MINISTERE DE L'ENSEIGNEMENT
SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE DE CARTHAGE
ECOLE POLYTECHNIQUE DU TUNISIE

Compte rendu du TP

Méthode des éléments finis

Date

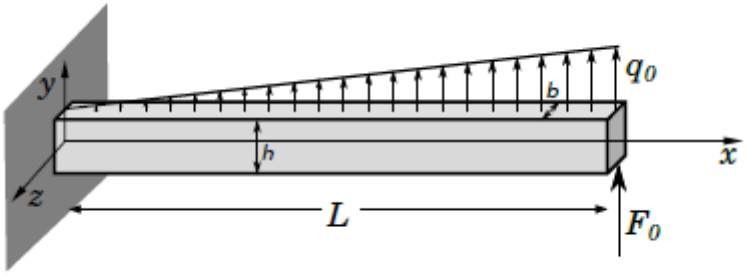
Mai 2020

Réalisé par : Maher Dissem

Année universitaire : 2019|2020

But du TP :

Le but de ce TP est d'implémenter la méthode des éléments finis pour la résolution du cas d'une poutre soumise à la flexion sous la densité de force linéique q et d'une force ponctuelle localisée à son extrémité avec le langage de programmation Matlab.



Avec las valeurs suivantes :

E = 200 GPa, I_{gz} =29 $10^{-6}m^4$, h = 150 mm, L = 1 m, F0=60 kN, q0 = 2.4 kN/m

Le code élaboré pendant les séances de ce TP permet d'obtenir la solution en déplacement ainsi que les valeurs des efforts tranchants et des moments fléchissant tout au long de la poutre pour le cas présenté plus haut et de les comparer avec les solutions exactes trouvées par un calcul de MMC.

Toutefois, j'ai essayé de rendre le code le plus général possible pour qu'il puisse traiter n'importe quel cas de flexion et les cas de poutres hétérogènes où le module d'Young E dépend de x, pour cela on doit fournir les valeurs, de q de E et du moment quadratique I le long de la poutre et modifier les vecteurs des variables primaires et secondaires pour tenir compte de nouvelles conditions imposées aux limites.

Ce compte rendu est divisé en quatre parties correspondantes au travail effectué lors des quatre séances effectuées.

Premier TP:

Après une introduction au langage de programmation Matlab, Il nous a été demandé d'Écrire des scripts qui permettent de calculer et tracer les fonctions de forme d'Hermite P3 dans le repère local pour K = [0, 1].

$$\psi_1^e(\bar{x}) = 1 - \frac{3\bar{x}^2}{h_e^2} + \frac{2\bar{x}^3}{h_e^3} \qquad \psi_1^{e\prime\prime}(\bar{x}) = -\frac{6}{h_e^2} + \frac{12}{h_e^3}\bar{x}$$

$$\psi_2^e(\bar{x}) = \bar{x} - \frac{2\bar{x}^2}{h_e} + \frac{\bar{x}^3}{h_e^2} \qquad \psi_2^{e\prime\prime}(\bar{x}) = -\frac{4}{h_e} + \frac{6}{h_e^2}\bar{x}$$

$$\psi_3^e(\bar{x}) = \frac{3\bar{x}^2}{h_e^2} + \frac{2\bar{x}^3}{h_e^3} \qquad \psi_3^{e\prime\prime}(\bar{x}) = \frac{6}{h_e^2} - \frac{12}{h_e^3}\bar{x}$$

$$\psi_4^e(\bar{x}) = -\frac{\bar{x}^2}{h_e} + \frac{\bar{x}^3}{h_e^3} \qquad \psi_4^{e\prime\prime}(\bar{x}) = -\frac{2}{h_e} + \frac{6}{h_e^2}\bar{x}$$

```
function y=fpsi(i,x,he)
  if i==1
    y=1-3*x.^2/he^2 +2*x.^3/he^3;
  elseif i==2
    y=x-2*x.^2/he+x.^3/he^2;
  elseif i==3
    y=3*x.^2/he^2-2*x.^3/he^3;
  elseif i==4
    y=-x.^2/he+x.^3/he^2;
  endif
endfunction
```

```
function y=psi2d(i,x,he)  % 2nd derivative
  if i==1
    y=-6/he^2+12/he^3*x;
  elseif i==2
    y=-4/he+6/he^2*x;
  elseif i==3
    y=6/he^2-12/he^3*x;
  elseif i==4
    y=-2/he+6*x/he^2;
  endif
endfunction
```

Deuxième TP :

Pour ce deuxième TP, on se propose d'utiliser les fonctions d'Hermite P3 à deux éléments développés précédemment pour définir :

- Une fonction kele(x1,x2,E,I) qui construit la matrice de rigidité élémentaire à partir de la valeur du module de Young $E(\overline{x})$ et du moment quadratique $I(\overline{x})$ pour un élément délimité par x1 et x2.
- Une fonction fele(x1,x2,q) qui construit le vecteur force élémentaire à partir de la valeur de la densité de force linéique q sur l'élément délimité par x1 et x2.

On démontre théoriquement qu'on a les résultats suivants :

$$K_{ij}^{e}(\overline{x}) = \int_{0}^{H_e} E(\overline{x})\ I_{gz}(\overline{x})\ \psi_i^{e\prime\prime}(\overline{x})\quad \psi_j^{e\prime\prime}(\overline{x})\quad d\overline{x}$$

$$f_{ij}^{e}(\overline{x}) = \int_{0}^{H_e} E(\overline{x})\ \psi_i^{e}(\overline{x})\quad d\overline{x} \qquad \text{avec } \overline{x} = x - x_e^1$$

```
function [k]=kele(x1,x2,E,I)
  he=x2-x1;
  k=zeros(4,4);
  x=linspace(0,he,101);
  for i=1:4
    for j=1:4
      k(i,j)=trapz( x, E.*I.*psi2d(i,x,he).*psi2d(j,x,he) );
    endfor
  endfor
endfunction
```

```
function [f]=fele(x1,x2,q)
  he=x2-x1;
  f=zeros(1,4)';
  x=linspace(0,he,101);
  for i=1:4
    f(i)=trapz( x, q.*fpsi(i,x,he) );
  endfor
endfunction
```

Où E(x), I(x) et q(x) sont des vecteurs de même taille que x.

Troisième TP :

Pour construire le système matriciel à résoudre, on se propose deux types de maillages :

- Un maillage régulier à éléments d'Hermite P3 avec n_h éléments de même longueur $h_1 = h_2 = \dots = h_{n_h} = \frac{L}{n_h}$ ou les nœuds sont donc situés aux positions $x_i = (i-1)\frac{L}{n_h}$ $i = 1, \dots, n_h + 1$
- Un maillage variable à éléments d'Hermite P3 avec n_h éléments dont les nœuds sont situés aux positions : $x_i = \frac{L}{2}\left(i - \cos\left(\frac{i-1}{n_h}\,\pi\right)\right)$

```
function x=maillage_regulier(nh,l)
  for i=1:nh+1
    x(i)=(i-1)*l/nh;
  endfor
endfunction
```

```
function x=maillage_variable(nh,l)
  for i=1:nh+1
    x(i)=l/2*(1-cos((i-1)/nh*pi));
  endfor
endfunction
```

Pour imposer les conditions d'équilibre, il est nécessaire d'additionner les 3eme et 4eme lignes de l'élément e à la 1ere et 2eme lignes de l'élément e + 1, on aura donc la forme globale de la matrice de rigidité suivante

$$[K] = \begin{bmatrix} K_{11}^1 & K_{12}^1 & K_{13}^1 & K_{14}^1 & 0 & 0 & \cdots \\ K_{12}^1 & K_{22}^1 & K_{23}^1 & K_{24}^1 & 0 & 0 & \cdots \\ K_{13}^1 & K_{23}^1 & K_{33}^1 + K_{11}^2 & K_{34}^1 + K_{12}^2 & K_{13}^2 & K_{14}^2 & \cdots \\ K_{14}^1 & K_{24}^1 & K_{34}^1 + K_{12}^2 & K_{44}^1 + K_{22}^2 & K_{23}^2 & K_{24}^2 & \cdots \\ 0 & 0 & K_{13}^2 & K_{23}^2 & K_{33}^2 + K_{11}^3 & K_{34}^2 + K_{12}^3 & \cdots \\ 0 & 0 & K_{14}^2 & K_{24}^2 & K_{34}^2 + K_{12}^3 & K_{44}^2 + K_{22}^3 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

$$\{f\} = \left\{ f_1^1, f_2^1, f_3^1 + f_1^2, f_4^1 + f_2^2, \cdots, f_4^{n_h-1} + f_2^{n_h}, f_3^{n_h}, f_4^{n_h} \right\}^T$$

On va donc assembler la matrice de rigidité et le vecteur force en utilisant les fonctions Kele et fele définis dans le TP 2 pour calculer les matrices élémentaires de rigidité et les vecteurs force élémentaires et les placer directement à leurs emplacements adéquats dans la matrice globale de rigidité [K] et le vecteur force global [f].
On doit donc à chaque fois pour créer la matrice élémentaire, extraire la partie de E(x) et de I(x) qui correspond à l'élément en question. De même pour fele et q(x).

```
function [k,f]=assemblage(nh,l,E,I,q,m)
  if m==0
    x=maillage_regulier(nh,l);
  elseif m==1
    x=maillage_variable(nh,l);
  endif
  k=zeros(2*nh+2);
  m=0;n=1;
  p=0;
  while p<2*nh
    m=m+1; n=n+1;
    x1=x(m);x2=x(n);
    z=101;          % on prend la partie de E et I qui appartient a l'element
    Eele=linspace(E(m),E(n),z);
    Iele=linspace(I(m),I(n),z);
    for i=1:4
      for j=1:4
        k(p+i,p+j)=k(p+i,p+j)+kele(x1,x2,Eele,Iele)(i,j);
      endfor
    endfor
    p=p+2;
  endwhile
  % de meme pour f
  f=zeros(2*nh+2,1);
  m=0;n=1;
  p=0;
  while p<2*nh
    m=m+1; n=n+1;
    x1=x(m);x2=x(n);
    z=101;           % on prend la partie de q qui appartient a l'element
    qele=linspace(q(m),q(n),z);    % z represente le nombre d'element du vecte
ur q qui doit etre le meme que celui de x dans fele()
    for i=1:4
      f(p+i)=f(p+i)+fele(x1,x2,qele)(i);
    endfor
    p=p+2;
  endwhile
endfunction
```

Application des conditions aux limites

L'encastrement effectué en x=0 impose qu'on ait en ce point un déplacement nul et une rotation nulle. Alors que les valeurs des efforts tranchants et du moment fléchissant en ce points sont non nuls et nous sont inconnues.

Puis pour les nœuds suivants on peut appliquer la condition d'équilibre pour obtenir :
$Q_3^e + Q_1^{e+1} = 0$ (Efforts extérieurs suivant y au noeud e $+$ 1 $nuls$)
$Q_4^e + Q_2^{e+1} = 0$ (Moments extérieurs autour de z au noeud e $+$ 1 $nuls$)
Au nœud $2n_h$ +1, une force ponctuelle F_0 est appliquée donc $Q_3^{n_h+1} = F_0$ et le moment en ce point est nul $Q_4^{n_h+1} = 0$

Remarque :

Lors de la définition des variables primaires et secondaires aux nœuds, la valeur NaN sera utilisée dans le code pour désigner les valeurs inconnues qui sont à déterminer. On pourra ainsi en tenir compte lors de la résolution et ne calculer la déformée qu'aux points ou elle est inconnue.

Cette démarche permet donc de traiter d'autres problèmes avec d'autres conditions où les valeurs de {u} peuvent être connues autre part qu'en x=0. Il suffit pour cela de modifier les fonctions suivantes.

```
function U=dep(nh,l)
  U=zeros(2*nh+2,1);
  U(1)=0;     % encastrement en x=0
  U(2)=0;
  for i=3:2*nh+2
    U(i)=NaN; %  la valeur NaN servira a reconnaitre les valeurs inconnues lor
s de la resolution
  endfor
endfunction
```

```
function Q=efforts(nh,l,F0)            % cond aux lims de la var secondaire
  Q=zeros(2*nh+2,1);
  Q(1)=NaN;
  Q(2)=NaN;
  for i=3:2*nh
    Q(i)=0;
  endfor
  Q(2*nh+1,1)=F0;
  Q(2*nh+2,1)=0;
endfunction
```

Quatrième TP :

On se propose maintenant de résoudre le système matriciel $[K]\{u\} = \{Q\} + \{f\}$

On dispose de la matrice de rigidité [K], du vecteur force linéique {f} calculés dans le TP précédent. Cependant, en imposant des conditions aux limites ou entre les éléments, on peut connaitre une partie des composantes des vecteurs {u} et {Q}. On doit donc écrire une fonction qui enlève les composantes connues de {u} et qui extrait la sous-matrice {K} et les sous-vecteurs {f} et {Q} relatives aux valeurs inconnues

```
function [vect,mat,eff]=inconnu(U,K,F)% retourne le vecteur compose uniquement
  n=length(U);                        % des valeurs inconnues et la matrice k
  vect=U; mat=K; eff=F;               % correspondante
  while n>0
    if isnan(vect(n))==0   % on enleve les valeurs != nan qui sont connues
      vect(n)=[];
      eff(n)=[];
      mat(:,n)=[];
      mat(n,:)=[];
    endif
    n=n-1;
  endwhile
endfunction
```

Ensuite, et en utilisant cette fonction, on écrit une fonction qui résout le système matriciel $[K]\{u\} = \{F\}$

```
function U=resolutionU(U,K,F)
  [U0,K0,F0]=inconnu(U,K,F);
  U0=K0\F0;                 % on calcule les valeurs inconnues de U
  j=1;
  for i=1:length(U)         % on met les nouvelles valeurs calculees de U
    if isnan(U(i))==1       % dans les composantes NaN
      U(i)=U0(j);
      j=j+1;
    endif
  endfor
endfunction
```

On peut maintenant résoudre le système pour calculer la déformée et comparer cette solution approchée avec la solution analytique obtenue par un calcul de MMC et qui est :

$$v(x) = \frac{F_0\, x^2}{6\, E\, I}\,(3L - x) + \frac{q_0\, L^4}{120\, E\, I}\,(20\frac{x^2}{L^2} - 10\frac{x^3}{L^3} + 5\frac{x^5}{L^5})$$

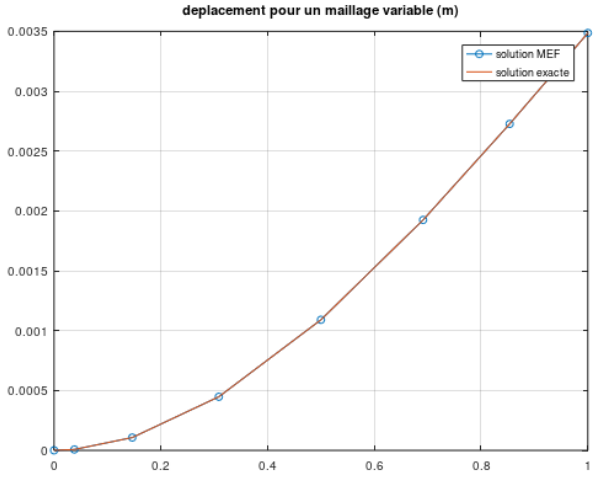
```
function v=vAnalytique(x,L,E,I,F0,q0)
  v=(F0/(6*E*I)*x.^2).*(3*L-x)+q0*L^4/(120*E*I)*(20*x.^2/L^2-
10*x.^3/L^3+x.^5/L^5);
endfunction
```

Pour cela on définit toutes les valeurs en unité du système international et leurs dérivés (m, N, Pa).
E(x), I(x) et q(x) doivent être entrés en tant que vecteurs de même taille que le maillage choisit. Etant constants dans notre cas, on définit leurs valeurs par E0, I0 et q0.

```
  E0=200*10^9; I0=29*10^-6; h=0.15; l=1; F0=60*10^3; q0=2.4*10^3;
  nh=8

  % maillage variable
  x=maillage_variable(nh,l);
  q=x*q0/l;
  for i=1:length(x)
    E(i)=E0;
    I(i)=I0;
  endfor
```

Il faut noter que dans le vecteur {u} il y a les composantes de translation et de rotation.

```
[k,f]=assemblage(nh,l,E,I,q,1);
  u=dep(nh,l);
  Q=efforts(nh,l,F0);
  u=resolutionU(u,k,Q+f);
  % pour dessiner le graphique on prend uniquement les valeurs de u relatives
a des translations
  j=1;
  for i=1:2*nh+2
    if mod(i,2)==1
      Ut(j)=u(i);
      j=j+1;
    endif
  endfor
  v=vAnalytique(x,l,E0,I0,F0,q0);
  subplot(1,4,1);
  plot(x,Ut,'o',x,v); legend("solution MEF","solution exacte"); grid(); title(
"deplacement pour un maillage variable (m)");
```



On va maintenant calculer les variables secondaires aux nœuds. Pour cela on va utiliser la forme locale de l'équilibre sur un élément :
$\{Q^e\} = [K^e]\{u^e\} - \{f^e\}$ Où {U} est le déplacement nodal adéquat.

```
function Q=Qele(e,u,x,E,I,q)          % retourne le vecteur Q de l'element e.
  x1=x(e);                            % on doit recalculer K et F de l'element e
  x2=x(e+1);                          % et utiliser la partie du deplacement u
  he=x2-x1;                           % relative a l'element en question.
  z=101;
  Eele=linspace(E(e),E(e+1),z); % on prend la partie de E, I et q qui appartie
nt a l'element
  Iele=linspace(I(e),I(e+1),z);
  qele=linspace(q(e),q(e+1),z);
  k=kele(x1,x2,Eele,Iele);
  f=fele(x1,x2,qele);
  uele=u(2*(e-1)+1:2*(e-1)+1+3);
  Q=k*uele-f;
endfunction
```

On peut donc utiliser cette fonction pour déterminer les valeurs des efforts et des moments au point x=0 ou ils nous sont inconnues.

```
  Qele1=Qele(1,u,x,E,I,q)
  printf("Ty(0)=%f N | Mfz(0)=%f Nm | Ty(he)=%f N | Mfz(he)=%f Nm \n",-
Qele1(1),-Qele1(2),Qele1(3),Qele1(4));
```

```
>>   printf("Ty(0)=%f N | Mfz(0)=%f Nm | Ty(he)=%f N | Mfz(he)=%f Nm \n",-Qele1(1),-Qele1(2),Qele1

Ty(0)=61200.000000 N | Mfz(0)=60800.001008 Nm | Ty(he)=61198.261702 N | Mfz(he)=58470.736754 Nm
>>

>>
```

Avec cette fonction on peut maintenant déterminer $T_y(x)$ et $M_{fz}(x)$ le long de la poutre et les comparer aux valeurs analytiques.

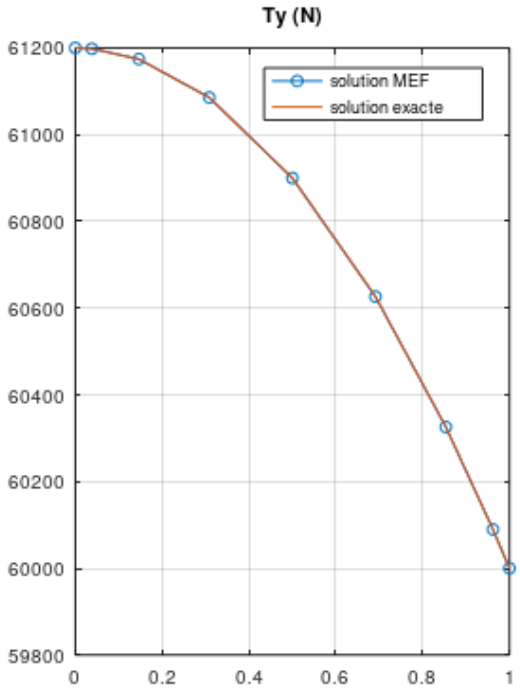
$$T_y(x) = F_0 + \frac{q_0}{2} L \left(1 - \frac{x^2}{L^2}\right)$$

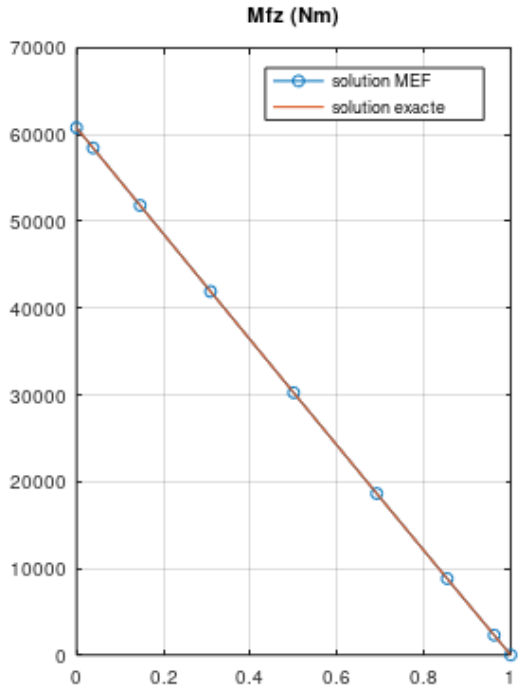
$$M_{fz}(x) = F_0(L - x) + \frac{q_0}{2} L^2 \left(2 - 3\frac{x}{L} - \frac{x^3}{L^3}\right)$$

Pour notre cas et par conséquence à la condition d'équilibre, on obtient $Q_3^e = -Q_1^{e+1}$ et $Q_4^e = -Q_2^{e+1}$ donc pour tracer les courbes on doit prendre une seule valeur par élément.

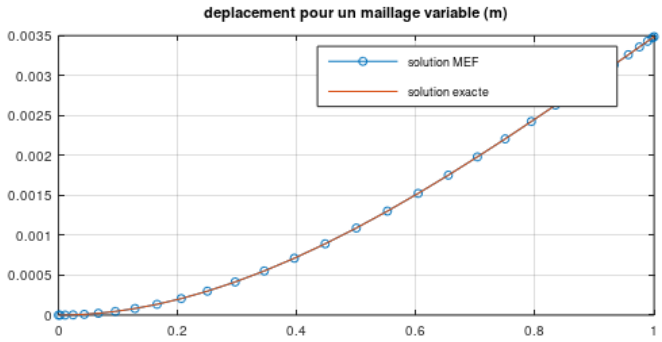
```
  % valeurs exactes de Ty Mfz
  Ty=F0+q0*l/2*(1-x.^2/l^2)
  Mfz=F0*(l-x)+q0*l^2/6*(2-3*x/l+x.^3/l^3)
```

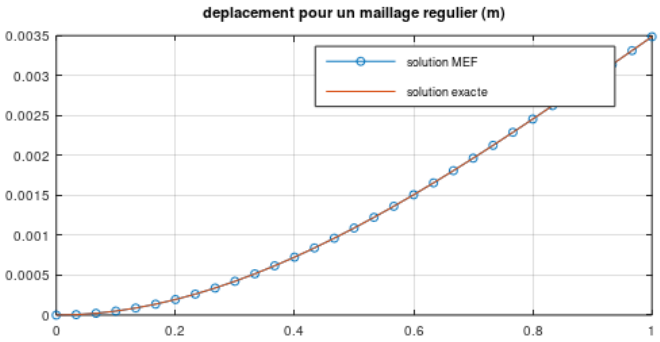
```
  % valeurs par la MEF
  for e=1:nh                              % il y a des valeurs qui se repetent
    Ty_mef(e)=-Qele(e,u,x,E,I,q)(1);      % pour des elements consecutifs
  endfor                                  % => on prend une valeur par element
  Ty_mef(nh+1)=Qele(nh,u,x,E,I,q)(3);
 for e=1:nh
    Mfz_mef(e)=-Qele(e,u,x,E,I,q)(2);
  endfor
  Mfz_mef(nh+1)=Qele(nh,u,x,E,I,q)(4);
  Mfz_mef

  subplot(1,2,1);
  plot(x,Ty_mef,'o',x,Ty); legend("solution MEF","solution exacte"); grid();
title("Ty (N)");
  subplot(1,2,2);
  plot(x,Mfz_mef,'o',x,Mfz); legend("solution MEF","solution exacte"); grid();
title("Mfz (Nm)");
```

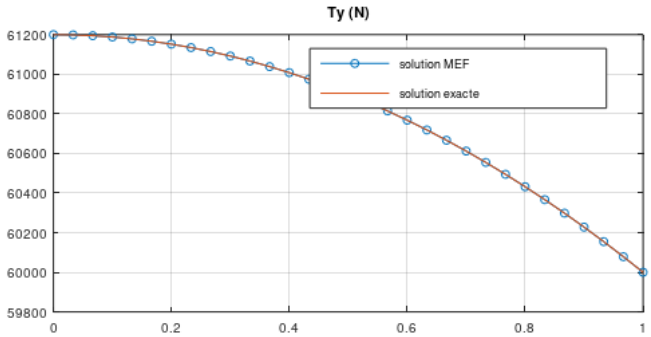


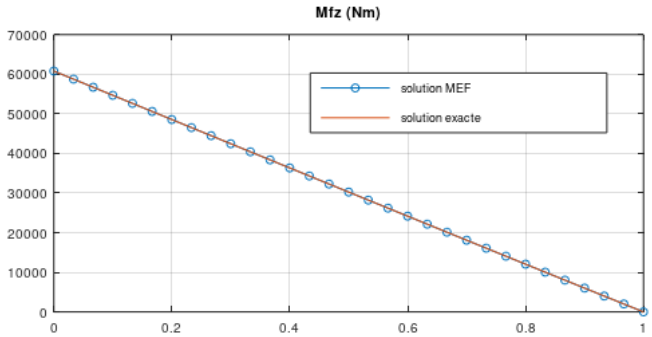


Récapitulatif des résultats et conclusions :









- On obtient une solution précise même avec un faible nombre d'éléments. La valeur de z (le nombre de division de l'intervalle lors de l'intégration) a également un rôle important pour l'obtention valeurs précises.
- La valeur maximale du déplacement pour notre cas est de 0.0035m. Si cette valeur ne convient pas pour l'utilisation de la poutre, on peut essayer de modifier le module d'Young (changer de matériaux) et refaire la simulation.
- On obtient une valeur maximale du moment fléchissant dans l'encastrement en x=0 et qui a pour valeur 60 KNm. On peut maintenant calculer $|\sigma_{max}| = \frac{|M_{fz}|}{\frac{I_{gz}}{|z_{max}|}}$ et le comparer avec la contrainte pratique admissible $R_{pe} = \frac{R_e}{s}$ pour savoir si la poutre va résister à la flexion dans le cas d'application souhaité.