

Rapport de travaux pratiques EDP2

Modélisation par Eléments Finis d'une poutre en traction compression

Nicolas Bonneel et Elise Durand
4ème année Génie mathématique et modélisation
Année 2004-2005

Table des matières

Introduction	2
1 Calcul de la solution exacte	3
2 Programme de résolution par éléments finis	6
2.1 Cas général	6
2.2 Application à notre poutre	7
3 Conclusion	14

Introduction

On cherche à résoudre un problème de poutre en traction-compression, dont le déplacement peut être calculé par :

$$-(EAu')'(x) + c(x).u(x) = F(x)$$

Le problème étudié est une poutre unidimensionnelle composée de deux matériaux distincts ayant des propriétés constantes, que l'on décomposera en N éléments et pour lesquels on appliquera la méthode des éléments finis. On comparera ensuite le résultat obtenu avec la solution analytique et on calculera l'erreur et le taux de convergence de la méthode employée.

1 Calcul de la solution exacte

Le problème général s'écrit :

$$-(EAu')'(x) + c(x).u(x) = F(x)$$

Ainsi, on déduit les deux équations de notre problème :

$$-a_0u'' = F_0 \text{ si } x \leq \frac{L}{2} \quad (1)$$

$$a_1u'' - c_1u = 0 \text{ si } x \geq \frac{L}{2} \quad (2)$$

- L'équation (1) nous permet d'écrire :

$$-a_0u'' = F_0 \Rightarrow u'(x) = -\frac{F_0}{a_0}.x + \alpha$$

$$\Rightarrow u(x) = -\frac{F_0}{2a_0}.x^2 + \alpha.x + \beta$$

où α et β sont des constantes.

Les conditions aux limites en $x=0$ et $x=\frac{L}{2}$ nous imposent :

$$u(0) = 0 \Rightarrow \beta = 0$$

$$u(\frac{L}{2}) = -\frac{F_0}{8a_0}.L^2 + \alpha.\frac{L}{2} \Rightarrow \alpha = (\Lambda + \frac{F_0}{8a_0}.L^2).\frac{2}{L} \text{ avec } \Lambda = u(\frac{L}{2})$$

On obtient donc :

$$u(x) = -\frac{F_0}{2a_0}.x(x - \frac{L}{2}) + \frac{2\Lambda}{L}.x \quad (3)$$

- Pour $\frac{L}{2} \leq x \leq L$ on a :

$$a_1u'' - c_1u = 0 \Rightarrow u'' - \frac{c_1}{a_1}.u = 0$$

$$\Rightarrow u(x) = A.e^{\sqrt{\frac{c_1}{a_1}}} + B.e^{-\sqrt{\frac{c_1}{a_1}}} \quad (4)$$

avec A et B constants.

– Détermination des constantes

Pour déterminer les constantes on utilise le raccord des forces élastiques en $x = \frac{L}{2}$ et la condition à la limite $x = L$.

On obtient un système de 3 équations à 3 inconnues :

$$\begin{aligned} A.e^{\sqrt{\frac{c_1}{a_1}}\frac{L}{2}} + B.e^{-\sqrt{\frac{c_1}{a_1}}\frac{L}{2}} - \Lambda &= 0 \\ A.e^{\sqrt{\frac{c_1}{a_1}}L} \cdot (k + \sqrt{\frac{c_1}{a_1}}.a_1) + B.e^{-\sqrt{\frac{c_1}{a_1}}L} (k - \sqrt{\frac{c_1}{a_1}}.a_1) &= 0 \\ A.e^{\sqrt{\frac{c_1}{a_1}}\frac{L}{2}} \cdot \sqrt{\frac{c_1}{a_1}}.a_1 - B.e^{-\sqrt{\frac{c_1}{a_1}}\frac{L}{2}} \sqrt{\frac{c_1}{a_1}}.a_1 - \Lambda \cdot \frac{2a_0}{L} &= -\frac{F_0L}{4} \end{aligned}$$

Ce système sera résolu sous Matlab :

```
function [A B Lambda] = poutre_analytique(a0,a1,c0,c1,L,k,F0);

a=(c1/a1)^0.5;
M = [ exp(a*L/2)      exp(-a*L/2)      -1      ;
      exp(a*L)*(k+a*a1)  exp(-a*L)*(k-a*a1)  0      ;
      a1*a*exp(a*L/2)   -a1*a*exp(-a*L/2)   -2*a0/L];
P = [ 0      ;
      0      ;
      -F0*L/4 ];

S = M\P;

A      = S(1);
B      = S(2);
Lambda = S(3);
```

On écrit ensuite un programme qui plotte la solution exacte de la poutre :

```
L = 5;
k = 0.05;
a0 = 2;
a1 = 1;
c0 = 0;
c1 = 0.05;
F0 = -3;
n = 501;
x=linspace(0,L, n);
```

```

[A B Lambda] = poutre_analytique(a0, a1, c0, c1, L, k, F0);
N0 = find(x<=L/2);
N1 = find(x>L/2);
v = zeros(size(x));
v(N0) = -F0*x(N0).*(x(N0)-L/2)./(2.*a0) + 2*Lambda.*x(N0)./L;
v(N1) = A*exp(a*x(N1)) + B*exp(-a*x(N1));
plot(x, v, 'r');

```

On obtient le résultat suivant :

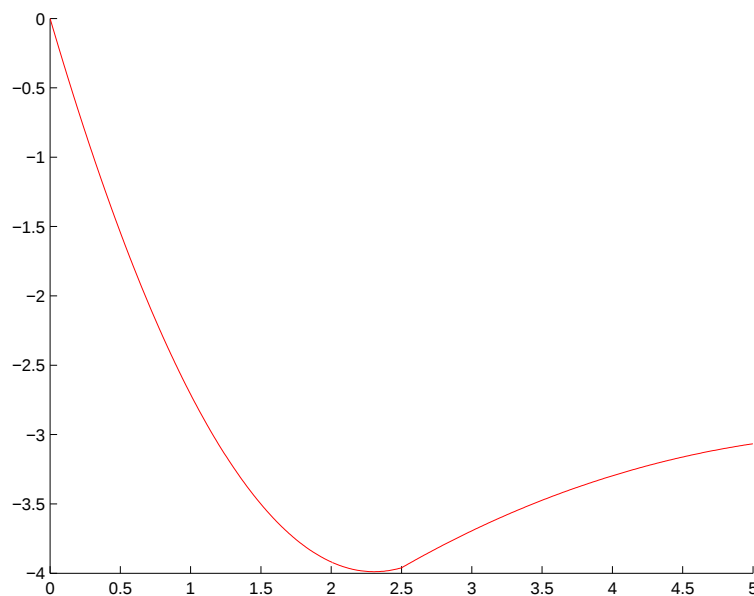


FIG. 1 – Solution exacte

On voit qu'il existe un point d'inflexion un peu avant le centre de la poutre qui est le lieu de la discontinuité. Ceci se produit parce que la force est transmise de la première poutre à la seconde poutre au lieu de les déformer à leur interface.

De même, on calcule la dérivée analytiquement qu'on multiplie par EA pour avoir sigma, les efforts internes d'élasticité :

```

s = zeros(size(x));
s(N0) = (-F0*x(N0)./a0 + (Lambda + F0*L^2/(8*a0))*(2/L))*a0;
s(N1) = (A*a*exp(a*x(N1)) - B*a*exp(-a*x(N1)))*a1;

```

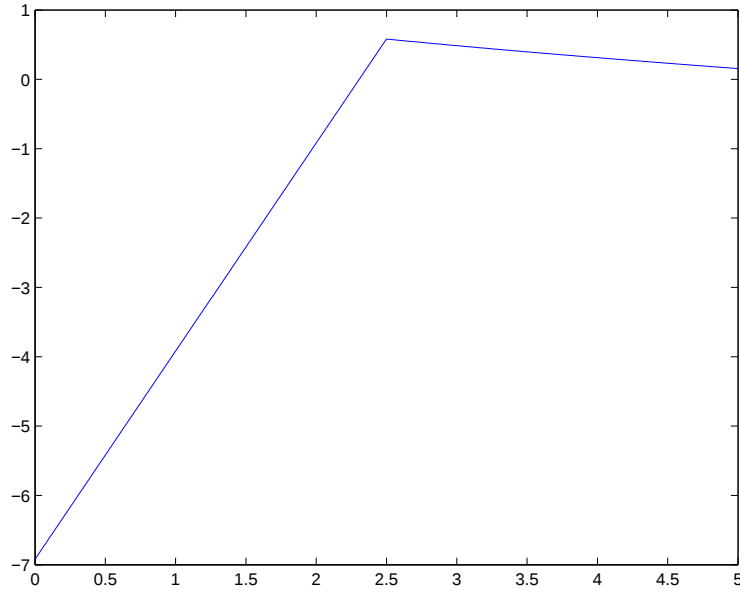


FIG. 2 – Efforts d'élasticité

2 Programme de résolution par éléments finis

2.1 Cas général

Pour obtenir le déplacement u de la poutre aux points de discrétisation, on résoud le système tridiagonal suivant :

$$\begin{pmatrix} d_1 & c_1 & & & \\ c_1 & d_2 & c_2 & & \\ & c_2 & d_3 & \ddots & \\ & & \ddots & \ddots & c_{N-1} \\ & & & c_{N-1} & d_N \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_N \end{pmatrix}$$

avec pour $e=1,\dots,(N-1)$:

$$\begin{aligned} d_e &= \left(\frac{E^{[e-1]}A^{[e-1]}}{h_{e-1}} + \frac{E^{[e]}A^{[e]}}{h_e} \right) + \left(\frac{c_{e-1/2}h_{e-1}}{4} + \frac{c_{e+1/2}h_e}{4} \right) \\ d_N &= \frac{E^{[N-1]}A^{[N-1]}}{h_{N-1}} + \frac{c_{N-1/2}h_{N-1}}{4} + k \\ c_e &= -\frac{E^{[e]}A^{[e]}}{h_e} + \frac{c_{e+1/2}h_e}{4} \\ b_e &= \left(\frac{F_{e-1/2}h_{e-1}}{2} + \frac{F_{e+1/2}h_e}{2} \right) \\ b_N &= \frac{F_{N-1/2}h_{N-1}}{2} \end{aligned}$$

Pour cela, on a créé la fonction `Matlab` suivante :

```

function u = poutre(N, h, EA, F, c, k)

de = zeros(1,N );
ce = zeros(1,N-1);
be = zeros(1,N );

de(1:N-1)=(EA(1:N-1)./h(1:N-1) + EA(2:N)./h(2:N)) + (c(1:N-1).* ...
                                                    h(1:N-1)/4 + c(2:N).*h(2:N)/4);

ce(1:N-1) = -EA(2:N)./h(2:N) + c(2:N).*h(2:N)/4;

de(N)=EA(N)./h(N) + c(N).*h(N)/4 + k;

be(1:N-1) = F(1:N-1).*h(1:N-1)/2 + F(2:N).*h(2: ...
                                                    N)/2;
be(N) = F(N)*h(N)/2;

A = sparse(N,N);

for i=1:N
    A(i,i) = de(i);
end;

for i=1:N-1
    A(i,i+1)=ce(i);
    A(i+1,i)=ce(i);
end;

u = [0; A\'(be')];

```

2.2 Application à notre poutre

On crée une fonction qui initialise la poutre en fonction du nombre d'éléments souhaités et des coefficients qui les caractérisent. On distinguera le cas où N est pair pour lequel la discontinuité au milieu de la poutre est exacte, du cas où N est impair pour lequel le point central sera la moyenne de ses voisins.

```

function [EA, c, h, F] = prepare(N,L,k,a0,a1,c0,c1,F0)

if (mod(N,2)==0)

```

```

F=[linspace(F0,F0, N/2) zeros(1,N/2)];
c=[linspace(c0,c0,N/2) linspace(c1,c1,N/2)];
h=linspace(L/N,L/N,N);
EA=[linspace(a0,a0,N/2) linspace(a1,a1,N/2)];

else
F=[ linspace(F0,F0, (N-1)/2) F0/2 zeros(1,(N-1)/2)];
c=[linspace(c0,c0,(N-1)/2) (c0+c1)/2 linspace(c1,c1,(N-1)/2)];
h=linspace(L/N,L/N,N);
EA=[linspace(a0,a0,(N-1)/2) (a0+a1)/2 linspace(a1,a1,(N-1)/2)];
end

```

Cette fonction renvoie donc les coefficients EA, c et F de chaque poutre, ainsi que leur taille h (ici, on gardera h constant).

On applique les fonctions "prepare" et "poutre" pour N=4 et les coefficients fournis. On obtient le résultat suivant :

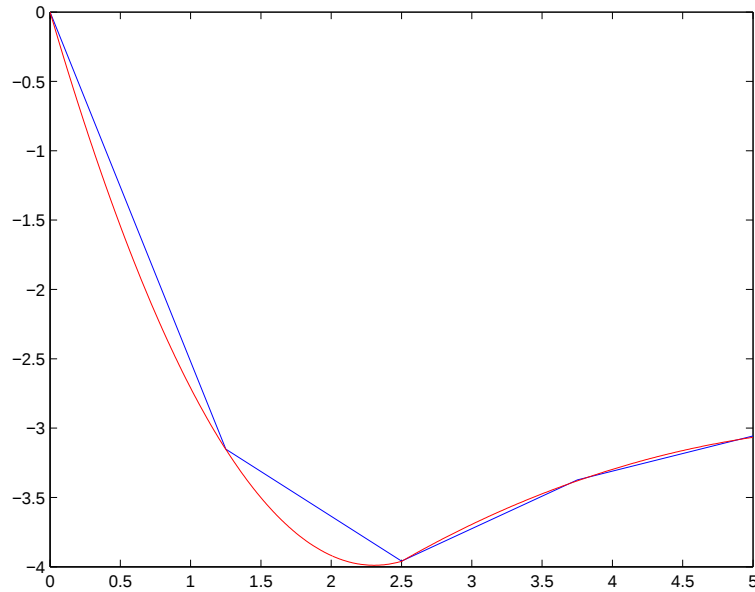


FIG. 3 – Solution approchée pour 4 poutres élémentaires et Solution exacte

On a bien la solution approchée qui est exacte aux points de discrétisation.
On fait de même avec $N = 5$:

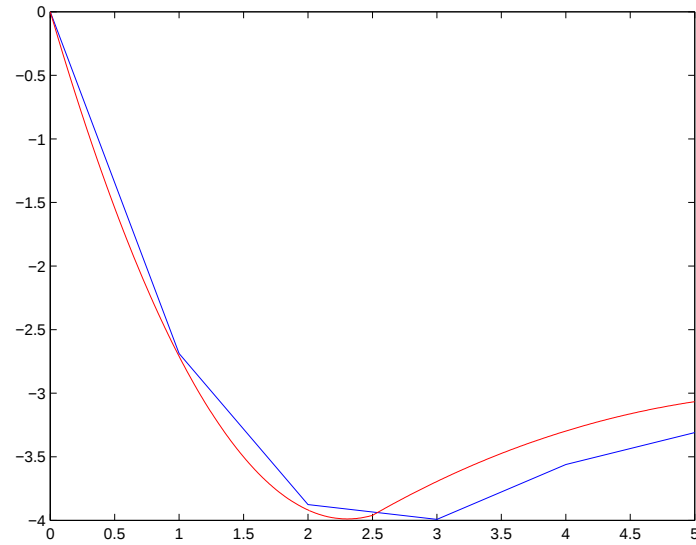


FIG. 4 – Solution approchée pour 5 poutres élémentaires et Solution exacte

On remarque que la solution approchée s'éloigne de la solution exacte car la solution exacte a été calculée pour un discontinuité parfaite en $\frac{L}{2}$ et non pour une "discontinuité approchée".

De même pour $N=10$, puis $N=11$:

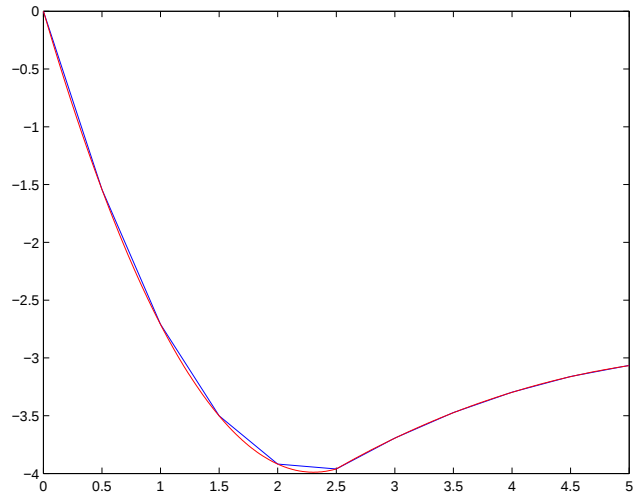


FIG. 5 – Solution approchée pour 10 poutres élémentaires et Solution exacte

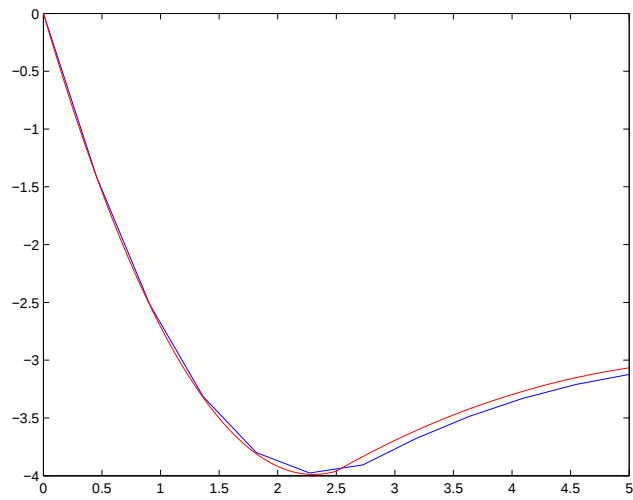


FIG. 6 – Solution approchée pour 11 poutres élémentaires et Solution exacte

On remarque que dans les deux cas, la solution est plus proche de la solution exacte, et la solution est toujours meilleure pour N pair.

On calcule ensuite l'erreur relative par la norme L^2 :

$$\int_a^b f(x)^2 dx \approx \frac{b-a}{N} \cdot \sum_{i=0}^{N-1} f^2\left(a + \left(i + \frac{1}{2}\right) * \left(\frac{b-a}{N}\right)\right)$$

avec ici $f(x) = u(x) - v(x)$ où u est la solution approchée et v la solution exacte (ici $\frac{b-a}{N} = h$ se simplifie car h est constant et que l'on calcule une erreur relative, donc un quotient des normes) :

```

numérateur = 0;
dénominateur = 0;
for i=1:N-1
    numérateur = numérateur + ((u(i)+u(i+1))/2 - v(i))^2;
    dénominateur = dénominateur+v(i)^2;
end
erreur=sqrt(numérateur/dénominateur)

```

avec u la solution obtenue par les éléments finis, et v la solution exacte calculée sur les N points au centre de chaque poutre. On itère sur les N poutres, et on prend la valeur de la solution exacte et approchée au milieu de chaque poutre. On fera attention pour les N impair à évaluer EA au centre de la poutre comme la moyenne des EA des deux éléments voisins (pour le calcul de la solution exacte sur peu de points au lieu des 500 points).

En itérant sur les N pairs de 4 à 20, on obtient la courbe log-log suivante, montrant une droite de pente 2.0715 évaluée par la commande polyval, correspondant à l'ordre du schéma. L'erreur relative calculée s'étend de 0.26% pour $N=20$ à 6.46% pour $N=4$.

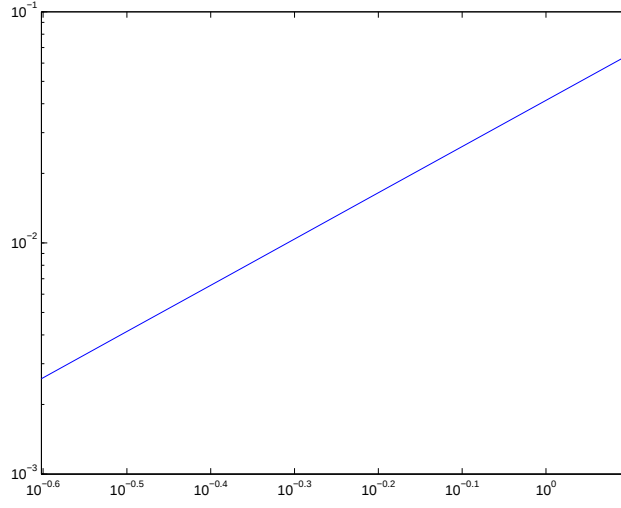


FIG. 7 – $\log\log(L/N, \text{erreur})$

Par ailleurs, pour les N impairs de 5 à 21, on obtient une pente de 1.7938, ce qui montre que le schéma est moins bon pour les N impairs (il converge plus lentement) :

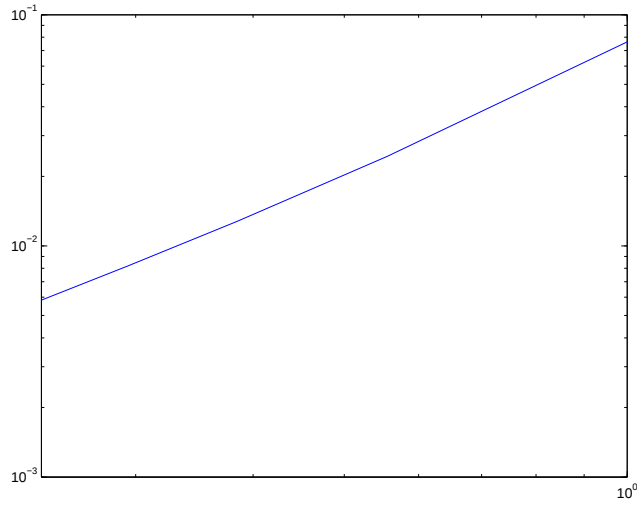


FIG. 8 – $\log\log(L/N, \text{erreur})$

On procède de même pour calculer l'erreur sur les efforts d'élasticité :

```

numérateur = 0;
dénominateur = 0;
for i=1:N-1
    numérateur = numérateur + ((EA(i)*(u(i+1)-u(i)))/h(i) - s(i))^2;
    dénominateur = dénominateur+s(i)^2;
end

erreur=sqrt(numérateur/dénominateur)

```

avec s la valeur de σ calculée analytiquement au centre de chaque poutre, et u la valeur du déplacement sur chaque poutre (donc $\frac{u_{i+1}-u_i}{h}$ donnera la dérivée au centre de la poutre à l'ordre 2 sauf pour le cas de la N ème poutre où elle donnera la dérivée décentrée à gauche à l'ordre 1).

On obtient pour le cas N pair (de 4 à 20 par pas de 4) une droite de pente

1.9738 (= ordre de convergence sur σ)

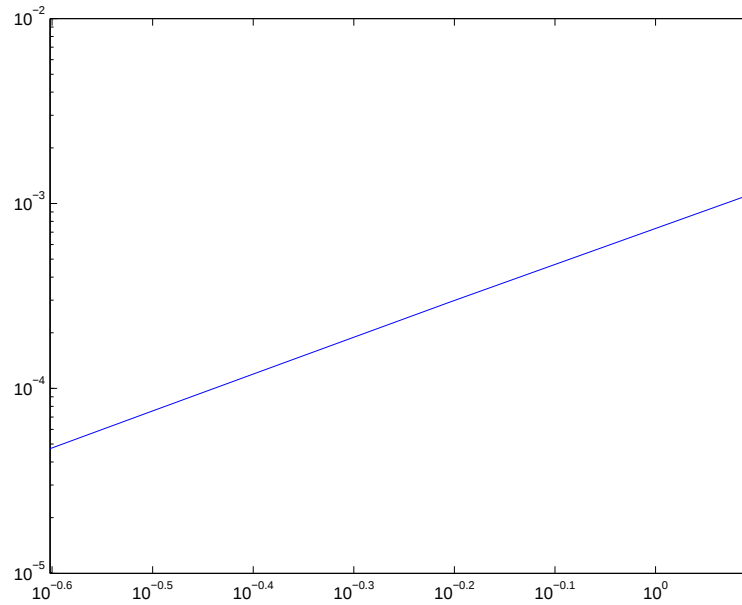


FIG. 9 – $\log\log(L/N, \text{erreur})$

et pour le cas N impair (de 5 à 21 par pas de 4) une droite de pente 1.5350 :

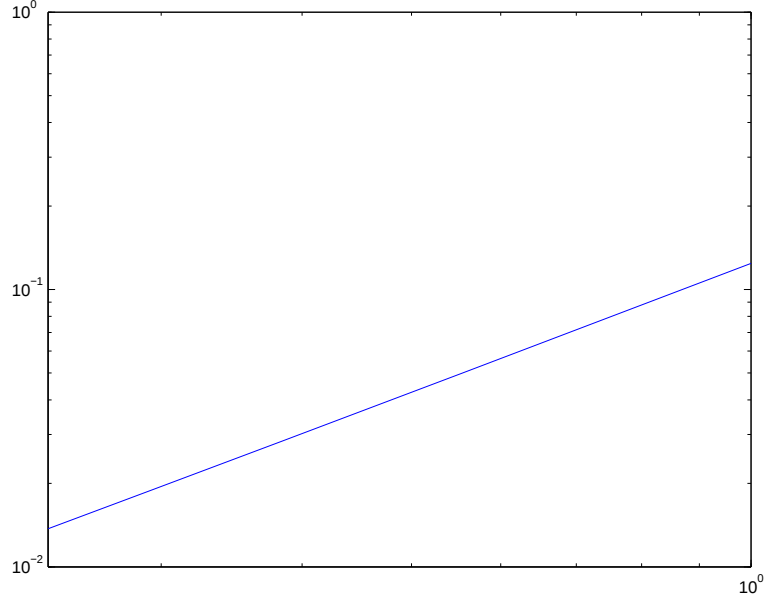


FIG. 10 – $\log\log(L/N, \text{erreur})$

3 Conclusion

On a pu observer que la méthode des éléments finis nous donnait une solution exacte aux points de discrétisation pour le cas N pair (puisque pour le cas N impair, la discontinuité au milieu de la poutre n'a pu qu'être approchée par une moyenne). De plus, on remarque que la solution en tout point converge à l'ordre 2 pour le cas N pair et 1.8 pour le cas N impair, et que les efforts d'élasticité convergent à l'ordre 2 pour le cas N pair et 1.5 pour le cas N impair (en utilisant la méthode des points milieux).

Dans le cas général, on s'évertuera à prendre un nombre pair de poutres pour éviter d'approximer la discontinuité au centre.