TP+1+-+Elements+Finis+-+Mohamed+Maamir-Xunjie+ZHANG v2016.10.21

October 22, 2016

1 TP 1 - Méthode des élémenst finis en 1D

In [58]:
```
%matplotlib inline
%autosave 60
import numpy as np
import scipy as sp
#import scipy.optimize as opt
import matplotlib.pyplot as plt
```

```
Autosaving every 60 seconds
```

1.1 Introduction

Dans ce *notebook*, nous allons étudier par la méthode des éléments fini en *1D* les **éfforts de traction** sur une poutre en rotation. Pour se faire, nous allons partir de **l'équation d'équilibre** et des conditions aux limites et donner **la formulation faible** du problème. On étudira une approximation de la solution dans le cas ou la solution est un **polynome d'odre 1** et dans le cas où il est **d'odre 2**.

1.2 Problème Physique

1.2.1 Equation d'équilibre

Un petit élément Δr est soumis à : 1. Force centrifuge : $(\rho * S(r) * \omega^2 * r) * \Delta r$ 2. Efforts internes : $ES(r + \Delta r)\frac{du}{dr}|_{r+\Delta r} - ES(r)\frac{du}{dr}|_r = \frac{d}{dr} * [ES(r)\frac{du}{dr}]$

Soit l'équation d'équilibre suivante :

$$\frac{d}{dr}(ES(r)\frac{du}{dr}) + (\rho S(r)\omega^2)r = 0$$

1.2.2 Formulation faible

On retrouve cette formulation faible :

Trouver $u(r)$ **tel que** $u(0) = 0$ **et** $ES(r)\frac{du}{dr}\Big|_{r=L} = 0$ **:**

$$\int_0^L ES(r)\frac{du}{dr}\,dr = \int_0^L (\rho S(r)\omega^2 r)v(r)\,dr$$

1.2.3 solution Analytique

Cas Section Constante On retrouve pour une section constrante une solution :

$$u(r)_{Sconst} = \frac{\rho\omega^2}{2 * E}(L^2 r - \frac{r^3}{3})$$

Cas Section fonction de r Pour une section S fonction de r : $s = s(r) = a*r+b$ avec $a = \frac{S(L)-S(0)}{L}$ et $b = S(0)$

Avec les condition limites:

$$u(0) = 0$$

$$\frac{du}{dr}\Big|_{r=L} = 0$$

On trouve une solution analytique :

$$u(r)_{S(r)} = \frac{\omega^2\rho}{36a^3E}\Big(ar(-4a^2r^2-3abr+6b^2)-6(b-2aL)(aL+b)^2\ln(ar+b)+6\ln(b)(b-2aL)(aL+b)^2\Big)$$

Résolu avec Wolfram : This link

1.3 Approximation par éléments finis

1.3.1 Forme de l'approximation dans un maillage

Pour un maillage M^h nous avons une aproximation sur un élément k de la forme :

$$u_k^h = \Big(\frac{U(r_k) - U(r_{k-1})}{r_k - r_{k-1}}\Big)r + \big(\frac{r_k U(r_{k-1}) - r_{k-1}U(r_k)}{r_k - r_{k-1}}\Big)$$

1.3.2 Matrice A*T=B question 2 et 3

1.4 Programme

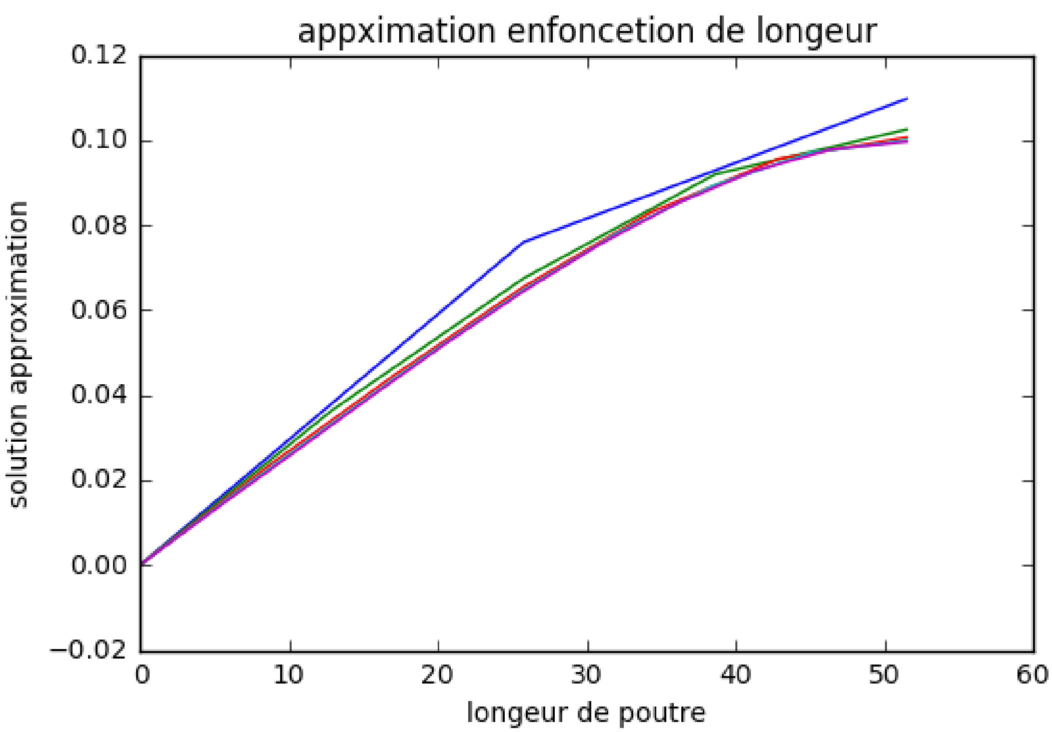
```
In [59]: #Foction S(r)
         def section(L,N,k,S0,SL):
             a=(SL-S0)/L
             b=S0
             return ((a*(L/N)*(k-1)+b)+a*(L/N)*k+b)/2

         #Algorithme de construction de la Matrice A et du Vecteur B
         def matrixA(E,L,N,S0,SL):
             A=np.zeros((N+1,N+1))
             h=L/N
             S1=section(L,N,1,S0,SL)
             SN=section(L,N,N,S0,SL)
             A[0,0]= (E*S1)/h
             A[0,1]= -(E*S1)/h
```

```
    A[1,0]= -(E*S1)/h
    A[1,1]= (E*S1)/h
    i=1
    while i<N:
        S=section(L,N,i,S0,SL)
        A[i,i]=A[i,i]+(E*S)/h
        A[i,i+1]=-(E*S)/h
        A[i+1,i]=-(E*S)/h
        A[i+1,i+1]=(E*S)/h
        i=i+1
    return A

def vecteurB(rho,omega,N,S0,SL):
    B=np.zeros(N+1)
    h=L/N
    S1=section(L,N,1,S0,SL)
    SN=section(L,N,N,S0,SL)
    R=1*L/N #k-0=0 donc r1
    B[0]=rho*omega*omega*S1*h*(R/3)
    B[1]=rho*omega*omega*S1*h*(R/6)
    i=1
    while i<N:
        R1=i*L/N
        R2=(i+1)*L/N
        S=section(L,N,i,S0,SL)
        B[i]=B[i]+rho*omega*omega*S*h*(R1/3+R2/6)
        B[i+1]=rho*omega*omega*S*h*(R1/3+R2/6)
        i=i+1
    return B
```

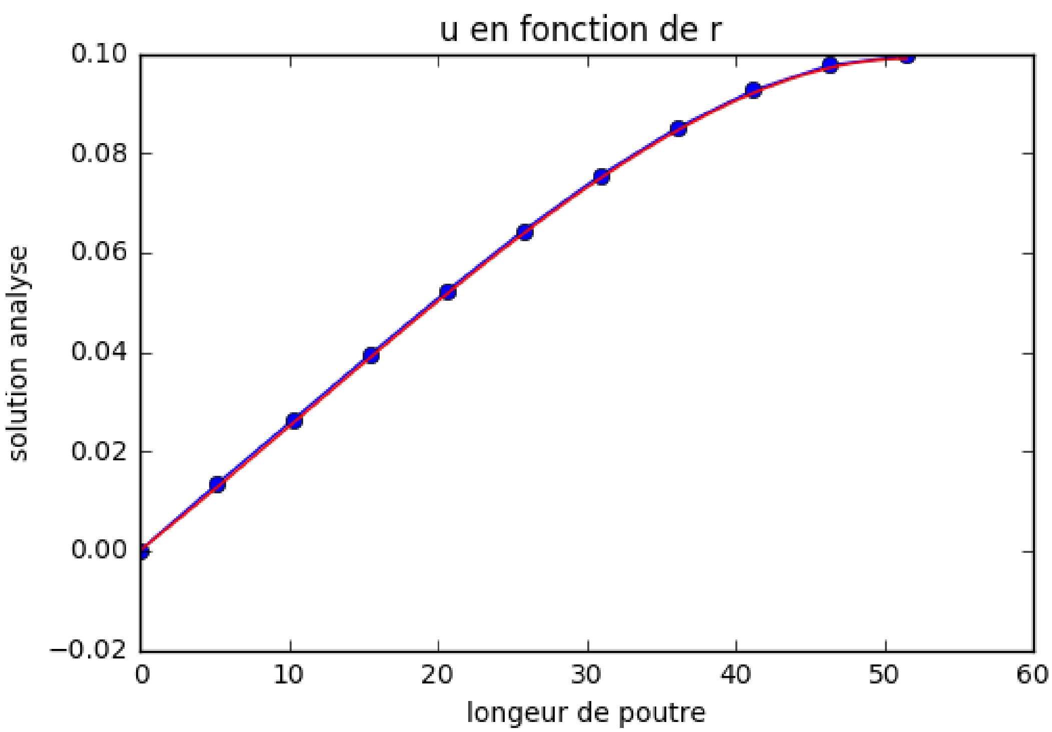
In [60]:
```
#Paramettre du problème:

S0=16.2
SL=6.7
L=51.5
err=0.076
rho=1600
E=21300*10**6
omega=2*np.pi
N=10

A=matrixA(E,L,N,S0,SL)
B=vecteurB(rho,omega,N,S0,SL)
#Condition limite:
A[0,0]=1
A[0,1]=0
B[0]=0
```

```
    #Resolution du sytème
    uh=np.linalg.solve(A,B) #uh - approximation
```

```
In [71]: for N in [2,4,6,8,10]:
             S0=16.2
             SL=6.7
             L=51.5
             err=0.076
             rho=1600
             E=21300*10**6
             omega=2*np.pi
             A=matrixA(E,L,N,S0,SL)
             B=vecteurB(rho,omega,N,S0,SL)
         #Condition limite:
             A[0,0]=1
             A[0,1]=0
             B[0]=0

         #Resolution du sytème
             uh=np.linalg.solve(A,B) #uh - approximation
             r=np.linspace(0,L,num=N+1)
             plt.plot(r,uh)
             plt.ylabel('solution approximation')
             plt.xlabel('longeur de poutre')
             plt.title('appximation enfoncetion de longeur')
```



In [62]:
```
#Etude de l'érreur relative

#Solution analytique
def u1(E,L,rho,omega,N,S0,SL,r): #S constant
    y=((rho*omega**2*r)*(L**2-(r**2)/3))/(E*2)
    return y

# def u2(E,L,rho,omega,N,S0,SL,x): #S variable
#     a=(SL-S0)/L
#     b=S0
#     c1=(a*L+b)*((rho*(omega**2)*(b**2))/(36*E*a**2)-(rho*(omega**2)*(b*l
#     y=((rho*(omega**2)*(b**2)*x)/(6*E*a**2)-(rho*(omega**2)*(b*x**2))/(
#     return y

def u2(E,L,rho,omega,N,S0,SL,r): #S variable
    a=(SL-S0)/L
    b=S0
    y=((omega**2*rho)/(36*E*a**3))*(a*r*(-4*a**2*r**2-3*a*b*r+6*b**2)-6*(b
    return y


#figure 1des petie valeur de N (ex N=5) + u analytique
x=np.linspace(0,L,100)
r=np.linspace(0,L,N+1)
u=u2(E,L,rho,omega,N,S0,SL,x)
p1=plt.plot(r,uh,marker='o')
p2=plt.plot(x,u,'r')
plt.xlabel('longeur de poutre')
plt.ylabel('solution analyse')
plt.title("u en fonction de r ")  # Problemes avec accents (plot_directive
plt.show()
```




```
In [70]: #figure numero 2 - err en fonction de r (mm)

         # def errorU(E,L,rho,omega,N,S0,SL):
         #     i=0
         #     rk=0
         #     error=np.zeros(N+1)
         #     while rk<=L:
         #         rk=i*L/N
         #         u=u2(E,L,rho,omega,N,S0,SL,rk)
         #         error[i]=np.absolute(u-uh[i])/(u)
         #         i=i+1
         #     return error

         u=u2(E,L,rho,omega,N,S0,SL,r)
         print len(u), len(uh)
         print u, uh
         error=np.absolute(u-uh)/u

         #http://ufrmeca.univ-lyon1.fr/moodle/mod/url/view.php?id=173

         p3=plt.plot(r,error,marker='o')
         plt.title("erreur")  # Problemes avec accents (plot_directive) !
         plt.show()
```

```
        #figure 3 : norme de err en fonction de h (1/N) et montrer que err diminue

        #+légende et commentaire
```

```
13 13
[  4.77428841e-17   1.04994553e-02   2.13132203e-02   3.22483767e-02
   4.31055185e-02   5.36771242e-02   6.37453835e-02   7.30792446e-02
   8.14303191e-02   8.85270708e-02   9.40663450e-02   9.77006295e-02
   9.90181729e-02] [ -2.61745638e-16   1.09878708e-02   2.17803991e-02   3.27040642
   4.35564552e-02   5.41270016e-02   6.41949288e-02   7.35265304e-02
   8.18714633e-02   8.89576149e-02   9.44838234e-02   9.81092796e-02
   9.94376088e-02]
```