Método dos Elementos Finitos

Claudemir Woche e Márcio Barros

19 de outubro de 2020

1 PVC1

Primeiro Problema de Valor de Contorno.

$$PVC1 : \begin{cases} \frac{d^2y(x)}{dx^2} - y(x) = 0 \\ y(0) = 0 \\ y(1) = 1 \end{cases} \tag{1}$$

Este PVC busca a função $y(x)$ que satisfaz a equação diferencial e as condições de contorno especificadas ($y(0) = 0$ e $y(1) = 0$). Neste exemplo, o domínio é o intervalo $[0, 1] \subset \mathbf{R}$.

1.1 Solução do Problema

A solução exata de um PVC deve satisfazer tanto a Equação Diferencial do problema quanto as condições de contorno. Assim, para o problema (1) – PVC1 –, a solução exata é

$$y(x) = \frac{1}{e^{-1} - e}(e^{-x} - e^x) \tag{2}$$

A solução aproximada obtida pelo método das diferenças finitas está compilada abaixo.

Para N = 8 temos o seguinte sistema de equações:

$$\begin{bmatrix} 16.08333 & -7.97917 & 0 & 0 & 0 & 0 & 0 \\ -7.97917 & 16.08333 & -7.97917 & 0 & 0 & 0 & 0 \\ 0 & -7.97917 & 16.08333 & -7.97917 & 0 & 0 & 0 \\ 0 & 0 & -7.97917 & 16.08333 & -7.97917 & 0 & 0 \\ 0 & 0 & 0 & -7.97917 & 16.08333 & -7.97917 & 0 \\ 0 & 0 & 0 & 0 & -7.97917 & 16.08333 & -7.97917 \\ 0 & 0 & 0 & 0 & 0 & -7.97917 & 16.08333 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 7.97917 \end{bmatrix} \tag{3}$$

Com as seguintes comparações:

Elementos Finitos	Diferenças Finitas	Solução Exata	Erro Relativo
0	0	0	0
0.10662	0.10666	0.10664	$2e^{-5}$
0.21491	0.21499	0.21495	$4e^{-5}$
0.32657	0.32666	0.32662	$5e^{-5}$
0.44334	0.44347	0.44340	$6e^{-5}$
0.56706	0.56719	0.56713	$7e^{-5}$
0.69966	0.69978	0.69972	$6e^{-5}$
0.84323	0.84330	0.84326	$3e^{-5}$
1	1	1	0

1.2 Algoritmo

Podemos observar o código, que recebe o número de divisões do intervalo e o comprimento dessas divisões, devolverá a matriz $A \in \mathbb{R}^{n\times n}$ e $b \in \mathbb{R}^{n\times 1}$, do sistema, e o $y \in \mathbb{R}^{1\times n}$ que é solução do sistema (3), que são nossos valores aproximados para $f(x_i) \simeq y_i$, x_i discretos.

```
def elem_finitos(N, dx):
    phi = [lambda s: (1-s)/2 , lambda s: (1+s)/2]
    phi_linha = [1/4,-1/4,1/4]                                          #[phi[0]'**2,phi[0]'*phi[1]',phi[1]'**2]

    f = lambda dx: lambda s: ((1-s)/2)**2 * dx/2
    phi_product1 = f(dx)
    g = lambda dx: lambda s: (1/4) * (1-s) * (s+1) * dx/2
    phi_product2 = g(dx)
    h = lambda dx: lambda s: ((1+s)/2)**2 * dx/2
    phi_product3 = h(dx)
    phi_list = [phi_product1,phi_product2,phi_product3]                 #[phi[0]**2,phi[0]*phi[1],phi[1]**2]
    A_aux = []
    aux = -1
    for k in range(N+1):
        A_aux.append(np.zeros((2,2)))
        for i in range(2):
            for j in range(i,2):
                aux+=1

                part1 = integrate.quad(lambda x: phi_linha[aux] * 2/dx ,a = -1,b = 1)[0]
                part2 = integrate.quad(phi_list[aux],a = -1,b = 1)[0]
                A_aux[k][i,j] = part1 + part2
                A_aux[k][j, i] = A_aux[k][i, j]
                if aux == 2: aux = -1

    A = np.zeros((N+2, N+2))
    for k in range(N):
        for i in range(2):
            for j in range(2):
                A[k + i, k + j] += A_aux[k][i, j]

    A = A[1:-2, 1:-2]
    b = np.zeros(N-1)
    b[-1] -= A[-2, -1]
    y = np.linalg.solve(A,b)
    return A, b, y
```