



UNIVERSITE MOHAMMED VI

POLYTECHNIQUE

DEPARTEMENT CSEHS

STAGE S2

Aspects numériques: procédure d'assemblage rapide

Réalisé par :

LAMSAHEL NOUREDDINE

Sous la direction des professeurs :

A. EL-KACIMI

A.RATNANI

Contents

| | | |
|-----|---|----|
| 0.1 | Introduction | 1 |
| 0.2 | Problématique et élément fini de Bernstein Bézier | 2 |
| 0.3 | Matrice de masse | 6 |
| 0.4 | Matrice de rigidité | 10 |
| 0.5 | Conclusion | 14 |

0.1 Introduction

La méthode des éléments finis joue un rôle très important dans une large ensemble des problèmes et applications en acoustique, optique et électromagnétique, en particulier les problèmes d'ondes courtes régis par l'équation de Helmholtz qui modélise de nombreux problèmes d'ondes dans le domaine fréquentiel, y compris la propagation, la guidage, le rayonnement et la diffusion des ondes. L'erreur relative hp -FM pour ces types de problèmes montre que pour avoir une bonne approximation il faut utiliser des polynômes d'approximation d'ordre élevé pour arriver à des résultats ressemblent au problème physique sur lequel on travail.

L'approximation d'ordre élevé donne des bonnes résultats, mais le problème est que quand le p augment, le temps CPU pour assembler la matrice de masse ou la matrice de rigidité augment aussi. Pour ce faire, on aura besoin des procédures d'assemblage rapide ou minimal, par exemple si on utilise les bases de Lagrange il faut nécessairement utiliser des règles de quadrature directe pour approximer les deux matrices. Mais chacun des deux matrices contiennent $\binom{p+2}{p}^2$ entrées (en travail dans la dimension 2) ce qui signifie que même si chaque entrée pouvait être calculé en $O(1)$ le coût global de calcul est au moins $O(p^4)$. Bien sur si la transformation entre l'élément de référence \hat{T} et le triangle du maillage T n'est pas affine, on a besoin de plus que $O(1)$ pour calculer chaque entrée. De plus tout règle de quadrature doit utiliser ou moins $O(p^2)$ point de quadrature, ce qui signifie qu'une implémentation coût $O(p^2)$ par entrée et donc le coût global pour l'assemblage de

chaque matrice est $O(p^6)$.

Le but de ce stage est de montrer que avec les éléments finis de Bernstein-Bézer on peut atteindre une complexité minimal de $O(p^5)$. Pour ce faire on va commencer par la construction de base de Bernstein-Bézer qui basé sur les polynôme de Bernstein, puis on attaque le problème de minimisation de complexité pour chacun des deux matrices, basant sur des résultats fondamentales d'intégration comme le théorème de changement de variable, le théorème de Fubini et aussi la méthode numérique d'approximation des intégrales de quadrature de Gauss-Jacobi.

0.2 Problématique et élément fini de Bernstein Bézier

Soient Ω un ouvert borné de \mathbb{R}^2 et $\mathbf{T}_h = \{\mathbf{T}\}_{ne}$ un maillage de Ω , avec ne est le nombre total des éléments de maillage et $h = \max_{1 \leq i \leq ne} h_{\mathbf{T}}$, $h_{\mathbf{T}} = \text{diam}(\mathbf{T})$. Un élément $\mathbf{T} \in \mathbf{T}_h$ est usuellement construit à partir d'un élément de référence $\hat{\mathbf{T}}$ (triangle) à l'aide d'une transformation inversible.

Pour l'équations de Helmholtz, l'erreur relative hp -FM dans le H^1 -seminorme, sur un hp -maillage uniforme est donnée par:

$$\frac{|u - u_h|_1}{|u|_1} \leq C_1 \left(\frac{kh}{2p} \right)^p + C_2 k \left(\frac{kh}{2p} \right)^{2p},$$

où C_1 et C_2 sont indépendants de kh .

Avec k est le nombre d'onde: grandeur proportionnelle à l'inverse de la longueur d'onde, avoir lieu dans un certain nombre d'applications physiques, par exemple les problèmes de diffusion des ondes et d'interaction fluide-solide. Le premier terme de l'erreur relative hp -FM est "le meilleur erreur d'approximation" et le deuxième terme c'est l'erreur de pollution numérique. En remarque que, si le nombre d'onde est élevé (problèmes d'ondes courtes) alors l'erreur est dominée par le deuxième terme, de plus si $\frac{kh}{2p}$ est assez petit, l'erreur de pollution pour $p \geq 2$ est petite.

Soit maintenant un triangle $\langle q_1, q_2, q_3 \rangle$ de \mathbb{R}^2 , noté \mathbf{T} , alors $\forall x \in \mathbb{R}^2$, il existe un unique triple $(\lambda_1, \lambda_2, \lambda_3)$, tel que :

$$x = \sum_{i=1}^3 \lambda_i q_i \quad \text{et} \quad \sum_{i=1}^3 \lambda_i = 1.$$

$\lambda_1, \lambda_2, \lambda_3$ s'appellent les coordonnées barycentriques, qui sont beaucoup plus utiles que les coordonnées cartésiennes pour travailler avec des polynômes sur des triangles.

En effet:

On a

$$x = \sum_{i=1}^3 \lambda_i q_i \quad \text{et} \quad \sum_{i=1}^3 \lambda_i = 1,$$

donc

$$\begin{pmatrix} 1 & 1 & 1 \\ q_1 & q_2 & q_3 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} = \begin{pmatrix} 1 \\ x \end{pmatrix}.$$

Notons $M = \begin{pmatrix} 1 & 1 & 1 \\ q_1 & q_2 & q_3 \end{pmatrix}$. Alors M est inversible et de plus $\det(M) = 2\text{mes}(T)$.

N.B: Les sommets q_1, q_2, q_3 sont numérotés dans le sens anti-horaire.

D'où le résultat.

De plus :

$$\lambda_1 = \frac{1}{2\text{mes}(T)} \begin{vmatrix} 1 & 1 & 1 \\ x & q_2 & q_3 \end{vmatrix}, \lambda_2 = \frac{1}{2\text{mes}(T)} \begin{vmatrix} 1 & 1 & 1 \\ q_1 & x & q_3 \end{vmatrix} \text{ et } \lambda_3 = \frac{1}{2\text{mes}(T)} \begin{vmatrix} 1 & 1 & 1 \\ q_1 & q_2 & x \end{vmatrix}. \quad (1)$$

Ce qui montre que si $x \in T$, alors $\lambda_i \geq 0$ et $\lambda_i(q_i) = 1$, pour $i \in \{1, 2, 3\}$. D'autre part $\lambda_i = 0$ si et seulement si $x \in [q_j, q_k]$, $i, j, k \in \{1, 2, 3\}$ $j, k \neq i$.

Soient \mathbb{P}_p l'ensemble des polynôme dans \mathbb{R}^2 de degré inférieur ou égale à $p \in \mathbb{N}^*$ et $I_p = \{\alpha \in \mathbb{N}^3, |\alpha| = p\}$, tel que $|\alpha| = \alpha_1 + \alpha_2 + \alpha_3$, alors pour tout $x \in \mathbb{R}^2$, on a

$$1 = (\lambda_1 + \lambda_2 + \lambda_3)^p = \sum_{\alpha \in I_p} \binom{p}{\alpha} \lambda^\alpha, \quad (2)$$

$$\text{où } \lambda = (\lambda_1, \lambda_2, \lambda_3), \quad \binom{p}{\alpha} = \frac{p!}{\alpha!}, \quad \alpha! = \alpha_1! \alpha_2! \alpha_3! \quad \text{et} \quad \lambda^\alpha = \lambda_1^{\alpha_1} \lambda_2^{\alpha_2} \lambda_3^{\alpha_3}.$$

Posons

$$B_\alpha^p = \binom{p}{\alpha} \lambda^\alpha, \quad \alpha \in I_p \quad \text{et} \quad \mathbb{B}^p = \{B_\alpha^p, \alpha \in I_p\}.$$

D'après (1) on remarque que chacun des λ_i , $i \in \{1, 2, 3\}$ est un polynôme linéaire en x_1 et x_2 avec $x = (x_1, x_2)$, il s'ensuit que B_α^p est un polynôme de degré p . Nous appelons ces polynômes les polynômes de base de Bernstein de degré p par rapport à T et \mathbb{B}^p est l'ensemble des polynôme de Bernstein relative à T , d'après (1) et (2), on a

$$0 \leq B_\alpha^p(x) \leq 1, \quad \alpha \in I_p \quad \text{et} \quad x \in T.$$

Les polynômes de Bernstein sont initialement introduit par Sergei Natanovich Bernstein pour faciliter une preuve constructive du théorème d'approximation de Weierstrass mais de plus ces polynômes ayant de nombreuses propriétés intéressantes, par exemple:

Théorème

-
- 1) $B_\alpha^n B_\beta^m = \frac{\binom{\alpha + \beta}{\alpha}}{\binom{n + m}{m}} B_{\alpha + \beta}^{n+m}, \quad \alpha \in I_n \text{ et } \beta \in I_m.$
 - 2) $\int_T B_\alpha^p dx = \frac{\text{mes}(T)}{\binom{p+2}{2}}, \quad \alpha \in I_p.$
 - 3) \mathbb{B}^p forme une base pour \mathbb{P}_p .

Démonstration

- 1) Soit $\alpha \in I_n$ et $\beta \in I_m$, alors

$$\begin{aligned}
 \frac{\binom{\alpha + \beta}{\alpha}}{\binom{n + m}{m}} B_{\alpha + \beta}^{n+m} &= \frac{\prod_{k=1}^3 \binom{\alpha_k + \beta_k}{\alpha_k}}{\frac{(m+n)!}{m!n!}} \frac{(m+n)!}{(\alpha + \beta)!} \lambda^{\alpha + \beta} \\
 &= \frac{m!n!}{\prod_{k=1}^3 (\alpha_k + \beta_k)!} \prod_{k=1}^3 \frac{(\alpha_k + \beta_k)!}{\alpha_k! \beta_k!} \lambda^\alpha \lambda^\beta \\
 &= \frac{m!n!}{\alpha! \beta!} \lambda^\alpha \lambda^\beta \\
 &= B_\alpha^n B_\beta^m.
 \end{aligned}$$

- 2) Soit $\alpha \in I_p$, on a

$$\begin{aligned}
 \int_T B_\alpha^p dx &= \int_T \frac{p!}{\alpha!} \lambda_1^{\alpha_1} \lambda_2^{\alpha_2} \lambda_3^{\alpha_3} dx \\
 &= \frac{p!}{\alpha!} \int_T \lambda_1^{\alpha_1} \lambda_2^{\alpha_2} (1 - \lambda_1 - \lambda_2)^{\alpha_3} dx,
 \end{aligned}$$

d'autre part $x = \sum_{i=1}^3 \lambda_i q_i = \lambda_1(q_1 - q_3) + \lambda_2(q_2 - q_3) + q_3$, alors

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} q_1 - q_3 & q_2 - q_3 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} + q_3.$$

Par suite

$$\begin{aligned}
\int_{\mathbb{T}} B_{\alpha}^p dx &= \frac{p!}{\alpha!} \int_0^1 \int_0^{1-\lambda_1} \lambda_1^{\alpha_1} \lambda_2^{\alpha_2} (1 - \lambda_1 - \lambda_2)^{\alpha_3} 2 \text{mes}(\mathbb{T}) d\lambda_2 d\lambda_1 \\
&= 2 \text{mes}(\mathbb{T}) \frac{p!}{\alpha!} \int_0^1 \lambda_1^{\alpha_1} (1 - \lambda_1)^{\alpha_2 + \alpha_3 + 1} \int_0^{1-\lambda_1} \left(\frac{\lambda_2}{1 - \lambda_1} \right)^{\alpha_2} \left(1 - \frac{\lambda_2}{1 - \lambda_1} \right)^{\alpha_3} \frac{d\lambda_2}{1 - \lambda_1} d\lambda_1 \\
&= 2 \text{mes}(\mathbb{T}) \frac{p!}{\alpha!} \int_0^1 t_1^{\alpha_1} (1 - t_1)^{\alpha_2 + \alpha_3 + 1} dt_1 \int_0^1 t_2^{\alpha_2} (1 - t_2)^{\alpha_3} dt_2 \\
&= 2 \text{mes}(\mathbb{T}) \frac{p!}{\alpha!} \frac{\alpha_1! (\alpha_2 + \alpha_3 + 1)!}{(|\alpha| + 2)!} \frac{\alpha_2! \alpha_3!}{(\alpha_2 + \alpha_3 + 1)!} \\
&= \frac{2 \text{mes}(\mathbb{T})}{(p + 2)(p + 1)} \\
&= \frac{\text{mes}(\mathbb{T})}{\binom{d + 2}{2}}.
\end{aligned}$$

3) On a $\text{Card}(\mathbb{B}^p) = \dim(\mathbb{P}_p) = \binom{p + 2}{2}$, alors il suffit de montrer que \mathbb{B}^p est libre ($\mathbb{B}^p \subseteq \mathbb{P}_p$).

Soit $c_{\alpha} \in \mathbb{R}$ avec $\alpha \in \mathbb{I}_p$, tel que:

$$\sum_{\alpha \in \mathbb{I}_d} c_{\alpha} B_{\alpha}^p = 0.$$

Alors

$$\sum_{\alpha \in \mathbb{I}_p} c_{\alpha} \binom{p}{\alpha} \lambda^{\alpha} = 0.$$

D'autre part $\forall x \in \mathbb{R}^2$, on a $\lambda_3(x) = 0 \iff x \in (q_1, q_2)$, soient alors $x \in (q_1, q_2)^C$ et $b_1(x) = \left(\frac{\lambda_1}{\lambda_3} \right)$ et $b_2(x) = \left(\frac{\lambda_2}{\lambda_3} \right)$, donc

$$\sum_{\alpha \in \mathbb{I}_p} c_{\alpha} \binom{p}{\alpha} b_1^{\alpha_1} b_2^{\alpha_2} = 0,$$

comme $\overline{(q_1, q_2)^C} = \mathbb{R}^2$ et le couple (b_1, b_2) est linéairement indépendant, alors $c_{\alpha} = 0$.

D'où \mathbb{B}^p forme une base pour $\mathbb{P}_p \otimes$

Ce qui signifie que $\forall P \in \mathbb{P}_p$ il existe un unique $c_{\alpha} \in \mathbb{R}$ avec $\alpha \in \mathbb{I}_p$, tel que : $P = \sum_{\alpha \in \mathbb{I}_p} c_{\alpha} B_{\alpha}^p$.

Soit alors

$$\varphi_{\alpha} : \begin{array}{ccc} \mathbb{P}_p & \longrightarrow & \mathbb{R} \\ P & \longmapsto & c_{\alpha} \end{array}, \quad \alpha \in \mathbb{I}_p.$$

Posons $\sum_p = \{\varphi_\alpha, \quad \alpha \in I_p\}$, donc le triple $(T, \sum_p, \mathbb{P}_p)$ est un élément finie.

0.3 Matrice de masse

Soient $M_{\alpha,\beta}^p$ la matrice de masse définie par :

$$M_{\alpha,\beta}^p = \int_T B_\alpha^p(F_T^{-1}(x)) B_\beta^p(F_T^{-1}(x)) dx$$

et $\hat{T} = \{x \in \mathbb{R}^2, \quad 0 \leq x_1 \leq 1, \quad 0 \leq x_2 \leq 1 - x_1\}$, tel que $T = F_T(\hat{T})$ où F_T est un difféomorphisme de classe C^1 .

Les coordonnées barycentriques relative à \hat{T} sont :

$$\hat{\lambda}_1 = x_1, \quad \hat{\lambda}_2 = x_2 \quad \text{et} \quad \hat{\lambda}_3 = 1 - x_1 - x_2$$

Si de plus F_T est affine, alors T a un bord droit et $\forall x \in \mathbb{R}^2, \quad F_T(x) = Ax + b$, avec $A \in M_2(\mathbb{R})$ et $b \in \mathbb{R}^2$, donc $DF_T(x) = A, \quad \forall x \in \mathbb{R}^2$ ce qui implique:

$$M_{\alpha,\beta}^p = \int_{\hat{T}} B_\alpha^p(x) B_\beta^p(x) |\det(A)| dx.$$

D'autre part $q_1 - q_2 = A(\hat{q}_1 - \hat{q}_2)$ et $q_1 - q_3 = A(\hat{q}_1 - \hat{q}_3)$, avec $\hat{T} = \langle \hat{q}_1, \hat{q}_2, \hat{q}_3 \rangle$ et $\hat{q}_1 = (1, 0), \quad \hat{q}_2 = (0, 1), \quad \hat{q}_3 = (0, 0)$.

Par suite $(q_1 - q_2 \quad q_1 - q_3) = A(\hat{q}_1 - \hat{q}_2 \quad \hat{q}_1 - \hat{q}_3)$, donc

$$M_{\alpha,\beta}^p = \int_{\hat{T}} B_\alpha^p B_\beta^p \frac{\text{mes}(T)}{\text{mes}(\hat{T})} dx,$$

d'après le théorème on a

$$M_{\alpha,\beta}^p = \frac{\text{mes}(T)}{\binom{2p+2}{2}} C_{\alpha,\beta},$$

$$\text{où } C_{\alpha,\beta} = \frac{\binom{\alpha+\beta}{\alpha}}{\binom{2p}{p}}.$$

Mais en générale F_T n'est pas affine, c'est le cas si une arête de T est courbé, alors

$$M_{\alpha,\beta}^p = \int_{\hat{T}} B_\alpha^p(x) B_\beta^p(x) |J_T(x)| dx,$$

où J_T désigne le déterminant jacobien de F_T .

Soit F l'application définie par:

$$F : \begin{array}{ccc} [0, 1]^2 & \longrightarrow & \hat{T} \\ (x_1, x_2) & \longmapsto & (x_1, (1 - x_1)x_2) \end{array},$$

F s'appelle la transformation de Duffy et est un difféomorphisme de classe C^1 , ce qui implique que

$$M_{\alpha, \beta}^p = \int_{[0, 1]^2} B_{\alpha}^p(F(x)) B_{\beta}^p(F(x)) |J_T(F(x))| (1 - x_1) dx.$$

D'autre part on a $B_{\alpha}^p(F(x)) = B_{\alpha_1}^p(x_1) B_{\alpha_2}^{p-\alpha_1}(x_2)$, $\forall x \in [0, 1]^2$ c'est la structure tensorielle de la base de Bernstein Bézier.

En effet:

Soit $x \in [0, 1]^2$, alors

$$\begin{aligned} B_{\alpha}^p(F(x)) &= B_{\alpha}^p(x_1, (1 - x_1)x_2) \\ &= \binom{p}{\alpha} x_1^{\alpha_1} (1 - x_1)^{\alpha_2} x_2^{\alpha_2} (1 - x_1 - (1 - x_1)x_2)^{\alpha_3} \\ &= \frac{p!}{\alpha_1! \alpha_2! \alpha_3!} x_1^{\alpha_1} (1 - x_1)^{\alpha_2} x_2^{\alpha_2} (1 - x_1)^{p-\alpha_1-\alpha_2} (1 - x_2)^{p-\alpha_1-\alpha_2} \\ &= \frac{p!}{\alpha_1! \alpha_2! \alpha_3!} x_1^{\alpha_1} (1 - x_1)^{p-\alpha_1} x_2^{\alpha_2} (1 - x_2)^{p-\alpha_1-\alpha_2} \\ &= \frac{p!}{\alpha_1! (p - \alpha_1)!} x_1^{\alpha_1} (1 - x_1)^{p-\alpha_1} \frac{(p - \alpha_1)!}{\alpha_2! (p - \alpha_1 - \alpha_2)!} x_2^{\alpha_2} (1 - x_2)^{p-\alpha_1-\alpha_2} \\ &= B_{\alpha_1}^p(x_1) B_{\alpha_2}^{p-\alpha_1}(x_2), \end{aligned}$$

d'où le résultat.

Ce qui donne :

$$\begin{aligned} M_{\alpha, \beta}^p &= \int_{[0, 1]^2} B_{\alpha_1}^p(x_1) B_{\alpha_2}^{p-\alpha_1}(x_2) B_{\beta_1}^p(x_1) B_{\beta_2}^{p-\beta_1}(x_2) |J_T(F(x))| (1 - x_1) dx \\ &= \int_0^1 B_{\alpha_2}^{p-\alpha_1}(x_2) B_{\beta_2}^{p-\beta_1}(x_2) \left(\int_0^1 B_{\alpha_1}^p(x_1) B_{\beta_1}^p(x_1) |J_T(F(x))| (1 - x_1) dx_1 \right) dx_2. \end{aligned}$$

Pour l'approximation de $M_{\alpha, \beta}^p$, rappelons la méthode d'approximation des intégrales de quadrature de Gauss-Jacobi suivant:

Soit f une application de $[0, 1]$ à valeurs dans \mathbb{R} , alors

$$\int_0^1 x^{\beta} (1 - x)^{\alpha} f(x) dx \approx (0.5)^{\alpha+\beta+1} \sum_{i=1}^q w_i^{(\alpha, \beta)} f(\eta_i^{(\alpha, \beta)}),$$

avec $\eta^{(\alpha, \beta)} = 0.5 + 0.5t^{(\alpha, \beta)}$, où $t^{(\alpha, \beta)}$ et $w^{(\alpha, \beta)}$ sont respectivement les noeuds et les poids de la méthode de Gauss-Jacobi, $\eta^{(\alpha, \beta)}$ sont les noeuds pour l'intervalle $[0, 1]$.

Alors

$$M_{\alpha,\beta}^p \approx (0.5) \sum_{s=1}^q w_s^{(0,0)} B_{\alpha_2}^{p-\alpha_1}(\eta_s^{(0,0)}) B_{\beta_2}^{p-\beta_1}(\eta_s^{(0,0)}) \left(\int_0^1 B_{\alpha_1}^p(x_1) B_{\beta_1}^p(x_1) |J_T(F(x_1, \eta_s^{(0,0)}))| (1-x_1) dx_1 \right)$$

$$\approx (0.5)^3 \sum_{s=1}^q w_s^{(0,0)} B_{\alpha_2}^{p-\alpha_1}(\eta_s^{(0,0)}) B_{\beta_2}^{p-\beta_1}(\eta_s^{(0,0)}) \left(\sum_{r=1}^q w_r^{(1,0)} B_{\alpha_1}^p(\eta_r^{(1,0)}) B_{\beta_1}^p(\eta_r^{(1,0)}) |J_T(F(\eta_r^{(1,0)}, \eta_s^{(0,0)}))| \right)$$

† Cas de F_T affine: On suppose que la transformation F_T est affine et on calcule la matrice de masse par les trois méthodes, les 5 figures suivantes montrent l'erreur de calcul et le temps CPU pour calculer la matrice de masse en fonction de degré p de polynôme de Berntienavec Pyccl, Numba et Python pure :

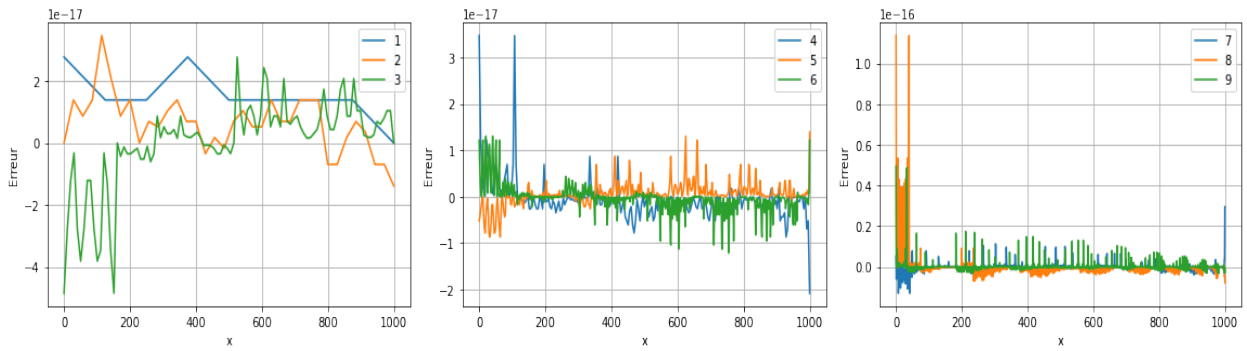


Figure 1 : Erreur entre la méthode analytique et la méthode de factorisation pour $p=1, 2, \dots, 9$.

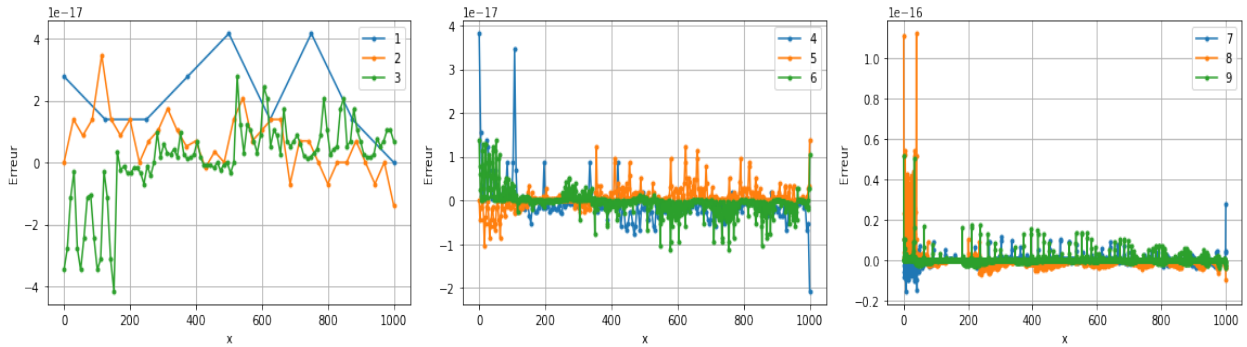


Figure 2 : Erreur entre la méthode analytique et la méthode de quadrature directe pour $p=1, 2, \dots, 9$.

Pyccl est un compilateur statique pour Python3 qui utilise Fortran ou C comme backend language. Il est créé par M.Ratnani Ahmad. L'objectif de Pyccl est de fournir un moyen simple de générer automatiquement du code parallèle de bas niveau et il permet également de convertir un code Python en code Fortran ou C, or Numba est un compilateur pour les fonctions Python qu'il transforme les fonctions Python en code machine optimisé au moment de l'exécution à l'aide de compilateur LLVM standard, de plus les algorithmes compilés par Numba en Python peuvent approcher les vitesses de Fortran pour des cas

particuliers.

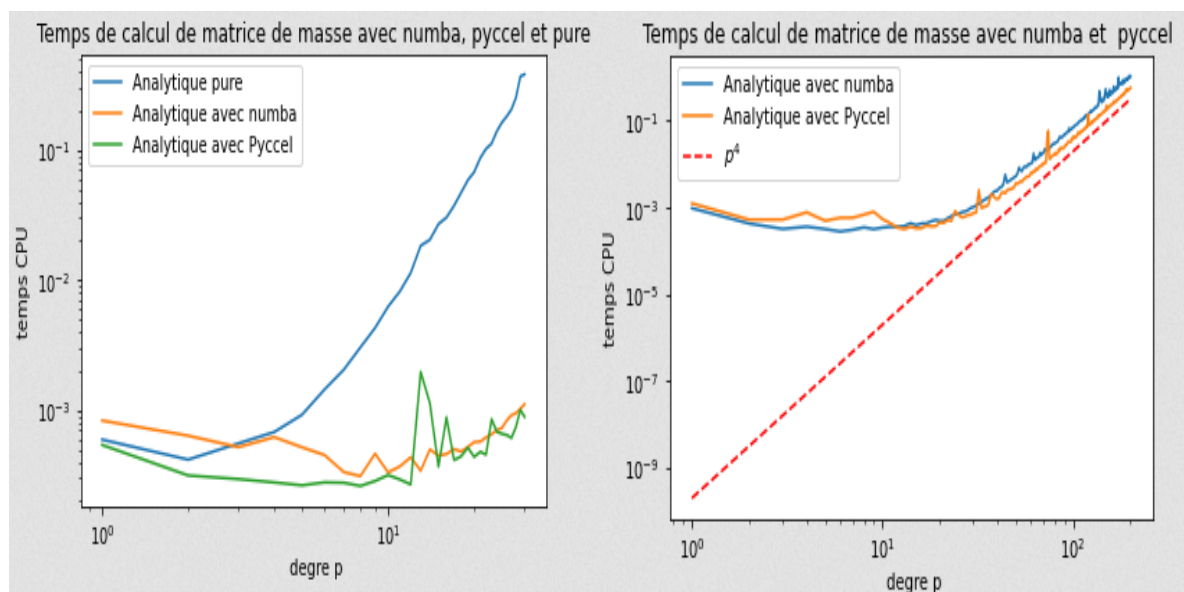


Figure 3 : Temps CPU pour la méthode analytique.

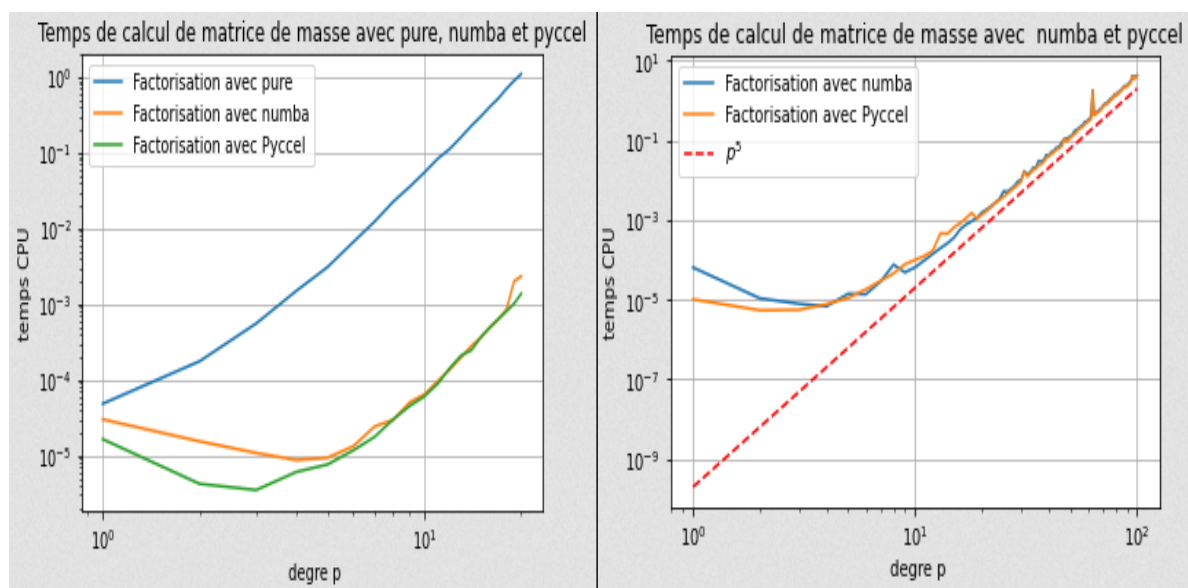


Figure 4 : Temps CPU pour la méthode Factorisation.

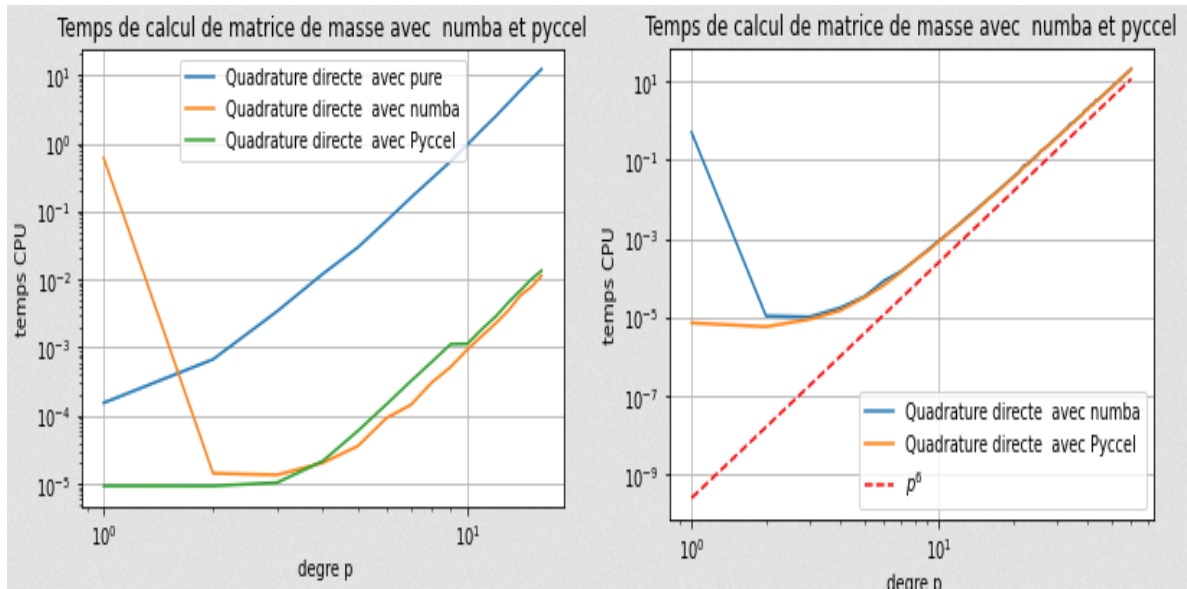


Figure 5 : Temps CPU pour la méthode Quadrature directe.

Remarques

A partir de ces figures on constate que le temps CPU calculé par la méthode analytique est $O(p^4)$ alors que le temps CPU qu'on a calculé par sum-factorisation est $O(p^5)$. On remarque aussi que Pyccel est optimal par rapport à Numba. Les figures montrent aussi que le temps CPU pour la méthode de quadrature directe est d'ordre $O(p^6)$.

0.4 Matrice de rigidité

Avec les mêmes notations de la partie 3, on définit la matrice de rigidité par:

$$K_{\alpha,\beta}^p = \int_{\hat{T}} |J_T| (J_T^{-1} \nabla B_\alpha^p) (J_T^{-1} \nabla B_\beta^p) dx, \quad \alpha, \beta \in I_p,$$

avec $J_T = (DF_T)^T$.

Lemme

Soit $T = \langle q_1, q_2, q_3 \rangle$ un triangle et λ_1, λ_2 et λ_3 les coordonnées barycentriques relatives à T . Alors

$$\nabla B_\alpha^p = d \sum_{i=1}^3 B_{\alpha^i}^{p-1} \nabla \lambda_i, \quad \alpha^i = \alpha - e_i.$$

Avec $B_{\alpha^i}^{p-1} = 0$, si α^i a une composante négative.

Démonstration

On a

$$\nabla B_\alpha^p = \binom{p}{\alpha} \nabla \lambda^\alpha,$$

donc

$$\begin{aligned}
\nabla B_\alpha^p &= \binom{p}{\alpha} (\nabla \lambda_1^{\alpha_1} \lambda_2^{\alpha_2} \lambda_3^{\alpha_3} + \lambda_1^{\alpha_1} \nabla \lambda_2^{\alpha_2} \lambda_3^{\alpha_3} + \lambda_1^{\alpha_1} \lambda_2^{\alpha_2} \nabla \lambda_3^{\alpha_3}) \\
&= \binom{p}{\alpha} (\alpha_1 \nabla \lambda_1 \lambda_1^{\alpha_1-1} \lambda_2^{\alpha_2} \lambda_3^{\alpha_3} + \alpha_2 \lambda_1^{\alpha_1} \nabla \lambda_2 \lambda_2^{\alpha_2-1} \lambda_3^{\alpha_3} + \alpha_3 \lambda_1^{\alpha_1} \lambda_2^{\alpha_2} \nabla \lambda_3 \lambda_3^{\alpha_3-1}) \\
&= p \frac{(p-1)!}{\alpha_1! \alpha_2! \alpha_3!} (\alpha_1 \lambda_1^{\alpha_1-1} \lambda_2^{\alpha_2} \lambda_3^{\alpha_3} \nabla \lambda_1 + \alpha_2 \lambda_1^{\alpha_1} \lambda_2^{\alpha_2-1} \lambda_3^{\alpha_3} \nabla \lambda_2 + \alpha_3 \lambda_1^{\alpha_1} \lambda_2^{\alpha_2} \lambda_3^{\alpha_3-1} \nabla \lambda_3) \\
&= p \sum_{i=1}^3 B_{\alpha^i}^{p-1} \nabla \lambda_i.
\end{aligned}$$

D'où le résultat \otimes

D'après ce lemme, on a :

$$\begin{aligned}
K_{\alpha,\beta}^p &= p^2 \int_{\hat{T}} |J_T| \left(J_T^{-1} \sum_{i=1}^3 B_{\alpha^i}^{p-1} \nabla \hat{\lambda}_i \right) \left(J_T^{-1} \sum_{j=1}^3 B_{\beta^j}^{p-1} \nabla \hat{\lambda}_j \right) dx \\
&= p^2 \int_{\hat{T}} \sum_{i,j=1}^3 |J_T| \left(J_T^{-1} \nabla \hat{\lambda}_i \right) \left(J_T^{-1} \nabla \hat{\lambda}_j \right) B_{\alpha^i}^{p-1} B_{\beta^j}^{p-1} dx \\
&= p^2 \sum_{i,j=1}^3 \int_{\hat{T}} |J_T| \left(J_T^{-1} \nabla \hat{\lambda}_i \right) \left(J_T^{-1} \nabla \hat{\lambda}_j \right) B_{\alpha^i}^{p-1} B_{\beta^j}^{p-1} dx.
\end{aligned}$$

Si de plus F_T est affine, alors J_T est constante et comme $\hat{\lambda}_1 = x_1$, $\hat{\lambda}_2 = x_2$ et $\hat{\lambda}_3 = 1 - x_1 - x_2$ donc $\nabla \lambda_i$ avec $i \in \{1, 2, 3\}$ est aussi constante, ce qui implique:

$$K_{\alpha,\beta}^p = p^2 \sum_{i,j=1}^3 \left(J_T^{-1} \nabla \hat{\lambda}_i \right) \left(J_T^{-1} \nabla \hat{\lambda}_j \right) |J_T| \int_{\hat{T}} B_{\alpha^i}^{p-1} B_{\beta^j}^{p-1} dx.$$

Or $|J_T| = \frac{\text{mes}(T)}{\text{mes}(\hat{T})}$, donc

$$K_{\alpha,\beta}^p = p^2 \sum_{i,j=1}^3 \left(J_T^{-1} \nabla \hat{\lambda}_i \right) \left(J_T^{-1} \nabla \hat{\lambda}_j \right) M_{\alpha_i, \beta_i}^{p-1}.$$

Mais si une arête de T est courbé, alors généralement le calcul analytique de $K_{\alpha,\beta}^p$ n'est pas faisable. Dans ce cas on va approximer $K_{\alpha,\beta}^p$ en utilisant la méthode de Gauss-Jacobi (voir partie 3).

Posons

$$g^{i,j} = |J_T| \left(J_T^{-1} \nabla \hat{\lambda}_i \right) \left(J_T^{-1} \nabla \hat{\lambda}_j \right), \quad i, j \in \{1, 2, 3\}$$

avec la transformation de Duffy on trouve

$$\begin{aligned}
K_{\alpha,\beta}^p &= p^2 \sum_{i,j=1}^3 \int_{[0,1]^2} g^{i,j}(F(x)) B_{\alpha^i}^{p-1}(F(x)) B_{\beta^j}^{p-1}(F(x)) (1-x_1) dx \\
&= p^2 \sum_{i,j=1}^3 \int_{[0,1]^2} g^{i,j}(F(x)) B_{\alpha_1^i}^{p-1}(x_1) B_{\alpha_2^i}^{p-1-\alpha_1^i}(x_2) B_{\beta_1^j}^{p-1}(x_1) B_{\beta_2^j}^{p-1-\beta_1^j}(x_2) (1-x_1) dx \\
&= p^2 \sum_{i,j=1}^3 \int_0^1 B_{\alpha_2^i}^{p-1-\alpha_1^i}(x_2) B_{\beta_2^j}^{p-1-\beta_1^j}(x_2) \left(\int_0^1 g^{i,j}(F(x)) B_{\alpha_1^i}^{p-1}(x_1) B_{\beta_1^j}^{p-1}(x_1) (1-x_1) dx_1 \right) dx_2.
\end{aligned}$$

ce qui donne finalement

$$K_{\alpha,\beta}^p \approx (0.5)^3 p^2 \sum_{i,j=1}^3 \left[\sum_{s=1}^q w_s^{(0,0)} B_{\alpha_2^i}^{p-1-\alpha_1^i}(\eta_s^{(0,0)}) B_{\beta_2^j}^{p-1-\beta_1^j}(\eta_s^{(0,0)}) \left(\sum_{r=1}^q w_r^{(1,0)} B_{\alpha_1^i}^{p-1}(\eta_r^{(1,0)}) B_{\beta_1^j}^{p-1}(\eta_r^{(1,0)}) \zeta_{r,s}^{i,j} \right) \right],$$

où $\zeta_{r,s}^{i,j} = g^{i,j}(F(\eta_r^{(1,0)}, \eta_s^{(0,0)}))$.

† Cas de F_T affine: comme dans la partie (3) On suppose que la transformation F_T est affine, de plus $F_T(x) = \sum_{i=1}^3 \hat{\lambda}_i(x) q_i$. Nous calculons la matrice de masse par les trois méthodes, les 5 figures suivantes montrent l'erreur de calcul et le temps CPU pour calculer la matrice de rigidité en fonction de degré p de polynôme de Berntien:

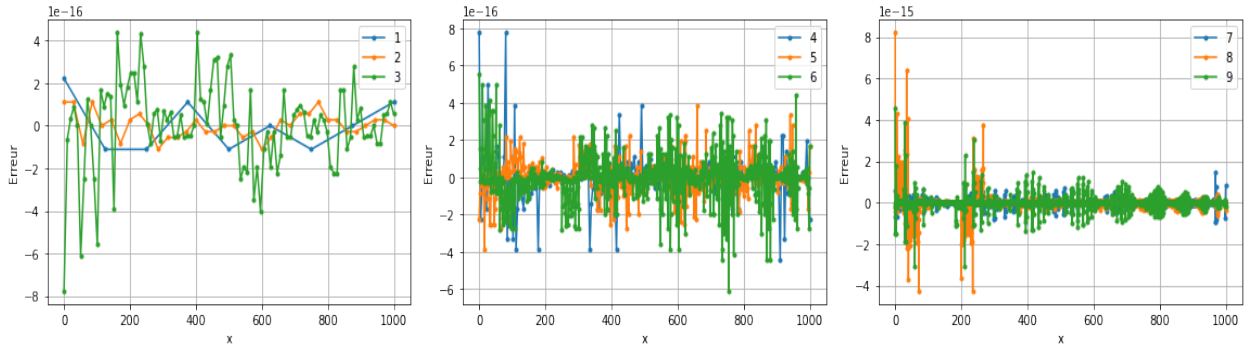


Figure 6 : Erreur entre la méthode analytique et la méthode de factorisation pour $p=1, 2, \dots, 9$.

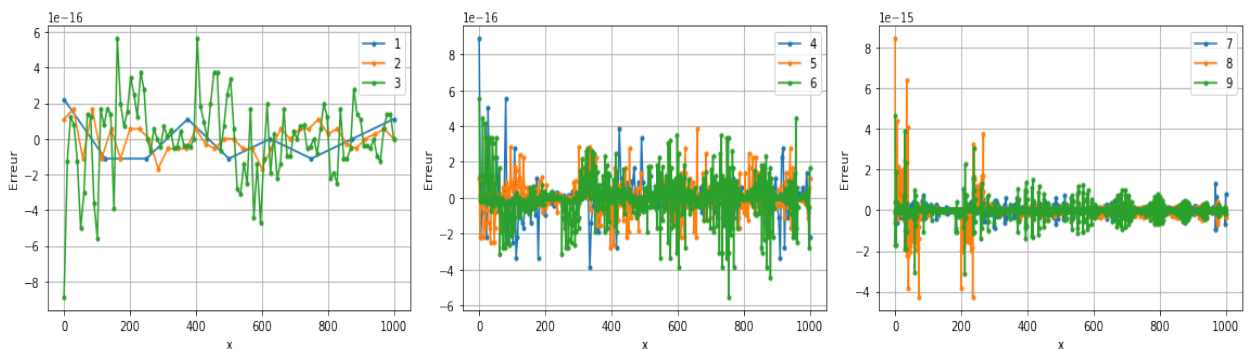


Figure 7 : Erreur entre la méthode analytique et la méthode de quadrature directe pour $p=1, 2, \dots, 9$.

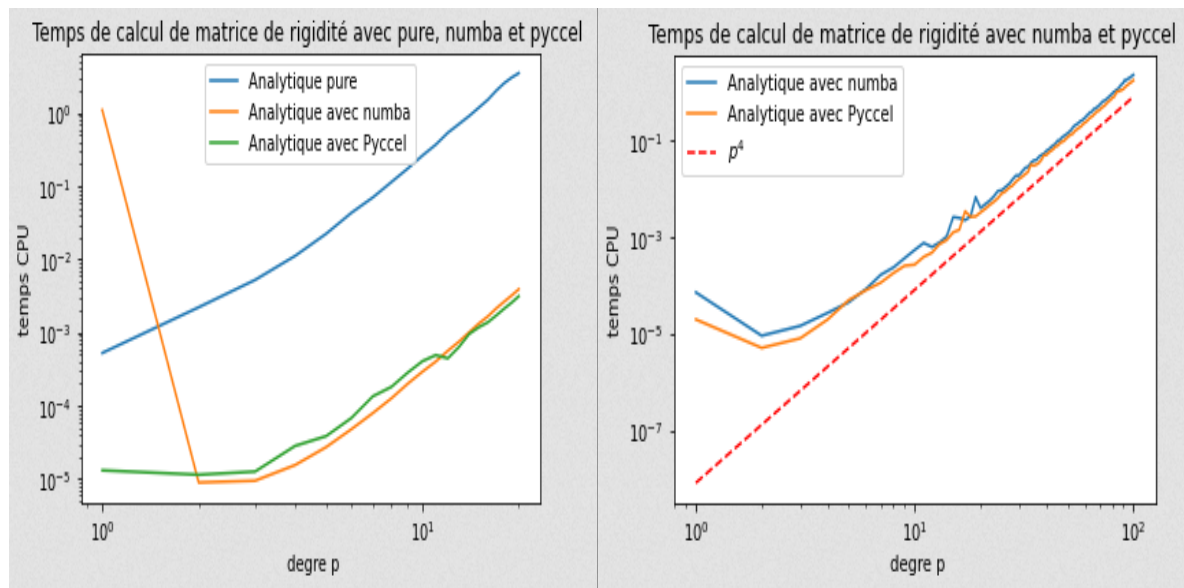


Figure 8 : Temps CPU pour la méthode analytique.

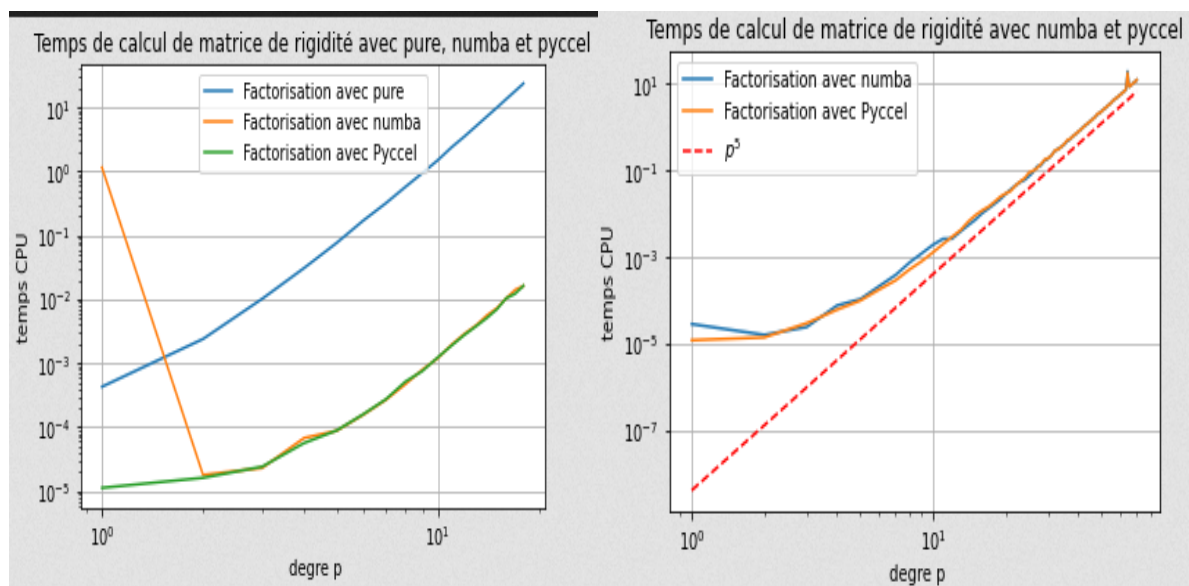


Figure 8 : Temps CPU pour la méthode de sum-factorisation.

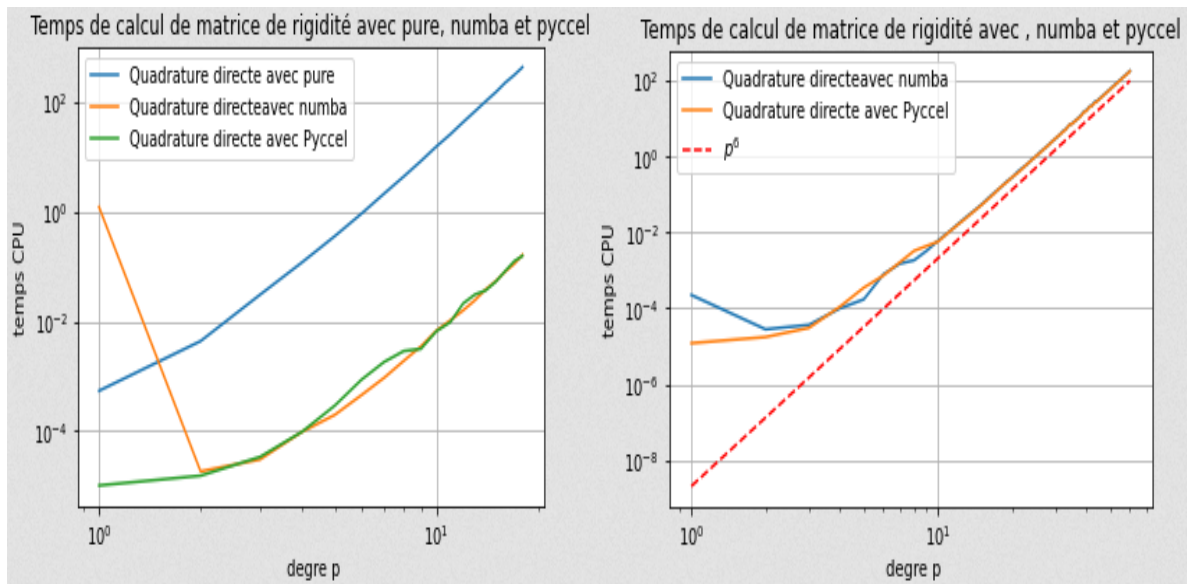


Figure 9 : Temps CPU pour la méthode de quadrature directe.

Les résultats numériques observés pour la matrice de rigidité sont similaires à ceux de la matrice de masse.

0.5 Conclusion

La méthode de factorisation par somme il permet d'assembler la matrice d'éléments avec une complexité optimal $O(p^5)$, de plus avec Pyccel en récupère le temps de calcul de Fortran, donc écrire un code Python avec la vitasse de Fortran. La propriété clé nécessaire pour utiliser la méthode de factorisation par somme est la construction tensorielle des fonctions de base de Bernstein. Cependant, les éléments qui n'ont pas de structure de produit tensoriel, par exemple les bases de Lagrange sont utilisées dans le cas d'une approximation d'ordre très bas.

Bibliography

- [1] F. IHLENBURG AND I. BABUSKA, Finite Element Solution of the Helmholtz Equation with High Wave Number Part I: The h-Version of the FEM, 1995.
- [2] F. IHLENBURG AND I. BABUSKA, Finite element solution of the helmholtz equation with high wave number part II: the h-p version of the fem, 1997.
- [3] A. El Kacimi O. Laghrouche, M.S. Mohamed AND J. Trevelyan, Bernstein–Bézier based finite elements for efficient solution of short wave problems, 2018.
- [4] MARK AINSWORTH, GAELLE ANDRIAMARO, AND OLEG DAVYDOV, BERNSTEIN–BEZIER FINITE ELEMENTS OF ARBITRARY ORDER AND OPTIMAL ASSEMBLY PROCEDURES, 2011.
- [5] MING-JUN LAI AND LARRY L. SCHUMAKER, Spline Functions on Triangulations, 2007.
- [6] Rida T. Farouki, The Bernstein polynomial basis: a centennial retrospective, 2012.