

Modélisation par éléments finis de la déformation d'une poutre sous l'action d'une force

Charles Roux-Pertus

2020

L'ensemble du code utilisé est disponible à l'adresse suivante :

https://github.com/charlesrx/simulation_elements_finis

Les notations tensorielles sont utilisées dans ce document. Par exemple le vecteur \vec{a} est noté \bar{a} car les vecteurs sont des tenseurs d'ordre 1. La matrice A est notée $\bar{\bar{A}}$ car c'est un tenseur d'ordre 2. $A:B$ représente le double produit contracté entre les tenseurs A et B .

1 Présentation du problème et formulation faible

Le but est de calculer les petites déformations élastiques d'un matériau isotrope sous l'action d'une force \bar{F} dans un espace de dimension 2.

Pour cela on utilise l'équation d'élasticité linéaire :

$$\bar{\nabla} \cdot \bar{\sigma} + \bar{F} = \bar{0} \quad (1)$$

Avec $\bar{\sigma}$ le tenseur des contraintes de Cauchy.

Pour utiliser cette équation dans le cadre des éléments finis il faut dériver la formulation faible de l'équation (1). Pour cela on multiplie l'équation par une fonction vectorielle test $\bar{v}(x, y)$ et on intègre l'équation sur le domaine de la poutre Ω qui est de longueur L et hauteur H .

$$\int_{\Omega} \bar{v} \cdot (\bar{\nabla} \cdot \bar{\sigma} + \bar{F}) d\Omega = \bar{0} \quad (2)$$

Comme le tenseur des contraintes est symétrique on peut utiliser l'égalité suivante :

$$\bar{v} \cdot \bar{\nabla} \cdot \bar{\sigma} = \bar{\nabla} \cdot (\bar{v} \cdot \bar{\sigma}) - \bar{\sigma} : \bar{\bar{\nabla}} \bar{v} \quad (3)$$

En combinant les équation (2) et (3) on obtient :

$$\int_{\Omega} \bar{\nabla} \cdot (\bar{v} \cdot \bar{\sigma}) d\Omega - \int_{\Omega} \bar{\sigma} : \bar{\bar{\nabla}} \bar{v} d\Omega = - \int_{\Omega} \bar{v} \cdot \bar{F} d\Omega \quad (4)$$

En utilisant le théorème de flux-divergence, l'égalité devient :

$$\int_{\partial\Omega} \bar{v} \cdot \bar{\sigma} \cdot \bar{n} d\partial\Omega - \int_{\Omega} \bar{\sigma} : \bar{\bar{\nabla}} \bar{v} d\Omega = - \int_{\Omega} \bar{v} \cdot \bar{F} d\Omega \quad (5)$$

En faisant l'hypothèse que les contraintes sont nulles aux bordures, le premier terme de l'équation est nul. On a finalement :

$$\int_{\Omega} \bar{\sigma} : \bar{\bar{\nabla}} \bar{v} d\Omega = \int_{\Omega} \bar{v} \cdot \bar{F} d\Omega \quad (6)$$

On utilise maintenant la loi de Hooke qui dans le cas d'un matériau isotrope s'écrit :

$$\bar{\sigma} = \lambda \text{tr}(\bar{\epsilon}) \bar{\bar{1}} + 2\mu \bar{\epsilon} \quad (7)$$

Avec $\bar{\bar{\epsilon}}$ le champ lié au vecteur déplacement \bar{u} par :

$$\bar{\bar{\epsilon}} = \frac{1}{2}(\bar{\bar{\nabla}}\bar{u} + \bar{\bar{\nabla}}\bar{u}^T) \quad (8)$$

On intègre maintenant les équations (7) et (8) dans l'équation (6), ce qui donne :

$$\int_{\Omega} \lambda(\nabla \cdot \bar{u})(\nabla \cdot \bar{v}) + \mu \bar{\bar{\nabla}}\bar{u} : \bar{\bar{\nabla}}\bar{v} + \mu \bar{\bar{\nabla}}\bar{u}^T : \bar{\bar{\nabla}}\bar{v} d\Omega = \int_{\Omega} \bar{v} \cdot \bar{F} d\Omega \quad (9)$$

Que l'on peut mettre sous la forme :

$$a(\bar{u}, \bar{v}) = l(\bar{v}) \quad (10)$$

Avec a une fonction symétrique bilinéaire et l une fonction linéaire.

2 Résolution numérique avec un élément fini

On illustre d'abord le principe de la méthode par éléments finis sur une poutre carrée de dimension 2×2 comprenant un unique élément fini.

Supposons que l'on ait un unique élément e_1 de largeur $2l=2$ et hauteur $2h=2$. Pour ce cas, on décide d'adopter le système de coordonnées illustré figure 1.

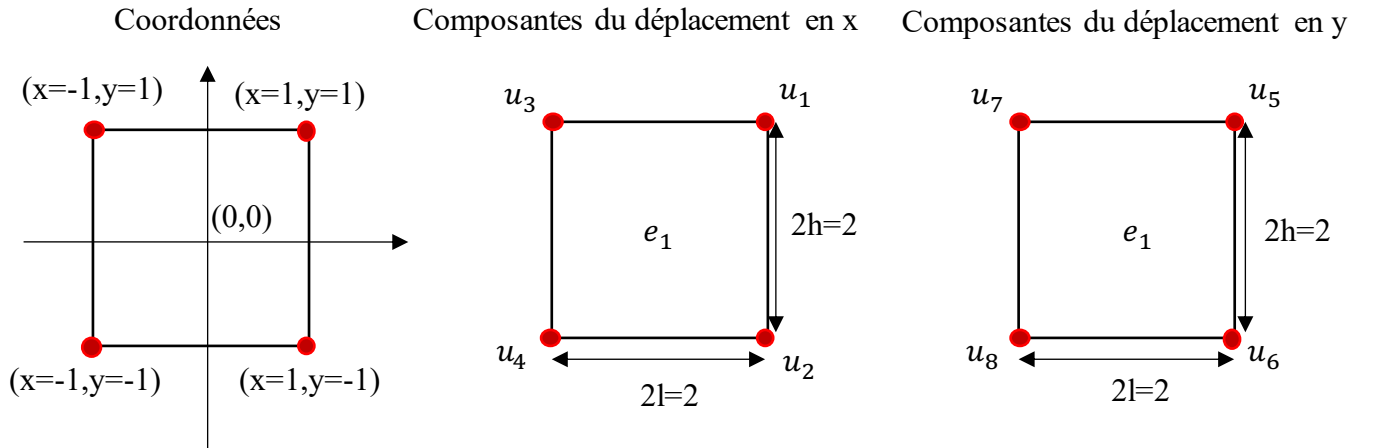


FIGURE 1 – Présentation du système de coordonnées.

On utilise la famille de fonctions nodales bilinéaire suivante pour approximer la solution aux quatres coins de l'élément e_1 :

$$\begin{aligned}
\bar{g}_1 &= \frac{(1+x)(1+y)}{4} \bar{x} \\
\bar{g}_2 &= \frac{(1+x)(1-y)}{4} \bar{x} \\
\bar{g}_3 &= \frac{(1-x)(1+y)}{4} \bar{x} \\
\bar{g}_4 &= \frac{(1-x)(1-y)}{4} \bar{x} \\
\bar{g}_5 &= \frac{(1+x)(1+y)}{4} \bar{y} \\
\bar{g}_6 &= \frac{(1+x)(1-y)}{4} \bar{y} \\
\bar{g}_7 &= \frac{(1-x)(1+y)}{4} \bar{y} \\
\bar{g}_8 &= \frac{(1-x)(1-y)}{4} \bar{y}
\end{aligned} \tag{11}$$

Remarque : il y a quatre coins et huit fonctions car les quatre premières fonctions permettent d'évaluer la composante du déplacement en x aux quatre coins de l'élément alors que les quatre autres permettent d'évaluer la composante du déplacement en y aux quatre coins de l'élément.

Les quatres premières fonctions sont illustrées en norme figure 2. Ces fonctions sont nulles en trois coins de l'élément et égales à 1 pour le dernier coin du même élément.

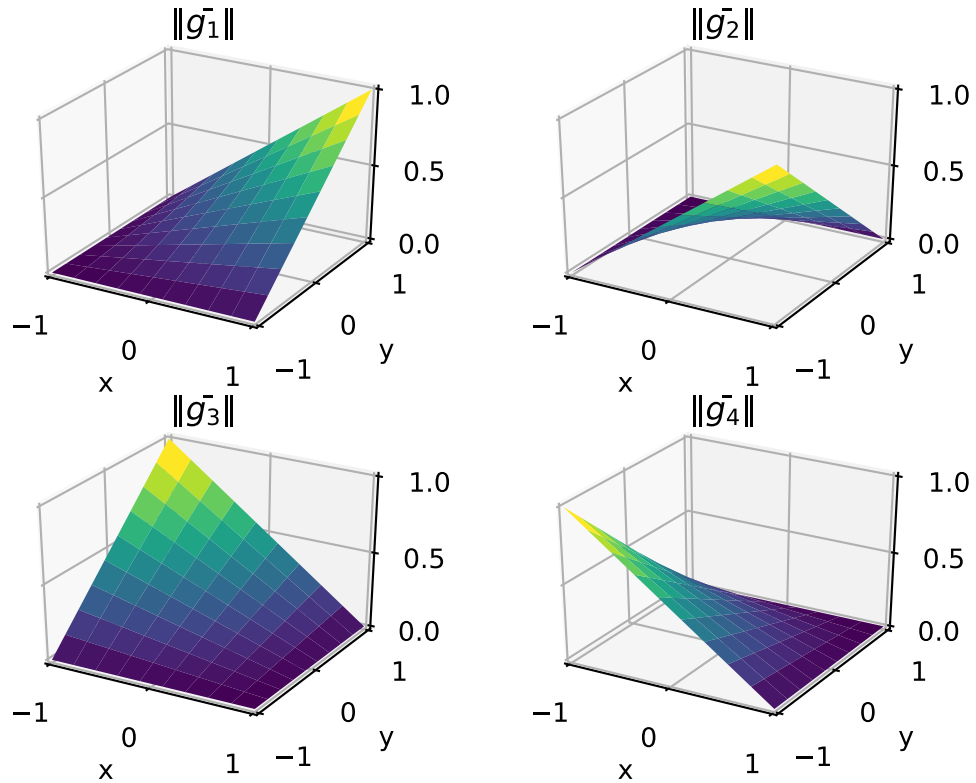


FIGURE 2 – Fonctions nodales

Pour chaque fonction de cette famille la formulation faible de l'équation est vérifiée, i.e. pour tout i appartenant à $\{1,2 \dots 8\}$ on a l'égalité suivante :

$$a(\bar{u}_s, \bar{g}_i) = l(\bar{g}_i) \quad (12)$$

Avec \bar{u}_s la fonction solution. Cette fonction peut être approchée par une combinaison linéaire de la famille de fonctions utilisées, on note alors la solution approximée \bar{u}_h avec :

$$\bar{u}_h = \sum_{j=1}^8 u_j \bar{g}_j \quad (13)$$

Remarque : les coefficients u_j sont égaux aux déplacements aux coins de l'élément. Par exemple, si on calcule \bar{u}_h au coin supérieur droit, on trouve

$$\bar{u}_h(1,1) = \sum_{j=1}^8 u_j \bar{g}_j(1,1) = u_1 \bar{g}_1(1,1) + u_5 \bar{g}_5(1,1) = u_1 \bar{x} + u_5 \bar{y} \quad (14)$$

En combinant les équations (12) et (13) il vient :

$$\forall i \in \{1, 2, \dots, 8\}, a\left(\sum_{j=1}^8 u_j \bar{g}_j, \bar{g}_i\right) = l(\bar{g}_i) \quad (15)$$

Comme a est une fonction bilinéaire :

$$\forall i \in \{1, 2, \dots, 8\}, \sum_{j=1}^8 u_j a(\bar{g}_j, \bar{g}_i) = l(\bar{g}_i) \quad (16)$$

Ce système d'équation peut être résumé par le système matriciel :

$$\bar{\bar{A}} \bar{u} = \bar{b} \quad (17)$$

Avec $\bar{\bar{A}}$, la matrice de terme général $(A_{i,j}) = a(\bar{g}_j, \bar{g}_i)$, \bar{u} le vecteur inconnu de terme général (u_i) et \bar{b} le vecteur de terme général $(b_i) = l(\bar{g}_i)$.

Avant de résoudre le système matriciel il faut calculer les termes de la matrice $\bar{\bar{A}}$ et du vecteur \bar{b} , en calculant les intégrales correspondantes. Les 64 termes de la matrice sont calculés exactement à l'aide de la fonction 'integrate' du package symbolique sympy de python.

```

1 from sympy import *
2 x,y,lam,mu=symbols('x,y,lam,mu')
3
4 g1=(1+x)*(1+y)/4
5 g2=(1+x)*(1-y)/4
6 g3=(1-x)*(1+y)/4
7 g4=(1-x)*(1-y)/4
8
9 def gg(ux,uy,vx,vy):
10     return integrate(lam*(diff(ux,x)+diff(uy,y))\
11         *(diff(vx,x)+diff(vy,y))+mu*(diff(ux,x)*\
12         diff(vx,x)+diff(ux,x)*diff(vx,x)+diff(uy,x)\
13         *diff(vx,y)+diff(ux,y)*diff(vx,y)+diff(ux,y)\
14         *diff(vy,x)+diff(uy,x)*diff(vy,x)+diff(uy,y)\
15         *diff(vy,y)+diff(uy,y)*diff(vy,y))\
16         ,(x,-1,1),(y,-1,1))
17
18 #On forme une base de 8 fonctions qui ont des

```

```

19 #coordonnées en x et y.
20 listeX=[]
21 listeX.append(g1)
22 listeX.append(g2)
23 listeX.append(g3)
24 listeX.append(g4)
25 listeX.append(Integer(0))
26 listeX.append(Integer(0))
27 listeX.append(Integer(0))
28 listeX.append(Integer(0))
29
30 listeY=[]
31 listeY.append(Integer(0))
32 listeY.append(Integer(0))
33 listeY.append(Integer(0))
34 listeY.append(Integer(0))
35 listeY.append(g1)
36 listeY.append(g2)
37 listeY.append(g3)
38 listeY.append(g4)
39
40 #A est la matrice qui représente la partie
41 #bilinéaire de la formulation faible
42 A=zeros(8,8)
43 for m in range(1,9):
44     for n in range(1,9):
45         A[m-1,n-1]=gg(listeX[m-1],listeY[m-1]\
46             ,listeX[n-1],listeY[n-1])

```

On obtient alors la matrice suivante

$$\begin{bmatrix}
 \frac{\lambda}{3} + \mu & \frac{\lambda}{6} & -\frac{\lambda}{3} - \frac{\mu}{2} & -\frac{\lambda}{6} - \frac{\mu}{2} & \frac{\lambda}{4} + \frac{\mu}{4} & -\frac{\lambda}{4} + \frac{\mu}{4} & \frac{\lambda}{4} - \frac{\mu}{4} & -\frac{\lambda}{4} - \frac{\mu}{4} \\
 \frac{\lambda}{6} & \frac{\lambda}{3} + \mu & -\frac{\lambda}{6} - \frac{\mu}{2} & -\frac{\lambda}{3} - \frac{\mu}{2} & \frac{\lambda}{4} - \frac{\mu}{4} & -\frac{\lambda}{4} - \frac{\mu}{4} & \frac{\lambda}{4} + \frac{\mu}{4} & -\frac{\lambda}{4} + \frac{\mu}{4} \\
 -\frac{\lambda}{3} - \frac{\mu}{2} & -\frac{\lambda}{6} - \frac{\mu}{2} & \frac{\lambda}{3} + \mu & \frac{\lambda}{6} & -\frac{\lambda}{4} + \frac{\mu}{4} & \frac{\lambda}{4} + \frac{\mu}{4} & -\frac{\lambda}{4} - \frac{\mu}{4} & \frac{\lambda}{4} - \frac{\mu}{4} \\
 -\frac{\lambda}{6} - \frac{\mu}{2} & -\frac{\lambda}{3} - \frac{\mu}{2} & \frac{\lambda}{6} & \frac{\lambda}{3} + \mu & -\frac{\lambda}{4} - \frac{\mu}{4} & \frac{\lambda}{4} - \frac{\mu}{4} & -\frac{\lambda}{4} + \frac{\mu}{4} & \frac{\lambda}{4} + \frac{\mu}{4} \\
 \frac{\lambda}{4} + \frac{\mu}{4} & \frac{\lambda}{4} - \frac{\mu}{4} & -\frac{\lambda}{4} + \frac{\mu}{4} & -\frac{\lambda}{4} - \frac{\mu}{4} & \frac{\lambda}{3} + \mu & -\frac{\lambda}{3} - \frac{\mu}{2} & -\frac{\lambda}{6} & -\frac{\mu}{2} \\
 -\frac{\lambda}{4} + \frac{\mu}{4} & -\frac{\lambda}{4} - \frac{\mu}{4} & \frac{\lambda}{4} + \frac{\mu}{4} & \frac{\lambda}{4} - \frac{\mu}{4} & -\frac{\lambda}{3} - \frac{\mu}{2} & \frac{\lambda}{3} + \mu & -\frac{\lambda}{6} - \frac{\mu}{2} & -\frac{\lambda}{6} \\
 \frac{\lambda}{4} - \frac{\mu}{4} & \frac{\lambda}{4} + \frac{\mu}{4} & -\frac{\lambda}{4} - \frac{\mu}{4} & -\frac{\lambda}{4} + \frac{\mu}{4} & \frac{\lambda}{6} & -\frac{\lambda}{6} - \frac{\mu}{2} & \frac{\lambda}{3} + \mu & -\frac{\lambda}{3} - \frac{\mu}{2} \\
 -\frac{\lambda}{4} - \frac{\mu}{4} & -\frac{\lambda}{4} + \frac{\mu}{4} & \frac{\lambda}{4} - \frac{\mu}{4} & \frac{\lambda}{4} + \frac{\mu}{4} & -\frac{\lambda}{6} - \frac{\mu}{2} & \frac{\lambda}{6} & -\frac{\lambda}{3} - \frac{\mu}{2} & \frac{\lambda}{3} + \mu
 \end{bmatrix} \quad (18)$$

On décide d'appliquer une force de -8.10^4 Newton selon \bar{y} en $(1,0)$. On calcule les terme du vecteur \bar{b} en posant $\bar{F} = -8e4\delta(1,0)\bar{y}$ dans l'expression de $l(\bar{v})$.

De plus, on impose un déplacement nul aux coins gauches, i.e. $u_3 = u_4 = u_7 = u_8 = 0$. On modifie donc la matrice en conséquence :

$$\begin{bmatrix}
 \frac{\lambda}{3} + \mu & \frac{\lambda}{6} & -\frac{\lambda}{3} - \frac{\mu}{2} & -\frac{\lambda}{6} - \frac{\mu}{2} & \frac{\lambda}{4} + \frac{\mu}{4} & -\frac{\lambda}{4} + \frac{\mu}{4} & \frac{\lambda}{4} - \frac{\mu}{4} & -\frac{\lambda}{4} - \frac{\mu}{4} \\
 \frac{\lambda}{6} & \frac{\lambda}{3} + \mu & -\frac{\lambda}{6} - \frac{\mu}{2} & -\frac{\lambda}{3} - \frac{\mu}{2} & \frac{\lambda}{4} - \frac{\mu}{4} & -\frac{\lambda}{4} - \frac{\mu}{4} & \frac{\lambda}{4} + \frac{\mu}{4} & -\frac{\lambda}{4} + \frac{\mu}{4} \\
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 \frac{\lambda}{4} + \frac{\mu}{4} & \frac{\lambda}{4} - \frac{\mu}{4} & -\frac{\lambda}{4} + \frac{\mu}{4} & -\frac{\lambda}{4} - \frac{\mu}{4} & \frac{\lambda}{3} + \mu & -\frac{\lambda}{3} - \frac{\mu}{2} & \frac{\lambda}{6} & -\frac{\lambda}{6} - \frac{\mu}{2} \\
 -\frac{\lambda}{4} + \frac{\mu}{4} & -\frac{\lambda}{4} - \frac{\mu}{4} & \frac{\lambda}{4} + \frac{\mu}{4} & \frac{\lambda}{4} - \frac{\mu}{4} & -\frac{\lambda}{3} - \frac{\mu}{2} & \frac{\lambda}{3} + \mu & -\frac{\lambda}{6} - \frac{\mu}{2} & \frac{\lambda}{6} \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{bmatrix} \quad (19)$$

Après résolution du système on trouve le déplacement suivant exagéré par un facteur 50000 pour la poutre.

Déformation de la poutre exagérée par un facteur :50000

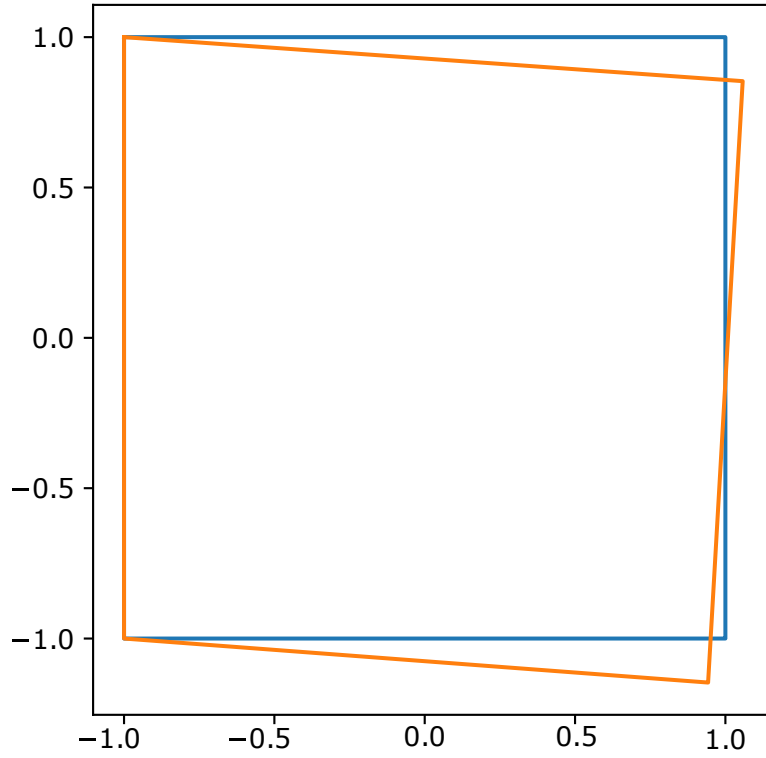


FIGURE 3 – Résolution pour un élément fini. Paramètres $E=1.18e11$ $\nu=0.31$.

Maintenant supposons que l'on ait un élément fini de dimensions arbitraires et non centré en $(0,0)$ mais en (x_0, y_0) . Comment les fonctions nodales sont-elles modifiées et que serait la matrice $\bar{\bar{A}}$?

Pour obtenir des fonctions nodales bilinéaires pour un élément centré en (x_0, y_0) et de largeur $2l$ et hauteur $2h$, on effectue le changement de variable suivant :

$$\phi(x, y) = \left(\frac{x - x_0}{l}, \frac{y - y_0}{h} \right) \quad (20)$$

On peut alors trouver la nouvelle fonction nodale \bar{g}_1^I en composant le changement de variable avec la fonction \bar{g}_1 . Cette opération est illustrée dans la figure 4.

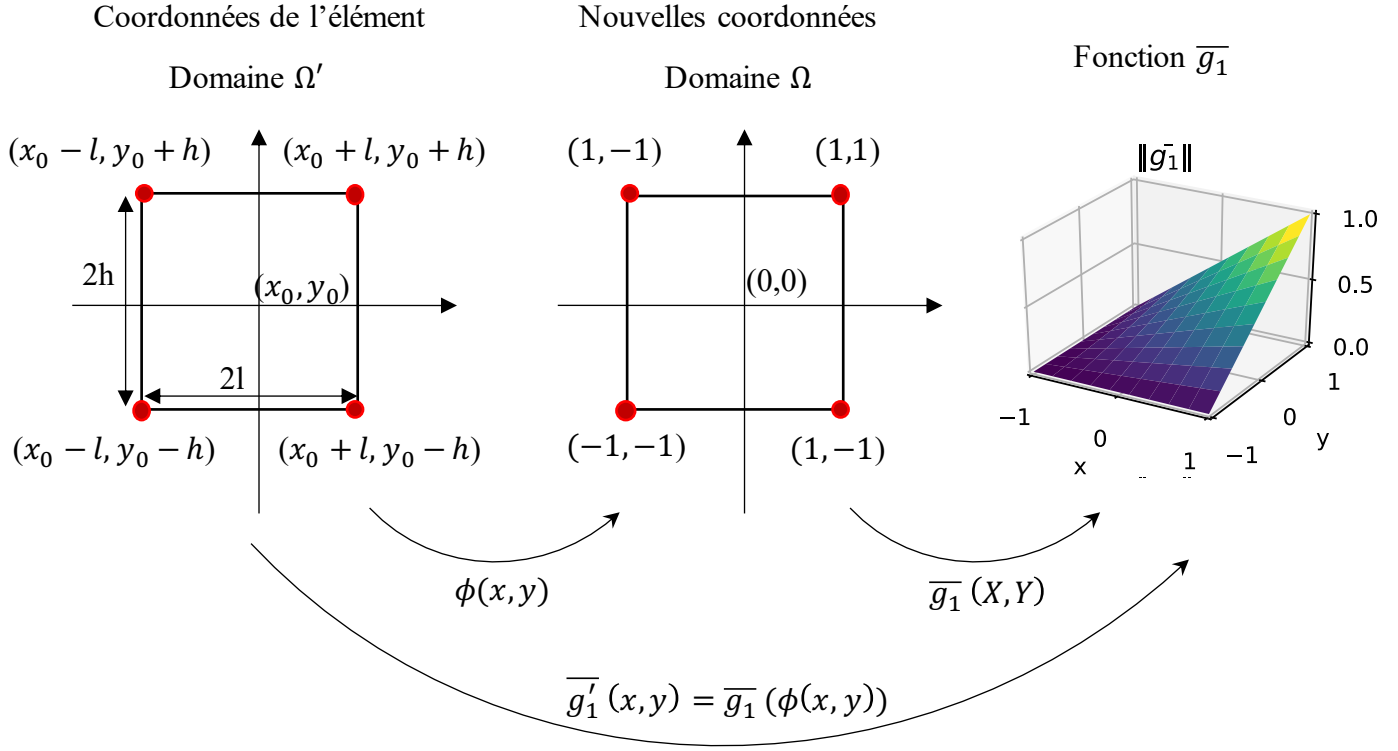


FIGURE 4 – Changement de variable pour calculer la fonction nodale \bar{g}_1

On vérifie alors bien que la fonction \bar{g}_1 est égale à 1 en norme au coin supérieur droit de l'élément et s'annule aux autres coins de l'élément. Il s'agit maintenant de calculer la nouvelle matrice $\bar{\bar{A}}'$ de terme général $(\bar{\bar{A}}'_{i,j}) = a(\bar{g}'_j, \bar{g}'_i)$. En appliquant la formule de changement de variable à une intégrale :

$$\int_{\Omega'} F(\phi(x, y)) |det J_\phi(x, y)| dx dy = \int_{\Omega} F(X, Y) dX dY \quad (21)$$

Avec F le terme général de l'intégrale de l'équation (9).

$$J_\phi(x, y) = \begin{pmatrix} 1/l & 0 \\ 0 & 1/h \end{pmatrix} \quad (22)$$

Donc

$$|det J_\phi(x, y)| = \frac{1}{l * h} \quad (23)$$

Ainsi :

$$a(\bar{g}'_j, \bar{g}'_i) = \int_{\Omega'} F(\phi(x, y)) dx dy = \frac{1}{|det J_\phi(x, y)|} * \int_{\Omega} F(X, Y) dX dY \quad (24)$$

Au final

$$a(\bar{g}'_j, \bar{g}'_i) = l * h * a(\bar{g}_j, \bar{g}_i) \quad (25)$$

Pour calculer la matrice $\bar{\bar{A}}'$ d'un élément on multiplie simplement la matrice $\bar{\bar{A}}$ par la demi-hauteur h et la demi-largeur l. C'est-à-dire $\bar{\bar{A}}' = l * h * \bar{\bar{A}}$.

3 Résolution numérique avec plusieurs éléments finis

On décide d'utiliser un maillage quadrilatéral sur la poutre. Ce maillage est illustré dans la figure 5. Le maillage est constitué de $N * N$ éléments $\{e_1, e_2, \dots, e_{N*N}\}$. La solution est calculée aux coins

de chaque éléments. Les valeurs de déplacement sont stockées dans le vecteur \bar{u} . Par exemple, sous l'action de la force le déplacement du coin supérieur gauche est u_1 , le déplacement du coin inférieur droit est $u_{(N+1)*(NN+1)}$. Pour un élément e_n , le déplacement u est calculé aux quatre coins de l'élément (voir figure 5). On peut remarquer que chaque élément a pour largeur $2l$ et pour hauteur $2h$.

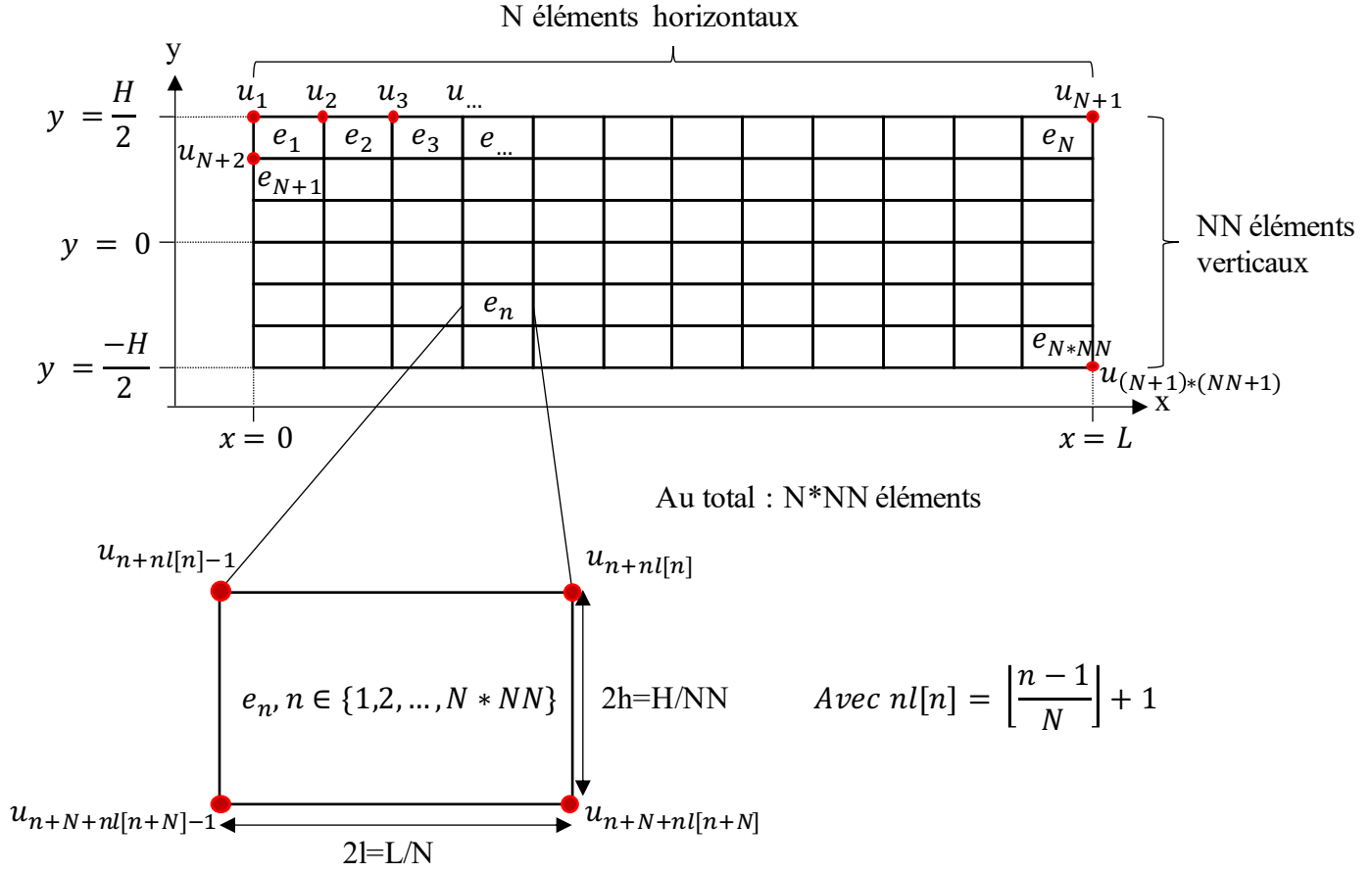


FIGURE 5 – Présentation du maillage

On décide de créer une matrice locale vers globale $\bar{l2c}$ qui permet d'indexer les coins de chaque élément. Chaque élément a huit données, quatre déplacements selon x et quatre déplacements selon y . Les déplacements sont indexés selon le schéma présenté dans la figure 6.

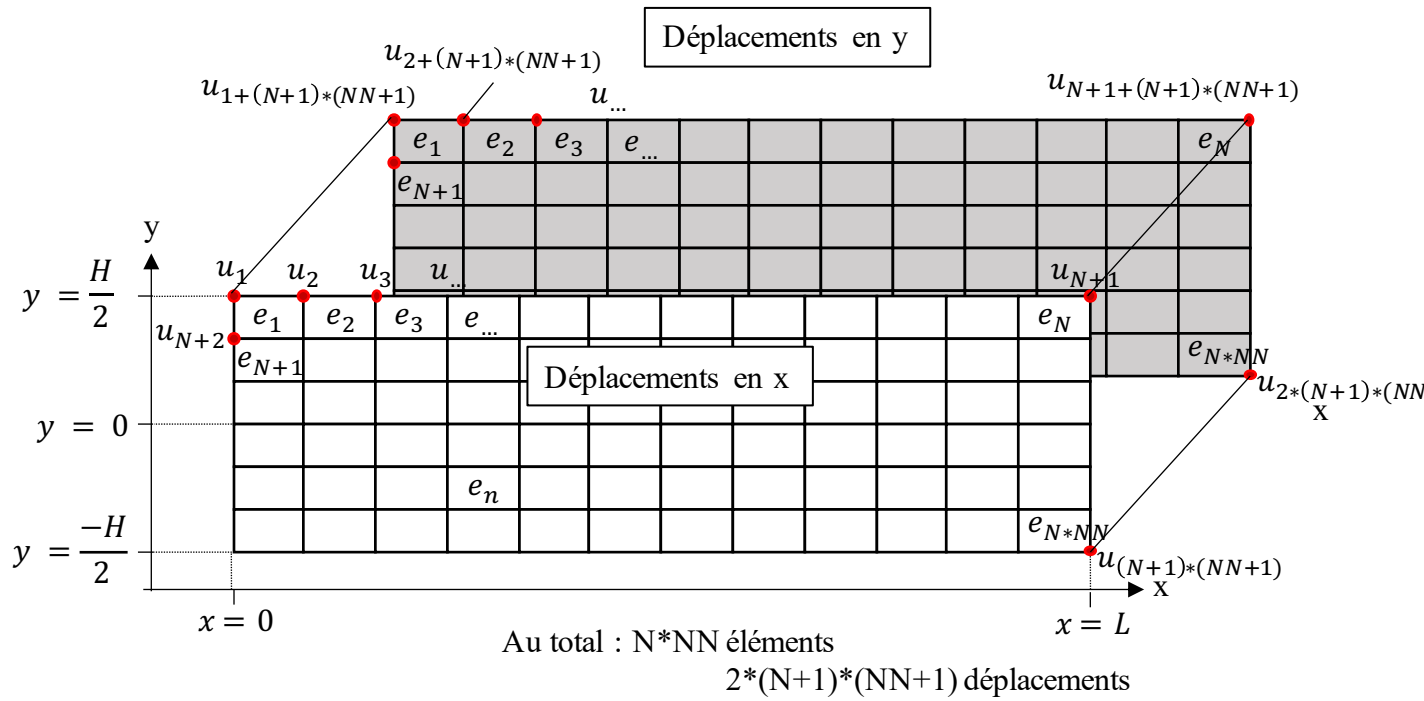


FIGURE 6 – Présentation de l'indexation

Sur le schéma figure 6, l'élément e_1 est associé à 8 valeurs de déplacement. Quatre valeurs de déplacement en x : $\{u_1, u_2, u_{N+2}, u_{N+3}\}$ et quatre valeurs de déplacements en y : $\{u_{1+(N+1)*(NN+1)}, u_{2+(N+1)*(NN+1)}, u_{N+2+(N+1)*(NN+1)}, u_{N+3+(N+1)*(NN+1)}\}$. La matrice $\bar{l}2c$ relie l'index de l'élément unique que l'on avait dans la première partie (problème local) à l'index dans le cas où l'on a plusieurs éléments finis (problème global). Ainsi la ligne n de la matrice $\bar{l}2c$ donne pour l'élément e_n les positions où sont stockées les valeurs dans le vecteur \bar{u} pour le problème global.

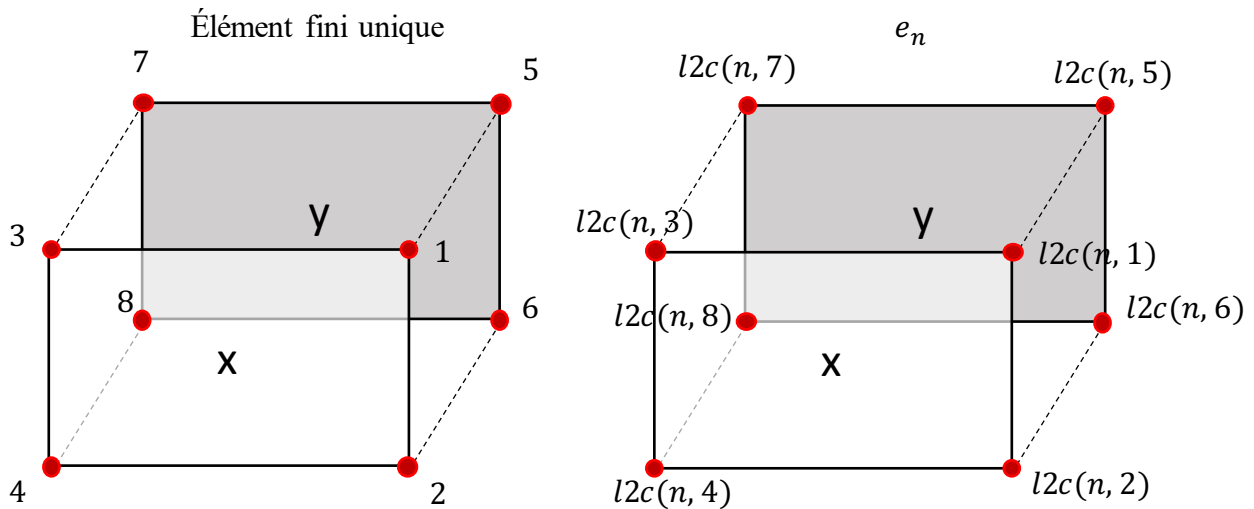


FIGURE 7 – Transformation des coordonnées de la situation avec un élément fini unique aux coordonnées du déplacement de l'élément fini e_n dans un maillage comportant plusieurs éléments finis.

Pour former cette matrice on utilise les formules suivantes :

$$\begin{aligned}
l2c(n, 1) &= n + \mathbf{nl}(n) \\
l2c(n, 2) &= n + N + \mathbf{nl}(n + N) \\
l2c(n, 3) &= n + \mathbf{nl}(n) - 1 \\
l2c(n, 4) &= n + N + \mathbf{nl}(n + N) - 1 \\
l2c(n, 5) &= n + \mathbf{nl}(n) + (N + 1) * (NN + 1) \\
l2c(n, 6) &= n + N + \mathbf{nl}(n + N) + (N + 1) * (NN + 1) \\
l2c(n, 7) &= n + \mathbf{nl}(n) - 1 + (N + 1) * (NN + 1) \\
l2c(n, 8) &= n + N + \mathbf{nl}(n + N) - 1 + (N + 1) * (NN + 1)
\end{aligned} \tag{26}$$

Avec

$$\mathbf{nl}(n) = \lfloor \frac{n-1}{N} \rfloor + 1 \tag{27}$$

Pour écrire cette matrice on utilise le code suivant :

```

1 #On initialise une matrice locale vers globale
2 l2c=np.zeros((N*NN,8))
3 #Pour cela on balaye les éléments et on associe les valeurs
4 #Pour chaque élément on balaye les coins pour x et y
5 for n in range(1,N*NN+1):
6     l2c[n-1,1-1]=n+nl(n)
7     l2c[n-1,2-1]=n+N+nl(n+N)
8     l2c[n-1,3-1]=n+nl(n)-1
9     l2c[n-1,4-1]=n+N+nl(n+N)-1
10 #on numérote aussi les coins y des éléments
11     l2c[n-1,5-1]=n+nl(n)+(N+1)*(NN+1)
12     l2c[n-1,6-1]=n+N+nl(n+N)+(N+1)*(NN+1)
13     l2c[n-1,7-1]=n+nl(n)-1+(N+1)*(NN+1)
14     l2c[n-1,8-1]=n+N+nl(n+N)-1+(N+1)*(NN+1)

```

On utilise maintenant la matrice $\bar{l2c}$ locale vers globale pour ajouter pour chaque élément les équations correspondantes dans le système globale d'équations. On remplit donc la matrice \bar{K} du problème global en balayant les éléments et en réarrangeant les termes de la matrice \bar{A} à chaque noeud. Il faut aussi prendre en compte le fait que les intégrales sont calculées sur des éléments finis de largeur et hauteur 2, après changement de variable il faut multiplier par les termes l et h . Pour cela on utilise le code suivant :

```

1 for n in range(1,N*NN+1):
2     for i in range(1,9):
3         for j in range(1,9):
4             K[int(l2c[n-1,i-1]-1),\
5               int(l2c[n-1,j-1]-1)]+=l*h*A[i-1,j-1]

```

La matrice \bar{K} , ainsi calculée, a l'allure présentée dans la figure 8 :

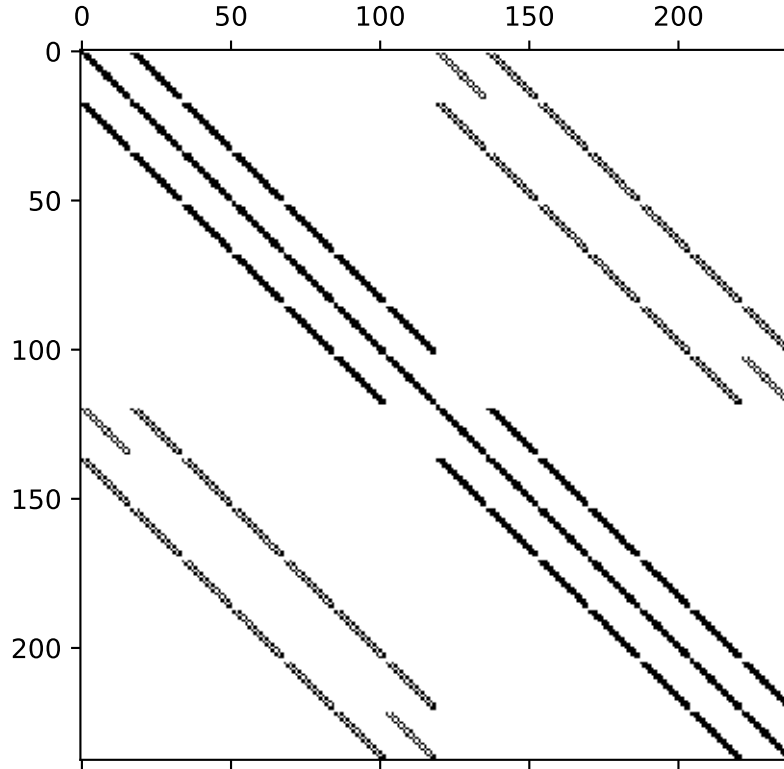


FIGURE 8 – Allure de la matrice représentant le système d'équations. Les espaces blancs correspondent à des 0 et les points noirs à des valeurs non nulles. Matrice obtenue avec $N=16$ et $NN=6$.

Au final, il ne reste plus qu'à résoudre le système $\bar{\bar{K}}\bar{u} = \bar{b}$. On choisit une force appliquée au coin supérieur droit dont l'expression est $\bar{F} = -4e4\delta(L, H)\bar{y}$.

On choisit de faire une simulation avec les paramètres suivants : une longueur horizontale L de 10 m, une hauteur verticale de 3 m. On choisit pour la simulation d'avoir $N=16$ et $NN=6$.

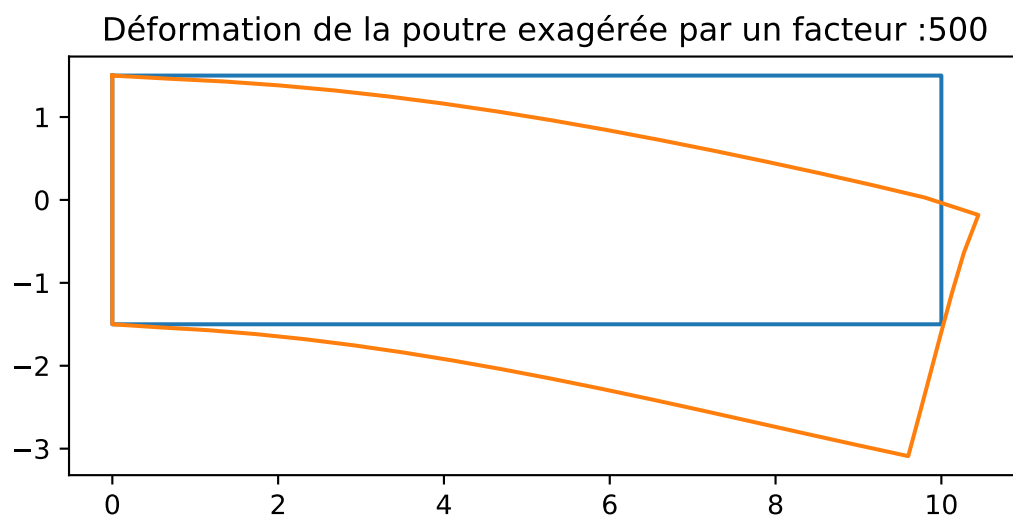


FIGURE 9 – Déformation de la poutre sous l'action d'une force.