Trabalho1

July 3, 2020

1 Metodo dos Elementos Finitos

Universidade Federal Fluminense

Disciplina ministrada pelo Prof. Marco Ferro

marco.ferro@uol.com.br

Aluno Noé de Lima

noe_lima@id.uff.br

Este trabalho visa aplicar o MEF para resolver uma equação diferencial ordinária de segunda ordem.

Primeiro semestre de 2020

[1]:
```
%display latex
from numpy import array,zeros,math,linspace
from scipy.interpolate import interp1d
from scipy.linalg import solve
import matplotlib.pyplot as plt
plt.close('all') # apaga plotagens anteriores
latex.add_to_preamble('\\usepackage[english,brazil]{babel}')
!uname -a
```

```
Linux DESKTOP-CR7O8A2 4.19.104-microsoft-standard #1 SMP Wed Feb 19 06:37:35 UTC
2020 x86_64 x86_64 x86_64 GNU/Linux
```

Contents

1.1 Exercício Teórico de MEF - 2 Elementos

Dada a Equação Diferencial Ordinária de Segunda Ordem abaixo:

$$\frac{d^2u}{dx} - u = 0 \tag{1}$$

Sua solução analítica é:

$$u\left(x\right) = \frac{e^x - e^{-x}}{e - e^{-1}} \tag{2}$$

Este exercício consiste em resolver esta equação numericamente utilizando o Método dos Elementos Finitos com 2 elementos, a princípio sem a utilização de programação. O intervalo do domínio utilizado será $0 \leq x \leq 1$, e as condições de contorno são:

$$\begin{cases} u\left(0\right) &= 0 \\ u\left(1\right) &= 1 \end{cases}$$

Assim, considerando 2 elementos, temos 3 nós. Cada elemento tem comprimento $h_e = \frac{1}{2}$.

1.1.1 Elementos

Temos os seguintes nós nos elementos:

$$\begin{bmatrix} u_1 &= 0 \\ u_2 &= \frac{1}{2} \\ u_3 &= 1 \end{bmatrix}$$

Portanto:

$$\begin{cases} 0 \leq x_1 \leq \frac{1}{2} \\ \frac{1}{2} \leq x_2 \leq 0 \end{cases}$$

Cada elemento tem comprimento $h_e = \frac{1}{2}$. As funções de interpolação serão, então:

$$\phi_1^e\left(x_e\right) = \frac{h_e - x_e}{h_e} = \frac{\frac{1}{2} - x_e}{\frac{1}{2}} = 1 - 2x_e$$

e

$$\phi_2^e\left(x_e\right) = \frac{x_e}{h_e} = \frac{x_e}{\frac{1}{2}} = 2x_e$$

Matriz de Rigidez dos Elementos Os elementos da Matriz de Rigidez local $\mathbf{K}_e$ são:

$$K_{ij}^e = \int_0^{\frac{1}{2}} \left(\frac{dN_i}{dx_e} \frac{dN_j}{dx_e} + N_i N_j \right) dx_e$$

Assim:

[2]:
```
x = var('x')
K_11 = integral((-2)*(-2)+(1-2*x)**2,x,0,1/2)
K_12 = integral((-2)*(2)+(1-2*x)*(2*x),x,0,1/2)
K_21 = integral((2)*(-2)+(2*x)*(1-2*x),x,0,1/2)
K_22 = integral((2)*(2)+(2*x)**2,x,0,1/2)
K_e = Matrix([[K_11,K_12],[K_21,K_22]])
print(K_e)
```

```
[  13/6 -23/12]
[-23/12   13/6]
```

Portanto, a Matriz de Rigidez Local fica:

$$\mathbf{K_e} = \begin{bmatrix} \frac{13}{6} & \frac{-23}{12} \\ \frac{-23}{12} & \frac{13}{6} \end{bmatrix}$$

A Matriz de Rigidez Global sera:

$$\mathbf{K_g} = \begin{bmatrix} K_{11} & K_{12} & 0 \\ K_{21} & K_{22} + K_{11} & K_{12} \\ 0 & K_{21} & K_{22} \end{bmatrix}$$

Portanto:

[3]:
```
K_g = Matrix([[K_11,K_12,0],[K_21,K_22+K_11,K_12],[0,K_21,K_22]])
print(K_g)
```

```
[  13/6 -23/12      0]
[-23/12   13/3 -23/12]
[     0 -23/12   13/6]
```

Ou seja,

$$\mathbf{K_g} = \begin{bmatrix} \frac{13}{6} & -\frac{23}{12} & 0 \\ -\frac{23}{12} & \frac{13}{3} & -\frac{23}{12} \\ 0 & -\frac{23}{12} & \frac{13}{6} \end{bmatrix}$$

1.1.2 Cálculo das Forças Nodais

Elemento 1

$$\begin{aligned} f_1^1 &= -N_1 \frac{d\bar{u}(0)}{dx} = -\frac{d\bar{u}(0)}{dx} \\ f_2^1 &= 0 \end{aligned}$$

Elemento 2

$$\begin{aligned} f_1^2 &= 0 \\ f_2^2 &= N_2 \frac{d\bar{u}(1)}{dx} = \frac{d\bar{u}(1)}{dx} \end{aligned}$$

Portanto,

$$\vec{f} = \begin{Bmatrix} -\frac{d\bar{u}(0)}{dx} \\ 0 \\ \frac{d\bar{u}(1)}{dx} \end{Bmatrix}$$

1.1.3 Equação final

Temos, portanto,

$$\mathbf{K} \cdot \vec{u} = \vec{f}$$

$$\begin{bmatrix} \frac{13}{6} & -\frac{23}{12} & 0 \\ -\frac{23}{12} & \frac{13}{3} & -\frac{23}{12} \\ 0 & -\frac{23}{12} & \frac{13}{6} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} -\frac{d\bar{u}(0)}{dx} \\ 0 \\ \frac{d\bar{u}(1)}{dx} \end{Bmatrix}$$

Ou,

$$\begin{array}{ccccc} 2,167u_1 & -1,917u_2 & 0u_3 & = & -\frac{d\bar{u}(0)}{dx} \\ -1,917u_1 & 4,334u_2 & -1,917u_3 & = & 0 \\ 0u_1 & -1,917u_2 & 2,167u_3 & = & \frac{d\bar{u}(1)}{dx} \end{array}$$

Rearrumando temos a seguinte equação:

$$\begin{bmatrix} 1 & -\frac{23}{12} & 0 \\ 0 & \frac{13}{3} & 0 \\ 0 & -\frac{23}{12} & -1 \end{bmatrix} \begin{Bmatrix} \frac{d\bar{u}(0)}{dx} \\ u_2 \\ \frac{d\bar{u}(1)}{dx} \end{Bmatrix} = -\begin{Bmatrix} \frac{13}{6} \\ -\frac{23}{12} \\ 0 \end{Bmatrix} u_1 - \begin{Bmatrix} 0 \\ -\frac{23}{12} \\ \frac{13}{6} \end{Bmatrix} u_3$$

[4]:
```
K = Matrix([[1,K_12,0],[0,K_22+K_11,0],[0,K_21,-1]])
f = -0*vector([K_11,K_21,0])-1*vector([0,K_12,K_22])
print("K' =\n", K, "\n\n")
print("f' =", f, "\n\n")

print("u' =", K\f)
```

```
K' =
 [     1 -23/12      0]
[      0   13/3      0]
[      0 -23/12     -1]
```

```
f' = (0, 23/12, -13/6)
```

```
u' = (529/624, 23/52, 823/624)
```

Logo,

$$\begin{bmatrix} 1 & -\frac{23}{12} & 0 \\ 0 & \frac{13}{3} & 0 \\ 0 & -\frac{23}{12} & -1 \end{bmatrix} \begin{Bmatrix} \frac{d\bar{u}(0)}{dx} \\ u_2 \\ \frac{d\bar{u}(1)}{dx} \end{Bmatrix} = \begin{Bmatrix} 0 \\ \frac{23}{12} \\ -\frac{13}{6} \end{Bmatrix}$$

Portanto,

$$\frac{d\bar{u}\,(0)}{dx} = \frac{529}{624} = 0,8478$$

$$u_2 = \frac{23}{52} = 0,4423$$

$$\frac{d\bar{u}\,(1)}{dx} = \frac{823}{624} = 1,3189$$

1.2 Exercício Prático de Programação

Os códigos a seguir resolvem o problema que consiste em criar um programa na linguagem Python para resolver, numericamente, a equação da Seçao anterior no domínio $x \leq 0 \leq 1$, utilizando para tanto, o Método dos Elementos Finitos. Entre os parâmetros de entrada, deverá estar o números de elementos para discretizar o sistema (malha), para, então, comparar o erro pencentual das soluções em relação à solução analítica.

1.3 Implementação da Solução Analítica (Exata)

[5]:
```python
def yexata(x):
    return (math.exp(1)**(x)-math.exp(1)**(-x))/(math.exp(1)-math.exp(-1))
```

1.4 Implementação do MEF

Na implementação do Método dos Elementos Finitos, os seguintes dados de entrada são necessários:

- Limites do Domínio
- Condições de Contorno (Dirichlet)
- Número de Elementos na Malha

Assim, a função retorna dois vetores:

- O Domínio discretizado
- A solução nos nós da Malha

A solução dos pontos intermediários é dada por interpolação simples entre os nós adjascentes.

[6]:
```
def mefn(a, b, ua, ub, n):
    x = array(linspace(a,b,n+1)) # Domínio
    # f = array([linspace(0,0,n+1)]).transpose() # Vetor de força nodal (matriz␣
 ↪coluna de zeros)
    K = zeros([(n+1),(n+1)]) # Matriz de Rigidez Global
    he = (b-a)/n # Subdomínio
    for i in range(n): # Montagem da Matriz de Rigidez Global
        K[i,i] += (1/he)+(he/3)
        K[i,i+1] += (-1/he)+(he/6)
        K[i+1,i] += (-1/he)+(he/6)
        K[i+1,i+1] += (1/he)+(he/3)
    # Rearranjo do Sistema
    # f -= ua*K[:,0] + ub*K[:,n] # Implementação alternativa com inicialização␣
 ↪de f
    f = - ua*K[:,0] - ub*K[:,n]
    K[:,0] *= 0
    K[:,n] *= 0
    K[0,0] = 1.
    K[n,n] = -1.
    # Solução do Sistema
    u = solve(K,f) # Vetor de solução
    u[0],u[n] = ua,ub # Substituição de du(a)/dx e du(b)/dx por u(a) e u(b)
    return x, u
```

1.4.1 Gráfico Comparativo das Soluções

Para comparar as soluções, será utilizada uma função para entrar com o número de elementos e gerar o gráfico comparativo

[7]:
```
def interpol(xd,yd,x): # Função para interpolar u linearmente entre os nós
    return interp1d(xd,yd,kind='linear')(x)
def plotmef(n, nplot): # Função que plota nplot pontos discretizados em n␣
 ↪elementos
    a,b = 0,1 # Limites de Integração
    ua,ub = yexata(a),yexata(b) # Condições de Contorno
    xd,yd = mefn(a,b,ua,ub,n)
    x = linspace(0.,1.,11)
    u = interpol(xd,yd,x)
    y = yexata(x)
    ref = int(len(x)/2)
    erro = 100*(y[ref]-u[ref])/y[ref]
    return x,u,erro
```

[8]:
```
nel = [1,2,3,5,10,100]
amostra = 50
```

```
comp = linspace(0,1,amostra)
plt.figure(1)
plt.title('Gráfico Comparativo')
plt.xlabel('x')
plt.ylabel('u(x)')
for e in nel:
    x,u,erro = plotmef(e,amostra)
    plt.plot(x,u,label='MEF '+str(e)+' - Erro='+str(round(erro,3))+'%')
plt.plot(comp,yexata(comp),'k--',label='Solução Exata')
plt.legend()
plt.show()
```