisés

Dans cette section, nous introduisons les polynômes de Bernstein généralisés et leur application à l'approximation des fonctions continues et de leurs dérivées sur un intervalle fermé [8].

Définition 1.4.1. Soit $y : [a, b] \rightarrow \mathbb{R}$ une fonction continue sur l'intervalle $[a, b]$. Les poly-

nômes de Bernstein généralisés de degré n sont définis par :

$$B_n(y; x) = \sum_{i=0}^n y\left(a + \frac{b-a}{n}i\right) b_i^n(x)$$

Théorème 1.4.1. Si $y \in C^p[a, b]$ pour un entier $m \geq 0$, alors :

$$\lim_{n \rightarrow +\infty} B_n^{(p)}(y; x) = y^{(p)}(x) ; \quad p = 1, 2, \dots, m$$

converge uniformément .

1.5 Dérivées des polynômes de Bernstein

Cette section énonce des résultats clés sur les dérivées des polynômes de Bernstein, tels que présentés dans [1], et nécessaires à la formulation de la méthode de collocation de Bernstein développée dans le chapitre suivant.

Théorème 1.5.1.

i) Les dérivées des polynômes de base généralisés de Bernstein de degré n sont données par :

$$\frac{d}{dx} b_i^n(x) = \frac{n}{b-a} \left[b_{i-1}^{n-1}(x) - b_i^{n-1}(x) \right] ; \quad i = 1, 2, \dots, n. \quad (1.3)$$

ii) Les premières dérivées des polynômes de base généralisés de Bernstein de degré n peuvent être écrites comme une combinaison linéaire des polynômes de base généralisés de Bernstein de degré n :

$$\frac{d}{dx} b_i^n(x) = \frac{1}{b-a} \left[(n-i+1)b_{i-1}^n(x) + (2i-n)b_i^n(x) - (i+1)b_{i+1}^n(x) \right]$$

Démonstration.

i) Évidente .

ii) En utilisant la relation (1.2) les égalités suivantes peuvent être écrites sous la forme :

$$\begin{aligned} b_i^{n-1}(x) &= \frac{n-i}{n} b_i^n(x) + \frac{i+1}{n} b_{i+1}^n(x) \\ b_{i-1}^{n-1}(x) &= \frac{n-i+1}{n} b_{i-1}^n(x) + \frac{i}{n} b_i^n(x) \end{aligned}$$

La première dérivée s'écrit :

$$\frac{d}{dx} b_i^n(x) = \frac{n}{b-a} \left[b_{i-1}^{n-1}(x) - b_i^{n-1}(x) \right]$$

En remplaçant les relations ci-dessus :

$$\frac{d}{dx} b_i^n(x) = \frac{n}{b-a} \left[\left(\frac{n-i+1}{n} b_{i-1}^n(x) + \frac{i}{n} b_i^n(x) \right) - \left(\frac{n-i}{n} b_i^n(x) + \frac{i+1}{n} b_{i+1}^n(x) \right) \right]$$

donc :

$$\frac{d}{dx}b_i^n(x) = \frac{1}{b-a} \left[(n-i+1)b_{i-1}^n(x) + (2i-n)b_i^n(x) - (i+1)b_{i+1}^n(x) \right]$$

□

Théorème 1.5.2. *Il existe une relation entre la matrice des polynômes de base généralisés de Bernstein et leurs dérivées sous la forme suivante :*

$$b^{(p)}(x) = b(x)T^p \quad ; \quad p = 1, 2, \dots, m.$$

Ici, les éléments de la matrice $T = (t_{ij})_{0 \leq i, j \leq n}$ sont définis comme suit

$$t_{ij} = \frac{1}{b-a} \begin{cases} n-i, & \text{si } j = i+1, \\ 2i-n, & \text{si } j = i, \\ -i, & \text{si } j = i-1, \\ 0, & \text{Sinon.} \end{cases}$$

Démonstration. D'après le Théorème 1.6.1(ii) et la condition $b_j^i(x) = 0$ si $j > 0$ ou $j < 0$, nous avons :

$$\begin{aligned} (b_0^n)'(x) &= \frac{1}{b-a} [-nb_0^n(x) - b_1^n(x)] \\ (b_1^n)'(x) &= \frac{1}{b-a} [nb_0^n(x) + (2-n)b_1^n(x) - 2b_2^n(x)] \\ (b_2^n)'(x) &= \frac{1}{b-a} [(n-1)b_1^n(x) + (4-n)b_2^n(x) - 3b_3^n(x)] \\ &\vdots \\ (b_{i-1}^n)'(x) &= \frac{1}{b-a} [2b_{i-2}^n(x) + (n-2)b_{i-1}^n(x) - nb_i^n(x)] \\ (b_i^n)'(x) &= \frac{1}{b-a} [b_{i-1}^n(x) + nb_i^n(x)] \end{aligned}$$

Ainsi, nous obtenons la relation matricielle

$$b'(x) = b(x)T$$

tel que

$$\begin{aligned} b(x) &= [b_0^n(x) \quad b_1^n(x) \quad b_2^n(x) \quad \cdots \quad b_{n-1}^n(x) \quad b_n^n(x)], \\ b'(x) &= [(b_0^n)'(x) \quad (b_1^n)'(x) \quad (b_2^n)'(x) \quad \cdots \quad (b_{n-1}^n)'(x) \quad (b_n^n)'(x)], \end{aligned}$$

et

$$T = \frac{1}{b-a} \begin{bmatrix} -n & n & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2-n & n-1 & \cdots & 0 & 0 & 0 \\ 0 & -2 & 4-n & \cdots & 0 & 0 & 0 \\ 0 & 0 & -3 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & n-4 & 2 & 0 \\ 0 & 0 & 0 & \cdots & 1-n & n-2 & 1 \\ 0 & 0 & 0 & \cdots & 0 & -n & n \end{bmatrix}.$$

De la même manière, la dérivée seconde devient

$$b''(x) = b'(x)T = b(x)T^2$$

On obtient ainsi des dérivées des polynômes de base de Bernstein généralisés sous la forme

$$b^{(p)}(x) = b^{(p-1)}(x)T = b(x)T^p$$

□

1.6 Intégrales des polynômes de Bernstein

Dans cette section, nous présentons quelques résultats sur les intégrales des polynômes de Bernstein, tirés de [4]. Ces résultats joueront un rôle important dans la formulation de la méthode de Bernstein–Petrov–Galerkin développée ultérieurement.

1.6.1 Propriétés intégrales des polynômes de Bernstein

Les intégrales des polynômes de Bernstein possèdent des propriétés simples et utiles. Chaque polynôme $b_i^n(x)$ a une intégrale égale à $\frac{b-a}{n+1}$ sur l'intervalle limité $[a, b]$

$$\int_a^b b_i^n(x) dx = \frac{b-a}{n+1}$$

L'intégrale indéfinie de la fonction de base de Bernstein est donnée par :

$$\int b_i^n(x) dx = \frac{b-a}{n+1} \sum_{j=i+1}^{n+1} b_n^{n+1}(x).$$

Le produit de deux bases de Bernstein conduit à :

$$b_i^n(x)b_k^m(x) = \frac{\binom{n}{i}\binom{m}{k}}{\binom{n+m}{i+k}} b_{i+k}^{n+m}(x). \quad (1.4)$$

Nous rappelons ci-dessous un résultat important concernant les intégrales successives des polynômes de Bernstein

Théorème 1.6.1. [4]

$$I^k b_i^n(x) = (b-a)^k \frac{n!}{(n+k)!} \sum_{j=i+k}^{n+k} \binom{j-i-1}{k-1} b_j^{n+k}(x) + \bar{\pi}_{k-1}(x)$$

où $I^k f(x)$ est la k -ième intégrale de $f(x)$ par rapport à x , et $\bar{\pi}_0(x)$ est un polynôme d'ordre $k-1$.

Nous ne donnons pas ici la démonstration de ce Théorème, laquelle a été présentée par E.H. Doha et al. [4]. Dans leur approche, les auteurs s'appuient sur un résultat de Farouki et Rajan [5], selon lequel

$$\int b_i^n(x) dx = \frac{b-a}{n+1} \sum_{n=i+1}^{n+1} b_j^{n+1}(x) + \bar{\pi}_0(x).$$

Ils utilisent ensuite le Théorème de Chu [9] pour aboutir au résultat final.

Comme conséquence du théorème, on obtient le corollaire suivant.

Corollaire 1.6.1.

$$\int_a^b I^k b_i^n(x) b_q^n(x) = \frac{(b-a)^{k+1} n!}{(2n+k+1)(n+k)!} \sum_{j=i+k}^{n+k} \frac{\binom{j-i-1}{k-1} \binom{n}{k} \binom{n+k}{j}}{\binom{2n+k}{q+j}}.$$

1.7 Théorème d'approximation de Weierstrass

La théorie de l'approximation joue un rôle fondamental en analyse numérique et dans l'approximation des fonctions continues. Elle stipule que toute fonction continue sur un domaine fermé peut être approximée par une suite de fonctions polynomiales. Cette théorie a des implications profondes tant en mathématiques théoriques que dans les applications pratiques, où les polynômes de Bernstein peuvent être exploités de manière computationnelle.

Définition 1.7.1. Soit f une fonction continue sur $[0, 1]$. Les polynômes de Bernstein de degré n associés à f sont définis par :

$$B_n(f; x) = \sum_{i=0}^n f\left(\frac{i}{n}\right) b_i^n(x) \tag{1.5}$$

Théorème 1.7.1. [10] Si f est une fonction continue sur l'intervalle compact $[0, 1]$, alors la suite $B_n(f; x)$ converge uniformément vers f sur $[0, 1]$:

$$\lim_{n \rightarrow +\infty} \sup_{x \in [0,1]} |B_n(f; x) - f(x)| = 0.$$

De plus, pour tout $\varepsilon > 0$, il existe un entier n suffisamment grand tel que, pour tout $x \in [0, 1]$, on ait :

$$|f(x) - B_n(f; x)| < \varepsilon.$$

Méthodes numériques basées sur les polynômes de Bernstein pour les équations différentielles

2.1 Introduction

Ce chapitre explore l'application des polynômes de Bernstein dans le cadre des méthodes numériques pour la résolution des équations différentielles. Nous commençons par rappeler les notions de base sur les équations différentielles, puis nous introduisons la méthode de collocation de Bernstein ainsi que la méthode de Bernstein–Petrov–Galerkin (BPG). Les formulations mathématiques, les étapes de mise en œuvre et l'analyse de l'erreur y sont détaillées.

2.2 Équations Différentielles

Les équations différentielles modélisent l'évolution d'une ou plusieurs fonctions inconnues en fonction d'une ou plusieurs variables indépendantes. Elles occupent une place essentielle en mathématiques appliquées, en physique, en ingénierie, ainsi que dans de nombreux autres domaines scientifiques.

En termes simples, une équation différentielle est une relation mathématique impliquant une fonction et ses dérivées, c'est-à-dire le taux de variation de cette fonction .

Définitions

- Une équation différentielle est une équation qui relie une fonction inconnue à une ou plusieurs de ses dérivées. Elle permet de modéliser l'évolution d'un phénomène en fonction d'une ou plusieurs variables indépendantes.
- L'ordre d'une équation différentielle est le rang de la dérivée la plus élevée de la fonction inconnue qui apparaît dans l'équation. En d'autres termes, c'est le plus grand nombre de différentiations successives appliquées à la fonction.

2.2.1 Types d'équations différentielles

Il existe deux types d'équations différentielles :

1. Équations aux dérivées partielles (EDP)

Une équation aux dérivées partielles, appelée EDP, est une équation différentielle contenant les dérivées partielles d'une ou plusieurs variables dépendantes avec plusieurs variables indépendantes.

La forme générale d'une EDP d'ordre n est donnée par :

$$F\left(x, y, \frac{\partial y}{\partial x_1}, \dots, \frac{\partial y}{\partial x_n}; \frac{\partial^2 y}{\partial x_1 \partial x_1}, \dots, \frac{\partial^2 y}{\partial x_1 \partial x_n}; \dots\right) = 0.$$

Exemple : $\frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial x^2} = -4x^2 + y$

2. Équations différentielles ordinaires (EDO)

Une équation différentielle est dite ordinaire (EDO), si la fonction inconnue dépend d'une seule variable indépendante, la variable dépendante réelle y , et certaines de ses dérivées.

La forme générale d'une EDO d'ordre n est donnée par :

$$F(x, y, y', \dots, y^{(n)}) = 0.$$

Exemple : $y'' + 2y = 0$

2.2.2 Équation différentielle ordinaire linéaire

Une équation différentielle ordinaire linéaire d'ordre n est une équation qui s'écrit sous la forme

$$a_n(x)y^{(n)}(x) + a_{n-1}(x)y^{(n-1)}(x) + \dots + a_1(x)y'(x) + a_0(x)y(x) = f(x),$$

où :

- $y(x)$ est la fonction inconnue,
- les coefficients $a_0(x), \dots, a_n(x)$ et le second membre $f(x)$ sont des fonctions données continues sur un intervalle I ,
- $a_n(x) \neq 0$ pour tout $x \in I$.

L'équation est dite *homogène* si $f(x) = 0$ sur I , sinon elle est *non homogène*.

Dans ce mémoire, nous nous intéressons exclusivement aux *équations différentielles ordinaires linéaires*, compte tenu de leur importance théorique et de leurs nombreuses applications dans les méthodes numériques étudiées.

2.3 Méthode de collocation de Bernstein

Dans cette section, nous présentons une méthode développée par [1], fondée sur la méthode de collocation et utilisant une généralisation des polynômes de Bernstein pour la résolution d'équations différentielles ordinaires linéaires d'ordre élevé, sous conditions initiales ou conditions aux bords. Cette méthode exploite également les propriétés des dérivées des polynômes de Bernstein présentées dans le chapitre précédent, ce qui permet de construire efficacement les équations de collocation et d'assurer une bonne précision dans l'approximation de la solution.

2.3.1 Formulation du problème

Nous partons de l'équation différentielle d'ordre m :

$$\sum_{p=0}^m a_p(x)y^{(p)}(x) = f(x) \quad ; \quad a \leq x \leq b \quad (2.1)$$

- Sous les conditions

$$\sum_{p=0}^{m-1} \gamma_{np}y^{(p)}(c) = \mu_n \quad ; \quad c \in [a, b] \quad (2.2)$$

- Ou

$$\sum_{p=0}^{m-1} [\alpha_{np}y^{(p)}(a) + \beta_{np}y^{(p)}(b)] = \vartheta_n \quad (2.3)$$

pour $n = 0, 1, \dots, m-1$, avec le polynôme de Bernstein généralisé :

$$y^{(p)}(x) \cong B_N^{(p)} = \sum_{i=0}^N y(a + \frac{(b-a)i}{N})(b_i^n)^{(p)}(x). \quad (2.4)$$

Ici, $a_p(x)$ et $f(x)$ sont des fonctions continues sur l'intervalle $[a, b]$, γ_{np} , α_{np} , β_{np} , μ_n et ϑ_n sont des constantes connues, tandis que $y(x)$ est la fonction inconnue que nous cherchons à résoudre.

Théorème 2.3.1. *Soit $x_i \in [0, 1]$; $i = 0, 1, 2, \dots, N$ les points de collocation. L'équation différentielle linéaire générale (2.1) non homogène d'ordre m peut être écrite sous forme matricielle comme suit :*

$$\sum_{p=0}^m A_p b T^p Y = F \quad (2.5)$$

Ici, les matrices sont :

$$Y = [y(a + (b-a)i/N)], \quad A_p = \text{diag}[a_p(x_i)], \quad b = [b_i^n(x_i)] \quad \text{et} \quad F = [f(x_i)], \quad \text{pour } i, j = 0, 1, \dots, N$$

Démonstration. L'expression (2.4) peut être représentée sous la forme matricielle suivante :

$$y^{(p)}(x) \cong B_N^{(p)}(y; x) = P^{(p)}(x)Y$$

En utilisant le Théorème 1.5.2, les dérivées des fonctions inconnues peuvent également être

écrites sous la forme suivante :

$$y^{(p)}(x) \cong P(x)T^pY \quad ; \quad p = 1, 2, \dots, m. \quad (2.6)$$

En substituant les points de collocation et la relation (2.6) dans l'équation (2.1), nous obtenons le système d'équations algébriques linéaires Suivant :

$$\sum_{p=0}^m a_p(x_i)P(x_i)T^pY = f(x_i) \quad ; \quad i = 1, 2, \dots, n$$

De sorte que $y^{(p)}(x_i) \cong B_N^{(p)}(y; x_i)$, Ce système d'équations peut être représenté sous la forme matricielle (2.5), ce qui complète la démonstration. \square

2.3.2 Les étapes de la méthode collocation de Bernstein

Nous pouvons résoudre l'équation différentielle à coefficients variables (2.1) sous les conditions (2.2) ou (2.3) en suivant les étapes suivantes :

Étape 1 : L'équation (2.5) peut être écrite sous une forme compacte

$$WY = F \quad \text{ou} \quad [W; F] \quad (2.7)$$

avec $W = \sum_{p=0}^m A_p b T^p$. Cette équation matricielle (2.7) correspond à un système algébrique linéaire avec des coefficients inconnus y_0, y_1, \dots, y_N .

Étape 2 : À partir de l'expression (2.6), les formes matricielles des conditions (2.2) et (2.3) peuvent être écrites respectivement

$$U_j = \sum_{p=0}^{m-1} \gamma_j y^{(p)}(c) = [U_j^0 \quad U_j^1 \quad \dots \quad U_j^N]$$

$$V_j = \sum_{p=0}^{m-1} [\alpha_{jp} y^{(p)}(a) + \beta_{jp} y^{(p)}(b)] = [V_j^0 \quad V_j^1 \quad \dots \quad V_j^N]$$

ou implicitement

$$U_j Y = \mu_j \quad \text{ou} \quad [U_j; \mu_j] \quad (2.8)$$

$$V_j Y = \nu_j \quad \text{ou} \quad [V_j; \nu_j] \quad (2.9)$$

Étape 3 : Pour obtenir la solution de l'équation (2.1) sous les conditions (2.2) ou (2.3), nous ajoutons les éléments des matrices lignes (2.8) ou (2.9) à la fin de la matrice (2.7). Ainsi, nous obtenons la nouvelle matrice augmentée $[\tilde{W}; \tilde{F}]$. Cette matrice augmentée est une matrice rectangulaire de dimension $(N + m + 1) \times (N + 1)$. Cette nouvelle équation matricielle peut être notée de manière concise par $\tilde{W}Y = \tilde{F}$.

Étape 4 : Si $\text{rank}(\tilde{W}) = \text{rank}[\tilde{W}; \tilde{F}] = N + 1$ alors les coefficients inconnus $y_i; i = 0, 1, \dots, N$ sont uniquement déterminés. Ces types de systèmes peuvent être résolus par les méthodes d'élimination de Gauss, d'inverse généralisée et de factorisation QR.

2.4 Méthode de Bernstein–Petrov–Galerkin

E.H. Doha et al. dans [4] ont proposé une méthode numérique pour la résolution d'équations différentielles linéaires d'ordre k , basée sur l'intégration k fois de l'équation considérée. Les formes intégrées ainsi obtenues sont ensuite développées sur la base des polynômes de Bernstein, ce qui permet d'exprimer les coefficients d'expansion à l'aide de formules spécifiques. Ces formules, faisant intervenir les intégrales des polynômes de Bernstein présentées dans le chapitre précédent, seront utilisées dans la description de la méthode. La méthode d'approximation Bernstein–Petrov–Galerkin (BPG) est alors appliquée pour déterminer une solution approchée de l'équation initiale.

Les polynômes de Bernstein se distinguent par une propriété particulière : tous les polynômes de la base s'annulent sur l'intervalle $[0, 1]$, à l'exception du premier au point $x = 0$ et du dernier au point $x = 1$, qui valent tous deux 1 à leurs extrémités respectives. Cette caractéristique offre une grande flexibilité lors de l'imposition des conditions aux limites aux bornes de l'intervalle. En conséquence, ces bases sont adoptées comme un outil numérique efficace pour la résolution des équations différentielles ordinaires d'ordre $2m$ [4].

2.4.1 Formulation du problème

Considérons l'équations différentielle suivante

$$y^{(2m)}(x) + \alpha y(x) = g(x) \quad , \quad \text{dans } I = [a, b], \quad (2.10)$$

soumise aux conditions aux limites suivantes :

$$y^{(k)}(a) = 0 \quad , \quad y^{(k)}(b) = 0 \quad , \quad 0 \leq k \leq m - 1$$

où α est une constante réelle et $g(x)$ est une fonction source donnée.

L'hypothèse de conditions aux limites homogènes ne restreint pas la généralité, car les problèmes aux limites non homogènes peuvent être facilement réduits à des problèmes équivalents avec des conditions homogènes en appliquant des transformations appropriées.

2.4.2 Les étapes de la méthode de BPG

Étape 1 : Nous étudions la forme intégrée de l'équation (2.10) avec ses conditions aux limites comme suit :

$$y(x) + \alpha \int^{(2m)} y(x)(dx)^{(2m)} = f(x) - \sum_{i=0}^{2m-1} d_i b_i^i(x) \quad (2.11)$$

où d_i sont des constantes d'intégration et

$$\int^{(k)} y(x)(dx)^{(k)} = \underbrace{\int \cdots \int}_{k \text{ fois}} y(x) \underbrace{dx dx \cdots dx}_{k \text{ fois}}, \quad f(x) = \int^{(2m)} g(x)(dx)^{(2m)}$$

Étape 2 : On note par $L^2(I)(I := [a, b])$ l'espace L^2 muni du produit scalaire $(y, v) = \int_I y(x)v(x)dx$, et de la norme associée $\|y\| = (y, v)^{1/2}$. Il est clair que l'ensemble $\{b_i^i(x) : n \geq 0, 0 \leq i \leq n\}$ forme un système complet dans $L^2(I)$. Par conséquent, on défini

$$S_N = \text{span} \{b_0^N(x), b_1^N(x), \dots, b_N^N(x)\},$$

$$V_N = \{v \in S_N : v^{(k)}(a) = v^{(k)}(b) = 0; 0 \leq k \leq m-1\}, \quad W_N = \{v \in S_N\}.$$

alors l'approximation BPG de (2.10) consiste à trouver $y_N \in V_N$ tel que

$$\alpha \left(\int^{(2m)} y(x)(dx)^{(2m)}, v \right) + (y_N(x), v) + \left(\sum_{i=0}^{2m-1} d_i b_i^i(x), v \right) = (f(x), v), \quad \forall v \in W_N \quad (2.12)$$

Étape 3 : Nous choisissons la solution approchée $y_N(x)$ et le dual de la base v , sous la forme

$$y_N(x) = \sum_{i=m}^{N-m} a_i b_i^N(x) \quad \text{et} \quad v = b_h^N(x), \quad 0 \leq h \leq N \quad (2.13)$$

Il est maintenant clair que (2.12) est équivalent à :

$$\begin{aligned} \sum_{i=m}^{N-m} a_i \left[\alpha \left(\int^{(2m)} b_i^N(x) (dx)^{(2m)}, b_h^N(x) \right) + (b_i^N(x), b_h^N(x)) \right] + \sum_{i=0}^{2m-1} d_i (b_i^i(x), b_h^N(x)) \\ = (f(x), b_h^N(x)), \quad \forall v \in W_N, \quad h = 0, 1, \dots, N. \end{aligned}$$

Étape 4 : Désignons par :

$$\begin{cases} f_h = (f, b_h^N(x)), & f = (f_0, f_1, \dots, f_N)^T \\ a = (d_0, \dots, d_{m-1}, a_m, \dots, a_{N-m}, d_m, \dots, d_{2m-1})^T \\ A = (a_{hj})_{0 \leq h, j \leq N}, & B = (b_{hj})_{0 \leq h, j \leq N}, & D = (d_{hj})_{0 \leq h, j \leq N}; \end{cases}$$

alors est équivalente à l'équation matricielle :

$$(A + \alpha B + D)a = f$$

où les éléments des matrices A, B et D de taille $(N+1) \times (N+1)$ pour $0 \leq h, j \leq N$, sont donnés explicitement dans le théorème suivant :

Théorème 2.4.1. *Si l'on prend v tel que défini dans (3.1), et si l'on désigne*

$$a_{hj} = \begin{cases} (b_j^N(x), b_h^N(x)) & \text{si } 0 \leq h \leq N, m \leq j \leq N - m, \\ 0 & \text{sinon} \end{cases}$$

$$b_{hj} = \begin{cases} \left(\int^{(2m)} b_j^N(x) (dx)^{(2m)}, b_h^N(x) \right) & \text{si } 0 \leq h \leq N, m \leq n \leq N - m, \\ 0 & \text{sinon} \end{cases}$$

$$d_{hj} = \begin{cases} (b_j^j(x), b_h^N(x)) & \text{si } 0 \leq h \leq N, 0 \leq j < m, \\ 0 & \text{si } 0 \leq h \leq N, m \leq j \leq N - m, \\ (b_{j-N+2m-1}^{j-N+2m-1}(x), b_h^N(x)) & \text{si } 0 \leq h \leq N, N - m < j \leq N, \end{cases}$$

Les éléments non nuls des matrices A, B et D dans le système sont donnés explicitement par :

$$a_{hn} = \frac{(b-a)}{(2N+1)} \frac{\binom{N}{j} \binom{N}{h}}{\binom{2N}{h+j}}, \quad b_{hj} = \frac{(b-a)^{2m+1} N! \binom{N}{h}}{(N+2m)! (2N+2m+1)} \sum_{i=j+2m}^{N+2m} \frac{\binom{N+2m}{i} \binom{i-j-1}{2m-1}}{\binom{2N+2m}{h+i}}$$

$$d_{hj} = \begin{cases} \frac{\binom{N}{H}}{\binom{N+j}{h+j}} \frac{b-a}{N+j+1} & \text{si } 0 \leq j \leq m, \\ \frac{\binom{N}{h}}{\binom{j+2m-1}{h+j-N+2m-1}} \frac{b-a}{j+2m} & \text{si } N - m \leq j \leq N, \end{cases}$$

Applications Numériques

3.1 Introduction

Ce chapitre est consacré à la présentation d'applications numériques illustrant l'efficacité des méthodes basées sur les polynômes de Bernstein pour résoudre des équations différentielles. Plusieurs exemples sont développés pour mettre en œuvre la méthode de collocation et la méthode de Bernstein–Petrov–Galerkin (BPG). Des comparaisons sont effectuées avec d'autres méthodes existantes afin d'évaluer la précision et la rapidité de convergence des approches proposées. L'ensemble des calculs numériques a été réalisé à l'aide du logiciel MATLAB, à travers des programmes spécialement conçus pour mettre en œuvre les différentes approches étudiées.

Au début du chapitre, nous introduisons également la définition de l'erreur d'approximation, qui permettra d'évaluer quantitativement les performances des méthodes appliquées.

3.2 Erreur d'approximation

Définition 3.2.1. *L'erreur d'approximation est notée $e_n(x) = y(x) - B_n(f; x)$ où $y(x)$ est la solution exacte et $B_n(y; x)$ est la solution approchée de Bernstein. L'erreur maximale peut alors être définie comme :*

$$E_n(y; [a, b]) = E_n(y) = \| e_n(x) \|_{\infty} = \max_{a \leq x \leq b} |e_n(x)|$$

Et sur les points de collocation, l'erreur maximale et l'erreur moyenne peuvent également être calculées numériquement comme suit :

$$E_{\max} = \max_{x_i} |e_n(x_i)| \quad , \quad E_{\text{mean}} = \frac{1}{n} \sum_{i=0}^{n-1} |e_n(x_i)|$$

Le calcul de ces erreurs sur les points de collocation est essentiel dans la pratique, car il permet d'évaluer la qualité de l'approximation obtenue. L'erreur maximale E_{\max} fournit une

mesure du pire cas, ce qui est crucial pour garantir la fiabilité de la méthode dans l'ensemble du domaine. Quant à l'erreur moyenne E_{mean} , elle donne une estimation globale de la précision moyenne de l'approximation, utile pour comparer différentes méthodes ou choix de paramètres numériques.

3.3 Applications de la Méthode de collocation de Bernstein

3.3.1 Exemple 1

Considérons le problème aux limites suivant :

$$\begin{cases} y'' + xy' - 2y = x \cos(x) - 3 \sin(x) , & 0 \leq x \leq 1, \\ y(0) = 0 , & y'(0) = 1. \end{cases}$$

Où la solution exacte est donnée par : $y = \sin(x)$.

Dans ce qui suit, nous allons approximer la solution $y(x)$ à l'aide de la méthode de collocation de Bernstein pour $N = 3$.

L'équation différentielle est de la forme :

$$\sum_{p=0}^m a_p(x)y^{(p)}(x) = f(x) \quad ; \quad 0 \leq x \leq 1$$

Ce qui donne :

$$m = 2, \quad f(x) = x \cos(x) - 3 \sin(x), \quad a_0(x) = -2, \quad a_1(x) = x, \quad a_2(x) = 1,$$

Nous allons écrire la forme :

$$\sum_{p=0}^2 A_p b N^p Y = F$$

C'est-à-dire :

$$A_0 b N^0 Y + A_1 b N^1 Y + A_2 b N^2 Y = F$$

$$A_0 b Y + A_1 b N^1 Y + A_2 b N^2 Y = F \implies [A_0 b + A_1 b N^1 + A_2 b N^2] Y = F \implies W Y = F$$

Nous avons :

$$A_0 = \begin{bmatrix} -2 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & -2 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 \\ 0 & 0 & 2/3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 8/27 & 4/9 & 2/9 & 1/27 \\ 1/27 & 2/9 & 4/9 & 8/27 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad N = \begin{bmatrix} -3 & 3 & 0 & 0 \\ -1 & -1 & 2 & 0 \\ 0 & -2 & 1 & 1 \\ 0 & 0 & -3 & 3 \end{bmatrix}$$

et

$$F = [0, -6.6660 \times 10^{-1}, -1.3312, -1.9841]^T$$

Ainsi, nous calculons la matrice W :

$$W = A_0b + A_1bN^1 + A_2bN^2 \implies W = \begin{bmatrix} 4 & -12 & 6 & 0 \\ 80/27 & -62/9 & -1/9 & 55/27 \\ 46/27 & -10/9 & -62/9 & 116/27 \\ 0 & 6 & -15 & 7 \end{bmatrix}$$

Prise en compte des conditions initiales

Nous devons maintenant intégrer les conditions aux matrices. D'après les équations (2.1) et (2.2), nous avons :

$$b(0)N^0 = 0 \implies [1 \ 0 \ 0 \ 0] = 0$$

$$b(0)N^1 = 1 \implies [-3 \ 3 \ 0 \ 0] = 1$$

Nous remplaçons ces dernières lignes dans la matrice W :

$$\tilde{W} = \begin{bmatrix} 4 & -12 & 6 & 0 \\ 80/27 & -62/9 & -1/9 & 55/27 \\ 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \end{bmatrix}, \quad \tilde{F} = \begin{bmatrix} 0 \\ -6.6660 \times 10^{-01} \\ 0 \\ 1 \end{bmatrix}$$

Nous résolvons ensuite le système linéaire :

$$\tilde{W}Y = \tilde{F} \implies Y = \tilde{W}^{-1}\tilde{F}$$

Ce qui donne la solution

$$Y = [-9.2520 \times 10^{-17}, 3.3333 \times 10^{-1}, 6.6667 \times 10^{-1}, 8.3640 \times 10^{-1}]^T$$

Par la relation (2.4) précédente, la solution approximative est donnée comme suite :

$$y(x) \simeq y_3(x) = \sum_{i=0}^4 Y(i+1)b_i^n$$

Ce qui donne le pôleynome suivant :

$$y_3(x) = -0.16362x^3 + 2.9999 \times 10^{-5}x^2 + 0.99999x - 9.252 \times 10^{-17}$$

Nous allons appliquer maintenant la méthode au problème étudié pour différentes valeurs de N , à savoir $N = 3, 9, 12$. Les résultats obtenus pour l'erreur absolue $e_n(x_i)$ aux points $x_i = \frac{i}{5}$ pour $i = 0, 1, 2, 3, 4, 5$ sont présentés dans le tableau 3.1. Les résultats montrent clairement que l'augmentation du nombre de points de collocation N conduit à une réduction significative des erreurs.

Méthode de collocation de Bernstein			
x_i	$N = 3, e_3(x)$	$N = 9, e_9(x)$	$N = 12, e_{12}(x)$
0	9.2520×10^{-17}	3.1341×10^{-16}	5.9565×10^{-18}
0.2	2.1846×10^{-05}	1.1946×10^{-12}	2.7756×10^{-16}
0.4	1.1107×10^{-04}	2.6019×10^{-12}	7.7716×10^{-16}
0.6	1.9306×10^{-05}	4.1220×10^{-12}	1.8874×10^{-15}
0.8	1.1208×10^{-03}	6.1541×10^{-12}	1.5543×10^{-15}
1	5.0739×10^{-03}	1.9885×10^{-10}	1.9762×10^{-14}

Tableau 3.1 – Erreurs absolues pour différentes valeurs de N .

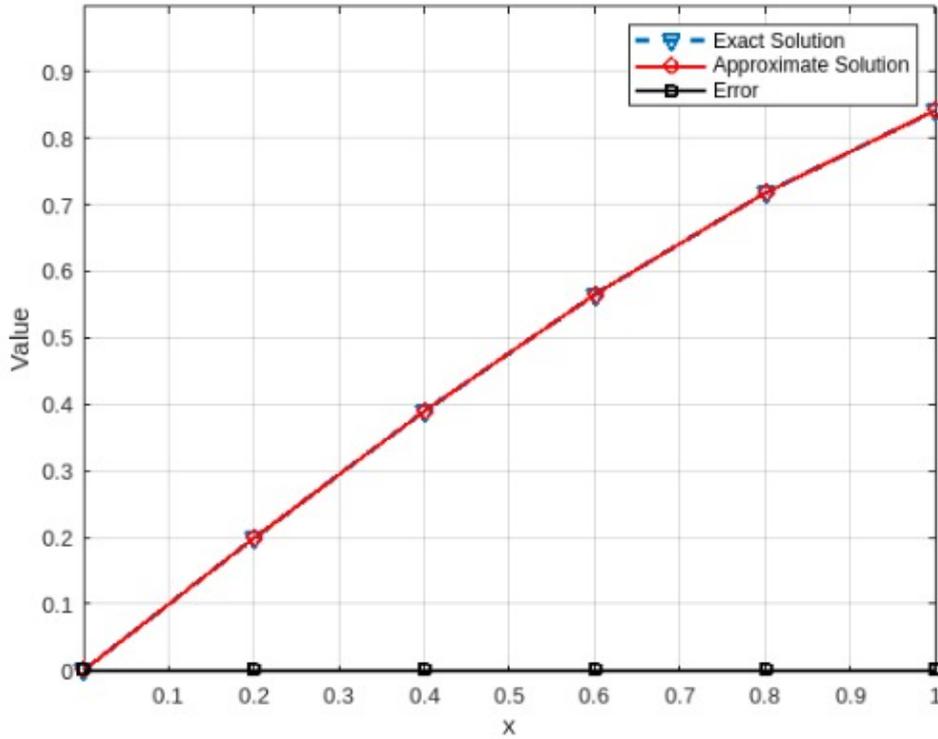


FIGURE 3.1 – Solution exacte, solution approchée pour $N = 5$ et erreur absolue de l'exemple 1.

Dans le Tableau 3.2, les erreurs maximales sont calculées en utilisant la méthode présentée et la méthode rationnelle [7]. Les résultats pour différentes valeurs de N montrent que la méthode de collocation de Bernstein donne des erreurs nettement plus faibles. De plus, plus la valeur de N est grande, plus la solution approchée est précise et l'erreur devient plus petite, démontrant ainsi l'efficacité et la rapidité de convergence de la méthode proposée par rapport à la méthode rationnelle.

N	Méthode de collocation de Bernstein	Rationnel [7]
5	3.404764×10^{-05}	7.2×10^{-03}
7	1.074951×10^{-07}	4.2×10^{-04}
10	1.616773×10^{-11}	3.6×10^{-08}

Tableau 3.2 – Comparaison de l'erreur maximale E_{\max} de l'exemple 1.

3.3.2 Exemple 2 : Étude comparative

Considérons le problème aux limites d'ordre 4 suivant [2] :

$$\begin{cases} y^{(4)} = y + y'' + e^x(x - 3) , & 0 \leq x \leq 1, \\ y(0) = 1 , \quad y'(0) = 0 , \quad y'(1) = -e , \quad y(1) = 0. \end{cases}$$

Où la solution exacte est donnée par : $y = (1 - x)e^x$

x_n	Solution exacte	Collocation de Bernstein (N=10)	DTM [6]	VIM [2]
0.0	1.0000000000000000	0.9999999999999999	1.0000000000000000	1.0000000000
0.1	0.994653826268083	0.994653826698218	0.9946538282	0.9946538262
0.2	0.977122206528136	0.977122208040954	0.9771222134	0.9771222064
0.3	0.944901165303202	0.944901168244292	0.944901178573	0.9449011656
0.4	0.895094818584762	0.895094822996323	0.8950948382513	0.8950948188
0.5	0.824360635350064	0.824360640971620	0.82436065997	0.8243606355
0.6	0.728847520156204	0.728847526422485	0.728847546856	0.7288475200
0.7	0.604125812241143	0.604125818276875	0.604125836844	0.6041258121
0.8	0.445108185698493	0.445108190334130	0.44510820332	0.4451081856
0.9	0.245960311115695	0.245960313171697	0.2459603181764	0.2459603111
1.0	0.0000000000	0.0000000000	0.0000000000	0.0000000000

Tableau 3.3 – Comparaison des solutions approchées obtenues par différentes méthodes numériques.

La méthode de collocation de Bernstein est appliquée pour $N = 10$ et $N = 12$ pour résoudre numériquement le problème, et les résultats sont comparés à ceux obtenus par les méthodes VIM (Variational iteration method) [2] et DTM (Differential transform method) [6], tels qu'ils sont présentés dans les travaux de ces références.

x_n	Collocation de Bernstein (N=10)	Collocation de Bernstein (N=12)	DTM [6]	VIM [2]
0.0	6.6613×10^{-16}	0	0	0
0.1	4.3013×10^{-10}	8.8274×10^{-13}	1.97×10^{-09}	7.08×10^{-06}
0.2	1.5128×10^{-09}	3.0959×10^{-12}	6.88×10^{-09}	1.34×10^{-05}
0.3	2.9411×10^{-09}	6.0275×10^{-12}	1.32×10^{-08}	1.85×10^{-05}
0.4	4.4116×10^{-09}	9.0726×10^{-12}	1.96×10^{-08}	2.18×10^{-05}
0.5	5.6216×10^{-09}	1.1625×10^{-11}	2.46×10^{-08}	2.29×10^{-05}
0.6	6.2663×10^{-09}	1.3076×10^{-11}	2.66×10^{-08}	2.17×10^{-05}
0.7	6.0357×10^{-09}	1.2805×10^{-11}	2.46×10^{-08}	1.84×10^{-05}
0.8	4.6356×10^{-09}	1.0183×10^{-11}	1.76×10^{-08}	1.84×10^{-05}
0.9	2.0560×10^{-09}	4.8546×10^{-12}	7.06×10^{-09}	1.84×10^{-06}
1.0	0	0	0	0

Tableau 3.4 – Comparaison des erreurs d'approximation : Collocation de Bernstein, DTM [6] et VIM [2].

Le Tableau 3.4 montre que la méthode de collocation de Bernstein fournit, pour $N = 10$, des erreurs déjà significativement plus faibles que celles des méthodes DTM et VIM. Lorsque l'ordre est augmenté à $N = 12$, les erreurs obtenues avec la méthode de collocation de Bernstein diminuent de façon notable sur l'ensemble des points considérés. Cette diminution systématique des erreurs avec l'augmentation de N confirme la bonne convergence de la méthode. Par comparaison, les méthodes DTM et VIM présentent des erreurs nettement plus élevées. En résumé, la méthode de collocation de Bernstein, en plus d'être compétitive pour $N = 10$, montre une amélioration significative de la précision lorsque l'ordre N augmente, ce qui en fait une méthode à la fois précise et efficace pour la résolution de ce type de problème.

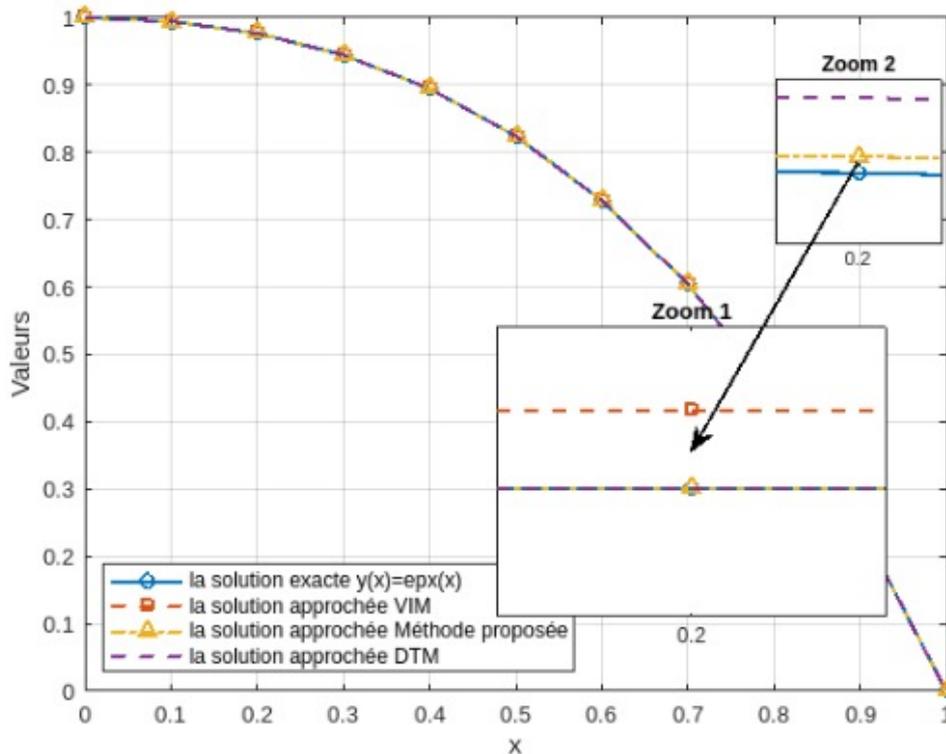


FIGURE 3.2 – Comparaison des méthodes numériques .

3.4 Application de la Méthode de Bernstein–Petrov–Galerkin

3.4.1 Problème aux limites homogènes

Considérons le problème aux limites suivant :

$$\begin{cases} y^{(2)}(x) - y(x) = (4 - 2x^2) \sin(x) + 4x \cos(x) , & 0 \leq x \leq 1, \\ y(0) = y(1) = 0 . \end{cases}$$

Ce problème est linéaire, d'ordre deux, avec des conditions aux limites homogènes. Il admet une solution exacte $y(x) = (x^2 - 1) \sin(x)$, ce qui permet d'évaluer la précision des méthodes numériques utilisées.

Nous allons appliquer la méthode de BPG pour $N = 3$ pour obtenir une solution numérique approximative du problème. Ici, $m = 1$, $\alpha = -1$ et $g(x) = (4 - 2x^2) \sin(x) + 4x \cos(x)$.

Nous allons calculer la solution du système de taille 4 suivant :

$$(A + \alpha B + D)a = f \quad ; \quad 0 \leq x \leq 1$$

où les matrices sont :

$$A = \begin{bmatrix} 0 & 1/14 & 1/35 & 0 \\ 0 & 3/35 & 9/140 & 0 \\ 0 & 9/140 & 3/35 & 0 \\ 0 & 1/35 & 1/14 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 1/480 & 1/1680 & 0 \\ 0 & 1/140 & 3/1120 & 0 \\ 0 & 17/1120 & 1/140 & 0 \\ 0 & 43/1680 & 7/480 & 0 \end{bmatrix} \text{ et } D = \begin{bmatrix} 1/4 & 0 & 0 & 1/20 \\ 1/4 & 0 & 0 & 1/10 \\ 1/4 & 0 & 0 & 3/20 \\ 1/4 & 0 & 0 & 1/5 \end{bmatrix}$$

et le vecteur f est

$$f = \left[-1.9093 \times 10^{-1} \quad -3.6461 \times 10^{-1} \quad -5.1414 \times 10^{-1} \quad -6.3432 \times 10^{-1} \right]^T$$

En substituant dans l'équation matricielle :

$$A + \alpha B + D = \begin{bmatrix} 2.5000 \times 10^{-1} & 6.9345 \times 10^{-2} & 2.7976 \times 10^{-2} & 5.0000 \times 10^{-2} \\ 2.5000 \times 10^{-1} & 7.8571 \times 10^{-2} & 6.1607 \times 10^{-2} & 1.0000 \times 10^{-1} \\ 2.5000 \times 10^{-1} & 4.9107 \times 10^{-2} & 7.8571 \times 10^{-2} & 1.5000 \times 10^{-1} \\ 2.5000 \times 10^{-1} & 2.9762 \times 10^{-3} & 5.6845 \times 10^{-2} & 2.0000 \times 10^{-1} \end{bmatrix}$$

Nous résolvons ensuite le système linéaire :

$$a = (A + \alpha B + D)^{-1} f$$

Il vient que :

$$(A + \alpha B + D)^{-1} = \begin{bmatrix} 1.5941 \times 10^1 & -2.3887 \times 10^1 & 1.5951 \times 10^1 & -4.0052 \times 10^0 \\ -3.1735 \times 10^1 & 7.7140 \times 10^1 & -5.9076 \times 10^1 & 1.3670 \times 10^1 \\ 1.3670 \times 10^1 & -5.9076 \times 10^1 & 7.7140 \times 10^1 & -3.1735 \times 10^1 \\ -2.3339 \times 10^1 & 4.5501 \times 10^1 & -4.0985 \times 10^1 & 1.8823 \times 10^1 \end{bmatrix}$$

Cela implique que :

$$a = \left[5.0777 \times 10^{-3} \quad -3.6469 \times 10^{-1} \quad -6.0178 \times 10^{-1} \quad -3.0015 \times 10^0 \right]^T$$

La solution approximative est donc :

$$y_3(x) = \sum_{i=1}^2 a_i b_i^3(x) \tag{3.1}$$

Ce qui donne :

$$y_3(x) = -3.6469 \times 10^{-1} \times b_1^3(x) - 6.0178 \times 10^{-1} \times b_2^3(x) \tag{3.2}$$

Maintenant, nous allons comparer la solution approchée $y_3(x)$ avec la solution exacte. Les valeurs de x utilisées sont $x_i = i/5$ pour $i = 0, 1, 2, 3, 4, 5$.

Les Erreurs absolues sont données par le vecteur suivant :

$$Error = \left[0 \quad 4.9754 \times 10^{-3} \quad 4.9630 \times 10^{-3} \quad 0 \right]^T$$

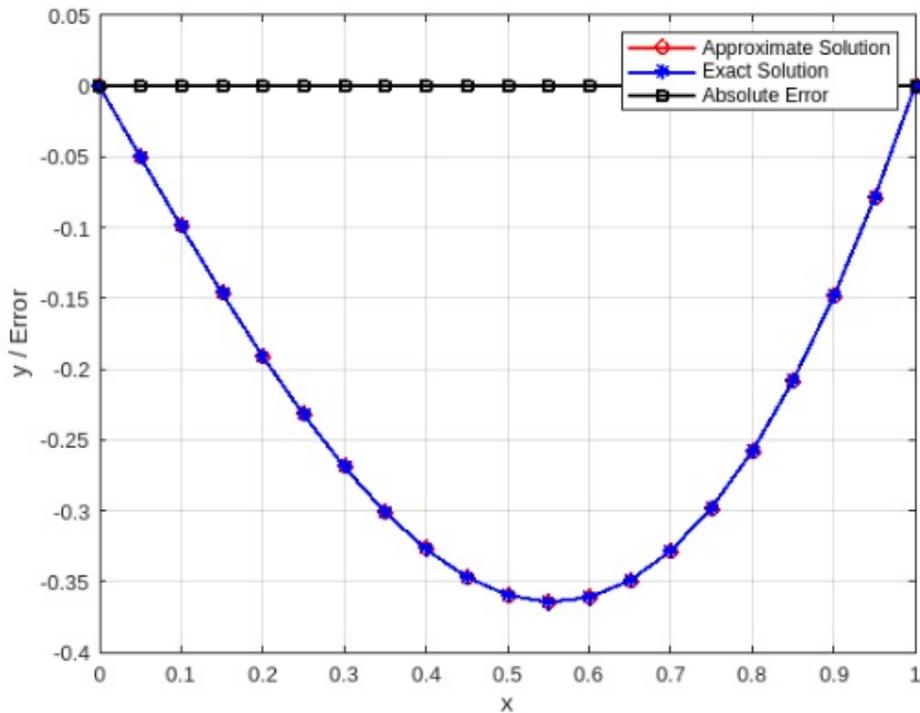
Après avoir détaillé les calculs pour $N = 3$, nous présentons ci-dessous un tableau récapitulatif des erreurs absolues obtenues pour les valeurs $N = 4, 6, 8$. Ces résultats permettent d'évaluer l'évolution de la précision de la méthode de BPG en fonction du degré de l'approximation.

Méthode de Bernstein–Petrov–Galerkin

x_n			
	$N = 4, e_4(x)$	$N = 6, e_6(x)$	$N = 8, e_8(x)$
0	0	0	0
0.2	4.0058×10^{-04}	5.5678×10^{-07}	2.1968×10^{-09}
0.4	3.0607×10^{-05}	1.0077×10^{-06}	1.1980×10^{-10}
0.6	8.8179×10^{-05}	8.8618×10^{-07}	2.3882×10^{-10}
0.8	3.5577×10^{-04}	4.6352×10^{-07}	2.0155×10^{-09}
1	0	0	0

Tableau 3.5 – Erreurs absolues pour différentes valeurs de N

Le tableau 3.5 montre que l'erreur diminue de manière significative à mesure que la valeur de N augmente, ce qui met en évidence la haute précision de la méthode de BPG et sa bonne capacité de convergence.

FIGURE 3.3 – Solution exacte, approximation BPG ($N = 6$) et erreur absolue.

3.4.2 Comparaison avec le solveur `bvp4c`

Afin d'évaluer davantage la performance de la méthode BPG, nous comparons les résultats obtenus de notre exemple avec ceux générés par le solveur numérique `bvp4c` de MATLAB, largement utilisé pour la résolution des problèmes aux limites.

Le solveur `bvp4c`

"`bvp4c`" est un solveur numérique de MATLAB conçu pour résoudre les problèmes aux limites d'équations différentielles ordinaires. Il est capable de traiter des problèmes linéaires ou non linéaires, avec des conditions aux limites posées en des points distincts, souvent sur un intervalle fermé $[a, b]$. Il repose sur une méthode de collocation basée sur des polynômes de degré 4 (méthode implicite de type Lobatto IIIa) [11].

x	La méthode de BPG			Erreur absolue par bvp4c
	$N = 7, e_7(x)$	$N = 8, e_8(x)$	$N = 10, e_{10}(x)$	
0.0	0	0	0	0
0.1	6.1719×10^{-08}	2.0592×10^{-09}	3.2555×10^{-12}	4.5784×10^{-07}
0.2	4.4521×10^{-08}	2.1968×10^{-09}	8.1030×10^{-12}	9.1931×10^{-07}
0.3	7.4560×10^{-08}	1.9865×10^{-09}	1.5305×10^{-11}	1.3881×10^{-06}
0.4	6.0966×10^{-08}	1.1980×10^{-10}	2.7884×10^{-11}	1.8681×10^{-06}
0.5	4.0043×10^{-08}	8.8371×10^{-11}	3.3083×10^{-11}	2.3634×10^{-06}
0.6	6.0607×10^{-08}	2.3882×10^{-10}	2.8437×10^{-11}	2.8782×10^{-06}
0.7	7.8407×10^{-08}	1.8610×10^{-09}	1.8735×10^{-11}	3.4166×10^{-06}
0.8	4.8733×10^{-08}	2.0155×10^{-09}	5.8339×10^{-12}	3.9832×10^{-06}
0.9	6.5736×10^{-08}	1.9575×10^{-09}	5.0326×10^{-13}	4.5831×10^{-06}
1.0	0	0	0	0

Tableau 3.6 – Comparaison des erreurs absolues entre la méthode BPG et le solveur **bvp4c** pour différentes valeurs de N

Le tableau 3.6 montre que la méthode de BPG offre une grande précision, avec des erreurs absolues qui diminuent progressivement lorsque N augmente, ce qui reflète son efficacité numérique. En revanche, la solution obtenue par **bvp4c** présente des erreurs nettement plus élevées, soulignant ainsi la supériorité de la méthode proposée.

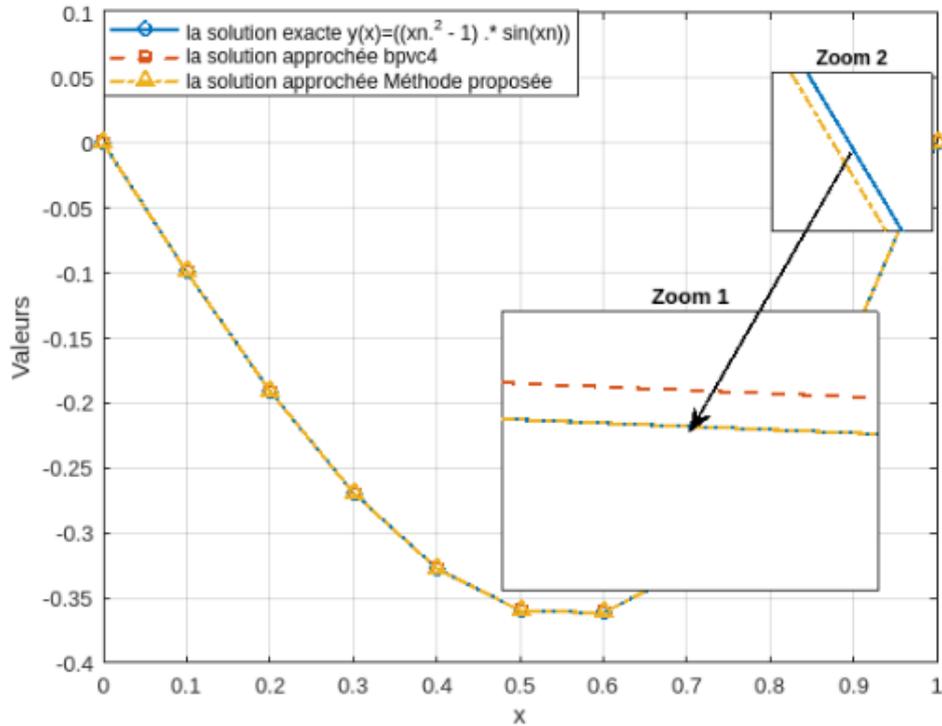


FIGURE 3.4 – Solution exacte, approximation par bpvc4 et approximation par BPG ($N = 8$).

3.5 Étude comparative entre la méthode de collocation de Bernstein et la méthode BPG

Dans cette section, nous appliquons les deux méthodes de Bernstein à un problème aux limites d'ordre quatre avec conditions non homogènes. L'objectif est de comparer les performances des deux approches en termes de précision numérique.

Considérer le problème aux limites suivants :

$$\begin{cases} y^{(4)}(x) - 3y(x) = -2e^x, & 0 \leq x \leq 1, \\ y(0) = 1, \quad y(1) = e, \quad y'(0) = 1, \quad y'(1) = e. \end{cases}$$

La solution exacte de ce problème est donnée par

$$y(x) = e^x$$

Afin de pouvoir appliquer la méthode de BPG, qui requiert des conditions aux limites homogènes, nous procédons à une transformation du problème initial. Cette transformation permet de ramener le problème à un système équivalent avec des conditions aux limites nulles, facilitant ainsi l'application de la méthode BPG.

Transformation du problème

Étape 1 : Construction d'une fonction auxiliaire

Nous définissons un polynôme cubique $w(x) = a_0 + a_1x + a_2x^2 + a_3x^3$ tel que :

$$w(0) = 1, \quad w(1) = e, \quad w'(0) = 1, \quad w'(1) = e.$$

Calculer la dérivée :

$$w'(x) = a_1 + 2a_2x + 3a_3x^2$$

Remplacer les conditions aux limites :

$$\begin{aligned} w(0) &= a_0 = 1, \\ w'(0) &= a_1 = 1, \\ w(1) &= a_0 + a_1 + a_2 + a_3 = e, \\ w'(1) &= a_1 + 2a_2 + 3a_3 = e, \end{aligned}$$

Remplacer les valeurs connues $a_0 = 1, a_1 = 1$ dans le système :

$$\begin{aligned} 1 + 1 + a_2 + a_3 &= e \Rightarrow a_2 + a_3 = e - 2, \\ 1 + 2a_2 + 3a_3 &= e \Rightarrow 2a_2 + 3a_3 = e - 1, \end{aligned}$$

Résolution du système, on obtient :

$$\begin{aligned} a_3 &= -e + 3, \\ a_2 &= 2e - 5, \end{aligned}$$

Étape 2 : Transformation

Définir :

$$v(x) = u(x) - w(x),$$

afin que :

$$v(0) = 0, \quad v(1) = 0, \quad v'(0) = 0, \quad v'(1) = 0.$$

L'équation différentielle transformée devient :

$$v^{(4)}(x) - 3v(x) = -2e^x - (-w^{(4)}(x) - 3w(x)).$$

Puisque $w(x)$ est un polynôme cubique, $w^{(4)}(x) = 0$, donc :

$$v^{(4)}(x) - 3v(x) = -2e^x + 3w(x).$$

Étape 3 : Problème final

$$\begin{cases} v^{(4)}(x) - 3v(x) = -2e^x + 3[1 + x + (2e - 5)x^2 + (-e + 3)x^3], \\ v(0) = 0, \quad v(1) = 0, \quad v'(0) = 0, \quad v'(1) = 0. \end{cases}$$

Les résultats obtenus après l'application des deux méthodes sont regroupés dans le tableau suivant.

N	E_{\max}	
	Méthode de collocation de Bernstein	Méthode de Bernstein–Petrov–Galerkin
5	3.2510×10^{-04}	2.3182×10^{-05}
6	3.8324×10^{-05}	2.4966×10^{-07}
7	3.2673×10^{-06}	3.2153×10^{-08}
8	2.2261×10^{-07}	7.7415×10^{-10}
9	1.2667×10^{-08}	6.8890×10^{-11}
10	6.2092×10^{-10}	3.5579×10^{-11}
11	2.5968×10^{-11}	7.2014×10^{-11}
12	6.7302×10^{-13}	2.7831×10^{-10}

Tableau 3.7 – Comparaison des erreurs maximales E_{\max} entre deux méthodes numériques basées sur les polynômes de Bernstein

Dans le tableau 3.7, On remarque que la méthode Bernstein–Petrov–Galerkin fournit des résultats plus précis par rapport à la méthode de collocation de Bernstein, les valeurs maximales de l'erreur E_{\max} étant nettement plus faibles dans la plupart des cas. De plus, les deux méthodes montrent une diminution régulière de l'erreur lorsque N augmente. Toutefois, cette diminution est plus rapide avec la méthode Bernstein–Petrov–Galerkin, en particulier pour les petites valeurs de N , ce qui témoigne de sa grande efficacité numérique. Cela confirme que cette méthode est mieux adaptée aux problèmes nécessitant une haute précision avec un nombre réduit de termes.

Cependant, un inconvénient notable de la méthode Bernstein–Petrov–Galerkin réside dans la nécessité de transformer préalablement le problème aux limites en un problème à conditions homogènes, ce qui peut complexifier légèrement sa mise en œuvre, notamment lorsque les conditions aux bords sont non homogènes.

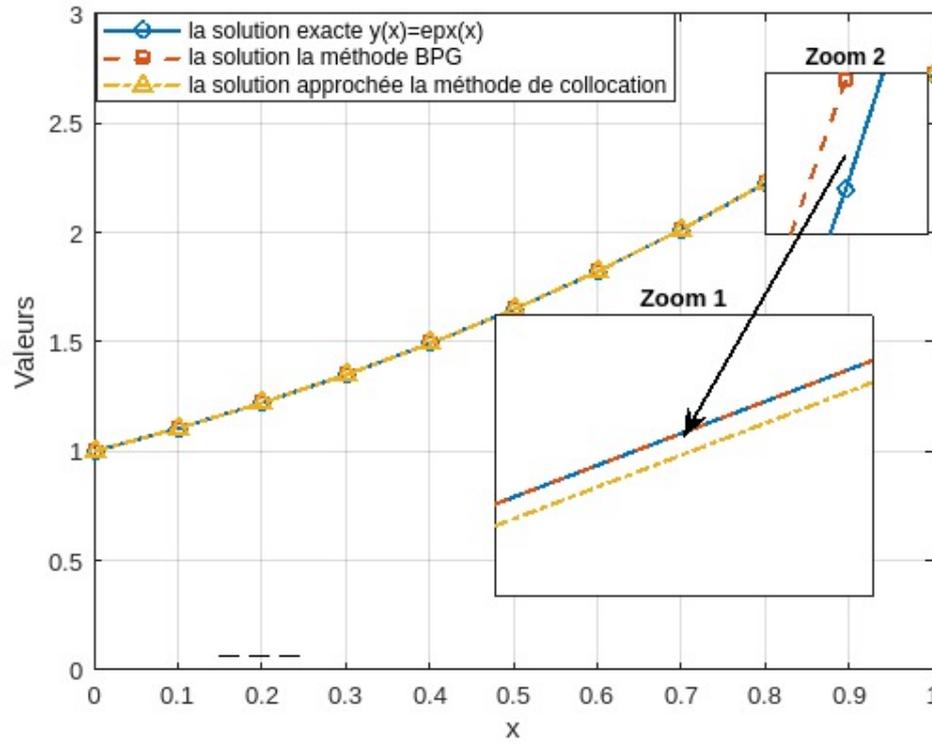


FIGURE 3.5 – Solution exacte, approximation par collocation de Bernstein et approximation par BPG ($N = 8$).

Conclusion

*A*u terme de ce travail, nous avons mis en lumière le rôle essentiel des polynômes de Bernstein dans l'approximation des fonctions et la résolution d'équations différentielles à l'aide de méthodes numériques à la fois précises et flexibles. L'étude théorique a permis de souligner certaines propriétés fondamentales de ces polynômes, telles que leur symétrie et leur positivité, qui en font un choix pertinent pour le développement de méthodes numériques stables et efficaces.

Deux méthodes numériques principales ont été présentées et appliquées : la méthode de collocation et la méthode de Bernstein–Petrov–Galerkin (BPG). Les résultats numériques ont démontré leur grande précision, en particulier lorsqu'ils sont comparés à d'autres méthodes numériques bien connues. Les expériences informatiques ont mis en évidence la capacité de ces approches à fournir des solutions approchées efficaces, avec une convergence rapide et une bonne flexibilité face à différentes conditions aux limites.

Cependant, il convient de noter que la méthode BPG nécessite la transformation préalable des conditions aux limites non homogènes en conditions homogènes, ce qui peut constituer une étape supplémentaire dans le processus de résolution. Malgré cela, son efficacité reste remarquable, notamment pour les problèmes exigeant une haute précision avec un nombre réduit de termes.

Bibliographie

- [1] Akyüz-Daşcıoğlu, A., & Isler Acar, N. (2013). Bernstein collocation method for solving linear differential equations. *Gazi University Journal of Science*, **26**(4), 527–534.
- [2] Barari, A., Omidvar, M., Gholitabar, S., & Ganji, D. D. (2008). An approximate solution for boundary value problems in structural engineering and fluid mechanics. *Mathematical Problems in Engineering*, **13**(2).
- [3] Bernstein, S. N. (1911). Démonstration du théorème de Weierstrass fondée sur le calcul des probabilités. *Communications de la Société Mathématique de Kharkov* (2^e série), **13**, 1–2.
- [4] Doha, E. H., Bhrawy, A. H., & Saker, M. A. (2011). Integrals of Bernstein polynomials : An application for the solution of high even-order differential equations. *Applied Mathematics Letters*, **24**, 559–565.
- [5] Farouki, R. T., & Rajan, V. T. (1988). Algorithms for polynomials in Bernstein form. *Computer Aided Geometric Design*, **5**, 1–26.
- [6] Ibrahim, R. A., & Saad, M. (2022). Application of differential transform method with Adomian polynomial for solving RLC circuits problems and higher order differential equations. *Engineering Research Journal (ERJ)*, **51**(4), 89–95.
- [7] Işık, O. R., Sezer, M., & Güney, Z. (2011). A rational approximation based on Bernstein polynomials for high order initial and boundary value problems. *Applied Mathematics and Computation*, **217**, 9438–9450.
- [8] Lorentz, G.G., (1986). Bernstein polynomials. *Chelsea Publishing, New York, N.Y.*.
- [9] Merris, R. (2003). Combinatorics (2nd ed.). *John Wiley & Sons*.
- [10] Rudin, W. (1976). *Principles of Mathematical Analysis*, 3rd Edition, McGraw-Hill.
- [11] Shampine, L.F., Reichelt, M. W., & Kierzenka, J. (2000) Solving boundary value problems for ordinary differential equations in MATLAB with `bvp4c`, *Tutorial Notes*.

Annexe A : Les codes MATLAB

Code 1 :

```
1      %Collocation de Bernstein
2  clc; clear; close all; % Nettoyer la fenetre de commande, les variables et
      fermer les figures
3
4  %% Etape 1 : Definir les points de la grille
5  n = 5;
6  x_values = linspace(0, 1, n + 1);
7
8  %% Etape 2 : Construire la matrice des polynomes de Bernstein b
9  b = zeros(n+1, n+1);
10 for j = 0:n
11 b(:, j+1) = nchoosek(n, j) * (x_values'.^j) .* ((1 - x_values').^(n - j));
12 end
13
14 %% Etape 3 : Construire la matrice tridiagonale N
15 N = zeros(n+1, n+1);
16 for i = 1:n+1
17 if i > 1
18 N(i, i-1) = - (i-1);
19 end
20 if i <= n+1
21 N(i, i) = 2 * (i-1) - n;
22 end
23 if i < n+1
24 N(i, i+1) = n - (i-1);
25 end
26 end
27
28 %% Etape 4 : Definir les autres matrices
29 A2 = eye(n+1);
30 A1 = diag(x_values);
31 A0 = -2 * eye(n+1);
```

```

32
33 %% Etape 5 : Calculer la matrice du systeme W
34 W = A2 * b * N^2 + A1 * b * N + A0 * b;
35
36
37
38 %% Etape 6 : Definir le vecteur fonction F
39 F = x_values' .* cos(x_values') - 3 * sin(x_values');
40 F = [F(1:n-1); 0; 1];
41
42 %% Etape 7 : Extraire les parties pertinentes de la matrice
43 part0 = W(1:n-1, 1:n+1);
44 part1 = b(1:1, 1:n+1);
45 part2 = (b * N);
46 part2 = part2(1:1, 1:n+1);
47 matrice = [part0; part1; part2];
48
49 %% Etape 8 : Calculer l'inverse de la matrice
50 inv_mat = inv(matrice);
51
52 %% Etape 9 : Resoudre Y
53 Y = inv_mat * F;
54
55 %% Etape 10 : Solution approchee
56 B = @(x,i) nchoosek(n, i) * x.^i .* (1 - x).^(n - i);
57 Yap = @(x) sum(arrayfun(@(i) Y(i+1) * B(x, i), 0:n));
58 m = 5; h = 1/m; xn = 0:h:1;
59 for i = 1:m+1
60 Yapp(i) = Yap(xn(i));
61 end
62
63 %% Etape 11 : Calculer l'erreur entre la solution exacte et approchee
64 y_exact = sin(xn');
65 xn = 0:0.2:1;
66 Error = abs(y_exact - Yapp');
67 E_mean = mean(Error);
68 E_max = max(Error);
69
70 %% Etape 12 : Afficher les erreurs
71 fprintf('E_mean : %.6e\n', E_mean);
72 fprintf('E_max : %.6e\n', E_max);
73
74 % Tracer les resultats
75 figure;
76 plot(x_values, sin(x_values), 'v--', 'LineWidth', 2); hold on;
77 plot(xn, Yapp, 'r-o', 'LineWidth', 1.5);
78 plot(xn, abs(y_exact - Yapp'), 'k-s', 'LineWidth', 1.5);
79 legend('Solution exacte', 'Solution approchee', 'Erreur');
80 xlabel('x');

```

```

81 ylabel('Valeur');
82 title('Approximation par polynomes de Bernstein vs Solution exacte');
83 grid on;

```

Code 2 :

```

1      %Bernstein Petrov Galerkin
2      clc; clear all;
3
4      % Parameters
5      N = 3; m = 1; a = 0; b = 1; Alfa = -1;
6
7      % Define Bernstein polynomial basis function B_{i,n}(x)
8      B = @(i,x) nchoosek(N, i) * x.^i .* (1 - x).^(N - i);
9      g = @(x) (4 - 2*x.^2).*sin(x) + 4*x.*cos(x); % Define f(x)
10
11     % Convert g to symbolic for integration
12     syms x
13     g_sym = (4 - 2*x.^2)*sin(x) + 4*x*cos(x);
14
15     % Compute h(x) = double integral of g(x)
16     h_sym = int(int(g_sym, x), x); % double primitive
17     h_handle = matlabFunction(h_sym); % convert to function handle
18
19     % Define integrand Fi(i) = ? h(x) * B_{i,N}(x) dx
20     Fi = @(i) integral(@(x) h_handle(x) .* B(i,x), a, b);
21
22     % Compute vector F for i = m:N-m
23     F = arrayfun(Fi, 0:N); % Or use 0:N if you want full vector
24
25     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
26
27     % Matrix A
28     A = zeros(N+1, N+1);
29     for k = 0:N
30         for j = m:N-m
31             A(k+1, j+1) = ((b - a) / (2*N + 1)) * ...
32                 (nchoosek(N, j) * nchoosek(N, k)) / ...
33                 nchoosek(2*N, k + j);
34         end
35     end
36
37     % Matrix BB
38     BB = zeros(N+1, N+1);
39     for k = 0:N
40         for j = m:N-m
41             % Sum term for BB

```

```

42     sum_term = 0;
43     for i = j + 2*m : N + 2*m
44         numerator = nchoosek(N + 2*m, i) * ...
45         nchoosek(i - j - 1, 2*m - 1);
46         denominator = nchoosek(2*N + 2*m, k + i);
47         sum_term = sum_term + (numerator / denominator);
48     end
49     BB(k+1, j+1) = (((b - a)^(2*m + 1) * factorial(N) * ...
50     nchoosek(N, k))) / ...
51     ((factorial(N + 2*m) * (2*N + 2*m + 1))) * sum_term;
52     end
53     end
54
55     % Matrix D
56     D = zeros(N+1, N+1);
57     for k = 0:N
58         for j = 0:N
59             if j < m
60                 D(k+1, j+1) = (nchoosek(N, k) / nchoosek(N + j, k + j)) * ...
61                 (b - a) / (N + j + 1);
62             elseif j > N - m
63                 D(k+1, j+1) = (nchoosek(N, k) / ...
64                 nchoosek(j + 2*m - 1, k + j - N + 2*m - 1)) * ...
65                 (b - a) / (j + 2*m);
66             else
67                 D(k+1, j+1) = 0;
68             end
69         end
70     end
71
72     % Final matrix
73     A + Alfa * BB + D;
74
75     %% The vector aa
76     aa = inv(A + Alfa * BB + D) * F'; % Solve for aa
77
78     %% Approximate solution
79     Yap = @(x) sum(arrayfun(@(i) aa(i+1) * B(i,x), m:N-m));
80     n = 20; h = 2/n; xn = 0:h:1;
81     Yapp = arrayfun(Yap, xn); % Approximate solution vector
82
83     % Compute Error Between Exact and Approximate Solution
84     y_exact = ((xn.^2 - 1) .* sin(xn))'; % Exact solution y(x) = sin(x)
85     Error = abs(y_exact - Yapp');
86     Ep = max(abs(y_exact - Yapp')); % Max error
87     Er = max(abs((y_exact - Yapp') ./ y_exact)); % Relative error
88
89     % Compute mean and max error
90     E_mean = mean(Error);

```

```

91     E_max = max(Error);
92
93     % ===== PLOTS =====
94     figure;
95     plot(xn, Yapp, 'r-o', 'LineWidth', 1.5); hold on;
96     plot(xn, y_exact, 'b-*', 'LineWidth', 1.5);
97     plot(xn, abs(y_exact - Yapp), 'k-s', 'LineWidth', 1.5);
98     legend('Approximate Solution', 'Exact Solution', 'Absolute Error');
99     xlabel('x');
100    ylabel('y / Error');
101    grid on;

```

Code 3 :

```

1         %Exemple 2 : Par le solver bvp4c
2     function solve_bvp2_with_bvp4c
3     % Resolution de l'exemple 1 avec bvp4c
4
5     %% Maillage initial
6     xn = linspace(0, 1, 11);
7
8     %% Estimation initiale de la solution
9     solinit = bvpinit(xn, @guess);
10
11    %% Resolution avec bvp4c
12    sol = bvp4c(@odefun, @bcfun, solinit);
13
14    %% Solution exacte
15    uex = (xn.^2 - 1) .* sin(xn);
16
17    %% Erreur absolue
18    Erro = abs(sol.y(1,:) - uex);
19
20    end
21
22    %% Systeme d'equations differentielles d'ordre 1
23    function dydx = odefun(x, y)
24    dydx = zeros(2,1);
25    dydx(1) = y(2); % y1' = y2
26    dydx(2) = y(1) + (4 - 2*x^2)*sin(x) + 4*x*cos(x); % y2' = u + membre droit
27    end
28
29    %% Conditions aux limites
30    function res = bcfun(ya, yb)
31    res = [ya(1); % u(0) = 0
32    yb(1)]; % u(1) = 0
33    end

```

```
34
35 %% Estimation initiale
36 function g = guess(x)
37 g = [sin(pi*x);           % estimation pour u(x)
38      pi*cos(pi*x)];      % estimation pour u'(x)
39 end
```

تتناول هذه المذكرة استخدام كثيرات الحدود من نوع برنشتاين في حل المعادلات التفاضلية العادية. تم تطوير طريقتين عدديتين: طريقة التجميع وطريقة بيتروف-غالركين ، وتم اختبارهما باستخدام برنامج ماتلاب. أظهرت النتائج دقة عالية وسرعة ، مقارنة بطرق عددية أخرى، كما أثبتت الطريقتان فعاليتهما في التعامل مع مختلف الشروط الحدية.

الكلمات المفتاحية: برنشتاين ، طرق عددية ، معادلات تفاضلية ، التجميع ، بيتروف-غالركين ، ماتلاب.

Abstract

This thesis explores the use of Bernstein polynomials for solving ordinary differential equations. Two numerical methods were developed: the Collocation method and the Bernstein–Petrov–Galerkin (BPG) method. These methods were implemented using MATLAB and tested on various examples. The results showed high accuracy and fast convergence compared to other numerical approaches. Both methods also proved effective in handling different boundary conditions.

Keywords: Bernstein, numerical methods, differential equations, Collocation, Petrov–Galerkin, MATLAB.

Résumé

Ce mémoire étudie l'utilisation des polynômes de Bernstein pour résoudre des équations différentielles ordinaires. Deux méthodes numériques ont été développées : la méthode de collocation et la méthode Bernstein–Petrov–Galerkin (BPG). Leur performance a été évaluée à l'aide de MATLAB sur plusieurs exemples, avec des résultats montrant une haute précision et une bonne rapidité , notamment comparées à d'autres méthodes classiques. Ces approches se sont révélées efficaces pour divers types de conditions aux limites.

Mots-clés : Bernstein, méthodes numériques, équations différentielles, collocation, Petrov–Galerkin, MATLAB.