

INTRODUÇÃO AOS MÉTODOS COMPUTACIONAIS EM BIOMEDICINA

3º ANO | LEBIOM

Relatório do Trabalho nº 1

Realizado por:

Ana Rita Damas Mendes (99641)
Inês Pereira Marques (99674)
Sofia Silvestre de Oliveira (99715)

ritadmendes@tecnico.ulisboa.pt
ines.p.marques@tecnico.ulisboa.pt
sofia.de.oliveira@tecnico.ulisboa.pt

Grupo 4

2022/2023 – 1º Semestre, P2

Conteúdo

1	Objetivos	2
2	Fundamentação teórica	2
3	Desenvolvimentos efetuados	3
3.1	Obtenção do sistema de equações lineares	3
3.2	Forças aplicadas nas extremidades da barra	4
4	Exemplos de situações com e sem forças impostos nas extremidades da barra	5
5	Evolução do erro com a diminuição do espaçamento dos nós	6
6	Discussão dos resultados obtidos	7
	Appendices	9
A	Implementação de código em MatLab	9

1 Objetivos

Este trabalho visa obter o que está estipulado no enunciado 1.4., o qual nos foi atribuído. Desta forma, este incidirá no cálculo aproximado do deslocamento de uma barra (discretizada numa malha de espaçamento uniforme), com o auxílio de um programa implementado em MatLab. Através da equação representativa do equilíbrio estático de uma barra, substituiu-se a segunda derivada por uma aproximação através do método das diferenças finita.

De seguida, mostraremos exemplos utilizando esta metodologia para casos de deslocamento conhecido, tendo em conta cargas axiais distribuídas, forças concentradas aplicadas nas extremidades e deslocamentos conhecidos nas extremidades, tendo como última finalidade a análise da evolução do erro das aproximações efetuadas, com a diminuição do espaçamento, e com diferentes ordens (primeira e segunda) para a diferença finita da primeira derivada, aplicadas nos casos com forças concentradas nas extremidades.

2 Fundamentação teórica

O objetivo do método das diferenças finitas é transformar um problema composto por equações diferenciais em um formado por equações algébricas que aproximam a solução em cada ponto x_i . Esta aproximação pode ser realizada de diferentes formas, dependendo da precisão desejada, da natureza do problema e das condições de fronteira.

O primeiro passo deste método é a chamada discretização do domínio da variável independente. A discretização consiste em dividir o domínio de cálculo em um certo número de subdomínios. No nosso caso, o domínio é uma barra de comprimento L, que será dividida por N nós e a distância entre esse nós será h.

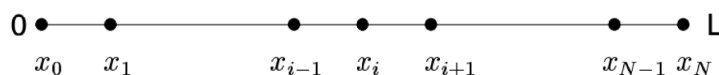


Figura 1: Discretização de uma barra de comprimento L

O segundo passo é gerar aproximações para as derivadas das variáveis dependentes que aparecem nas equações diferenciais, nos pontos discretos x_i , isto é, obter $u'(x_i)$ e $u''(x_i)$, utilizando apenas os valores destes pontos discretos, $u(x)$. Estas fórmulas de aproximação obtêm-se a partir da expansão da Série de Taylor e onde se pode observar o erro obtido inerente a essa aproximação. Por exemplo, dado que a Série de Taylor para a função $u(x)$ em torno de x_i é equivalente a:

$$u(x) = u(x_i) + \frac{1}{1!} \frac{du(x_i)}{dx} (x - x_i) + \frac{1}{2!} \frac{d^2u(x_i)}{dx^2} (x - x_i)^2 + \dots + \frac{1}{n!} \frac{d^nu(x_i)}{dx^n} (x - x_i)^n + O((x - x_i)^{n+1}) \quad (1)$$

Então,

$$u_{n+1} = u(x_i + h) = u(x_i) + u'(x_i) \cdot h + \frac{u''(x_i)}{2} \cdot h^2 + \frac{u'''(\xi_+)}{3!} \cdot h^3 \quad (2)$$

$$u_{n-1} = u(x_i - h) = u(x_i) - u'(x_i) \cdot h + \frac{u''(x_i)}{2} \cdot h^2 + \frac{u'''(\xi_-)}{3!} \cdot h^3 \quad (3)$$

Desprezando os termos de ordem igual ou superior a 2 das equações (2) e (3) encontramos duas aproximações para $u'(x_i)$, respetivamente aproximação progressiva e regressiva, sendo a aproximação central a média dessas duas diferenças finitas. Por exemplo, a aproximação de $u'(x)$ por diferença finita progressiva ficaria

$$u'(x) = \frac{u(x+h) - u(x)}{h} - h \left(\frac{u''(\xi_+)}{2} \right) \quad (4)$$

com erro inerente de ordem h e por diferença finita central ficaria

$$u'(x) = \frac{u(x+h) - u(x-h)}{2h} - h^2 \left(\frac{u'''(\xi_+) - u'''(\xi_-)}{2 \cdot 3!} \right) \quad (5)$$

com erro inerente de ordem h^2 . Além disso, também podemos reescrever formas diferentes usando 3 pontos (2^a ordem).

De modo análogo à primeira derivada, podemos obter a aproximação central da segunda derivada da seguinte forma:

$$\frac{\partial^2 u}{\partial x^2}_{x=x_i} = \frac{\partial}{\partial x} \left(\frac{\partial u}{\partial x} \right) \approx \frac{1}{\Delta x} \left(\frac{\partial u}{\partial x}_{i+1} - \frac{\partial u}{\partial x}_i \right) = \frac{1}{\Delta x} \left(\frac{u_{i+1} - u_n}{\Delta x} - \frac{u_i - u_{i-1}}{\Delta x} \right) = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} \quad (6)$$

Ficam claras, agora, duas características do método de diferenças finitas: a aplicação das equações diferenciais é local, isto é, em cada ponto x_i , e a solução obtida é composta por um conjunto finito de pontos (e onde os valores da solução são conhecidos).

3 Desenvolvimentos efetuados

3.1 Obtenção do sistema de equações lineares

Caso na barra esteja a atuar apenas uma carga axial distribuída, a partir do enunciado do problema, podemos integrar duas vezes a equação de modo a obter uma função implícita do deslocamento da seguinte forma, caso $f(x)$ fosse uma constante:

$$EA \frac{\partial^2 u}{\partial x^2} = -f(x) \rightarrow EA \frac{\partial u}{\partial x} = -f(x)x + c_1 \rightarrow EAu(x) = -\frac{f}{2}x^2 + c_1x + c_2 \quad (7)$$

Para uma equação deste tipo, onde x varia de 0 a L (comprimento da barra) e sabemos *a priori* os deslocamentos nessas mesmas extremidades ($u(0) = u_1$ e $u(L) = u_N$), para descobrir o deslocamento nos pontos interiores da barra, podemos usar uma diferença finita central como aproximação da segunda derivada do deslocamento da seguinte forma:

$$E \cdot A \cdot \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} = -f(x) \quad (8)$$

Deste modo, forma-se o sistema linear seguinte para os nós 2 a N-1 da barra, tendo em conta que

$$\text{o MATLAB considera } x_1 \text{ o início da barra: } \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots \\ -2 & 1 & 0 & 0 & 0 & \dots \\ 1 & -2 & 1 & 0 & 0 & \dots \\ 0 & 1 & -2 & 1 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \dots \\ u_N \end{bmatrix} = \begin{bmatrix} u_1 \\ f(x_2) \cdot \frac{h^2}{-E \cdot A} \\ f(x_3) \cdot \frac{h^2}{-E \cdot A} \\ f(x_4) \cdot \frac{h^2}{-E \cdot A} \\ \dots \\ u_N \end{bmatrix}$$

Na função Usemforca.m colocou-se o valor 1 na posição (1,1) e (N,N) uma vez que se assume que o lado direito da equação equivale a u_1 e u_N já conhecidos. A diferença finita central é escolhida em detrimento da diferença finita progressiva ou regressiva devido ao seu menor erro inerente equivalente a $-\frac{1}{12} \cdot h^2 \cdot u''''(\xi)$

A resolução do sistema matricial obtido irá fornecer os deslocamentos pretendidos.

3.2 Forças aplicadas nas extremidades da barra

À semelhança do que foi feito para dois deslocamentos conhecidos, os deslocamentos dos pontos interiores são calculados tendo por base uma aproximação da segunda derivada pela diferença finita central mas o deslocamento do nó onde é aplicada a força (nó da extremidade direita nos nossos exemplos) foi calculado através da expressão da diferença finita regressiva para a primeira derivada de primeira e segunda ordem, tendo sido comparados os resultados entre estas últimas.

Se a força concentrada fosse aplicada na extremidade inicial e o deslocamento conhecido na última extremidade usaria-se a diferença finita progressiva, também apresentada no código caso o utilizador pretender resolver casos semelhantes. Para a obtenção deste deslocamento, considera-se que F_{total} consiste na soma da carga distribuída na extremidade em análise com a força concentrada aplicada na mesma.

Considerando, por exemplo, $f(x)$ como constante; uma força aplicada na extremidade direita e deslocamento conhecido à esquerda:

$$\begin{aligned} A \cdot E \cdot \frac{\partial^2 u}{\partial x^2} = -f(x) &\Leftrightarrow A \cdot E \cdot \frac{\partial u}{\partial x_{x=L}} = -f(L) \times L + c_1 \quad (9) \\ A \cdot E \cdot \frac{\partial u}{\partial x_{x=L}} = F_{total} &\Leftrightarrow -f(L) \cdot L + c_1 = f(L) + F_{concentrada} \quad (10) \end{aligned}$$

Através da equação (10), consegue-se chegar ao valor de c_1 . De forma a obter a função exata de $u(x)$, ao integrar duas vezes a equação inicial (9), encontra-se o valor de c_2 a partir de $u(0)$.

Para encontrar a solução através das diferenças finitas substituímos $\frac{\partial u}{\partial x_{x=L}}$ por $\frac{u(L)-u(L-h)}{h}$ (1ª ordem) ou por $\frac{3u(L)-4u(L-h)+u(L-2h)}{2h}$ (2ª ordem) e os restantes deslocamentos através da mesma forma que em 3.1., ou seja a matriz coeficientes utilizando a regressiva de 2ª ordem neste caso seria:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots \\ -2 & 1 & 0 & 0 & 0 & \dots \\ 1 & -2 & 1 & 0 & 0 & \dots \\ 0 & 1 & -2 & 1 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & 1 & -4 & 3 \end{bmatrix}$$

4 Exemplos de situações com e sem forças impostas nas extremidades da barra

Sem força aplicada nas extremidades

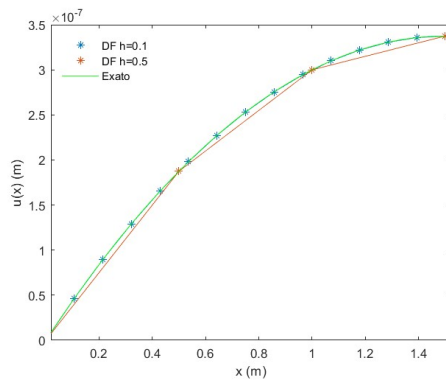
Nos seguintes exemplos vamos utilizar: $E = 100 \text{ GPa}$; $A = 400 \text{ mm}^2$; $L=1.5\text{m}$

1. Carga axial constante $f = 12 \text{ N/m}$; Condições fronteira: $u(0)=0\text{m}$; $u(\text{end})=0.3375\mu\text{m}$

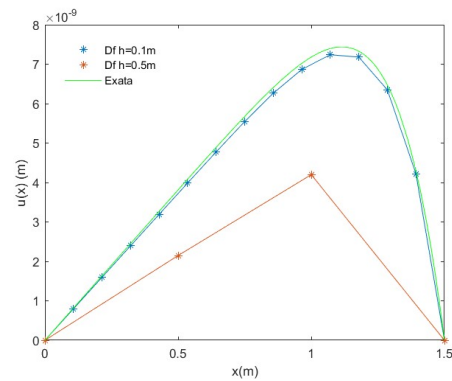
$$u(x)_{\text{exato}} = (-0.15x^2 + 0.45x)10^{-6} \text{ N/m}$$

2. Carga axial $f = x^6 \text{ N/m}$
Condições fronteira: $u(0)=0\text{m}$; $u(\text{end})=0\text{m}$

$$u(x)_{\text{exato}} = \frac{\frac{x^8}{56} - \frac{L^7}{56}x}{-EA} \text{ N/m}$$



(a) Exemplo 1



(b) Exemplo 2

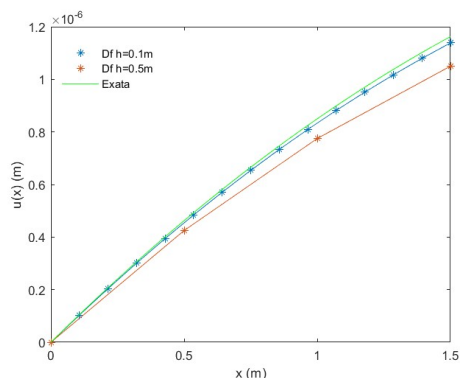
Figura 2: Deslocamentos com diferenças finitas e Deslocamento exato

Força aplicada na extremidade direita

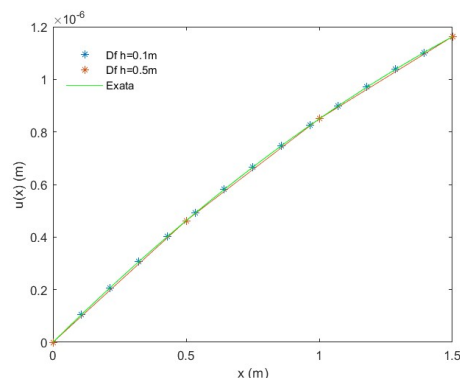
Nos seguintes exemplos vamos utilizar: $E = 100 \text{ GPa}$; $A = 400 \text{ mm}^2$; $L=1.5\text{m}$

3. Carga axial constante $f = 12 \text{ N/m}$;
Condições fronteira: $u(0)=0$; $F_{\text{esq}}=10\text{N}$

$$u(x)_{\text{exato}} = \frac{\frac{fx^2}{2} - (F_{\text{esq}} + fL + f(L))x}{-EA} \text{ N/m}$$



(a) Primeira derivada de Primeira ordem

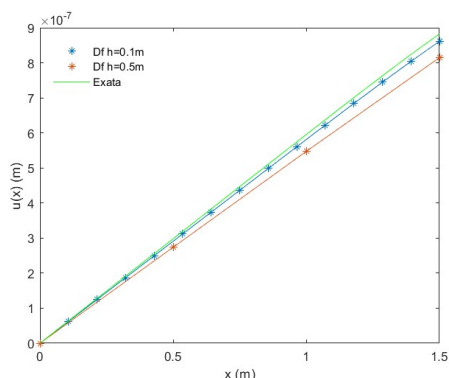


(b) Primeira derivada de Segunda ordem

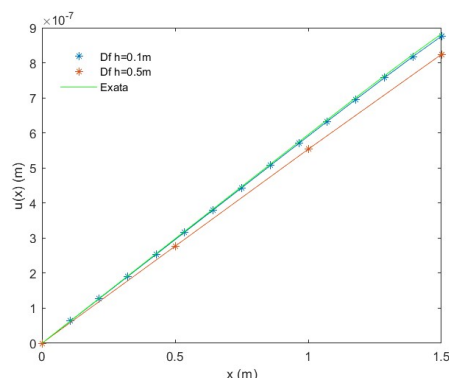
Figura 3: Deslocamentos com diferenças finitas e Deslocamento exato

4. Carga axial $f=x^6$ N/mCondições fronteira: $u(0)=0$ m ; $u(\text{end})=0$ m

$$u(x)_{\text{exato}} = \frac{\frac{x^8}{56} - (L^6 + F_{\text{esq}} + \frac{L^7}{7})x}{-EA} N/m$$



(a) Primeira derivada de Primeira ordem



(b) Primeira derivada de Segunda ordem

Figura 4: Deslocamentos com diferenças finitas e Deslocamento exato

5 Evolução do erro com a diminuição do espaçamento dos nós

Em relação ao gráfico (a) : Declive para carga axial constante : ≈ 0 Declive para carga axial x^6 : ≈ 1.9942 Em relação ao gráfico (b) (quando temos força aplicada na extremidade e carga axial x^6):Declive para primeira derivada primeira ordem : ≈ 0.9727

Declive para primeira derivada segunda ordem x^6 : ≈ 1.9545

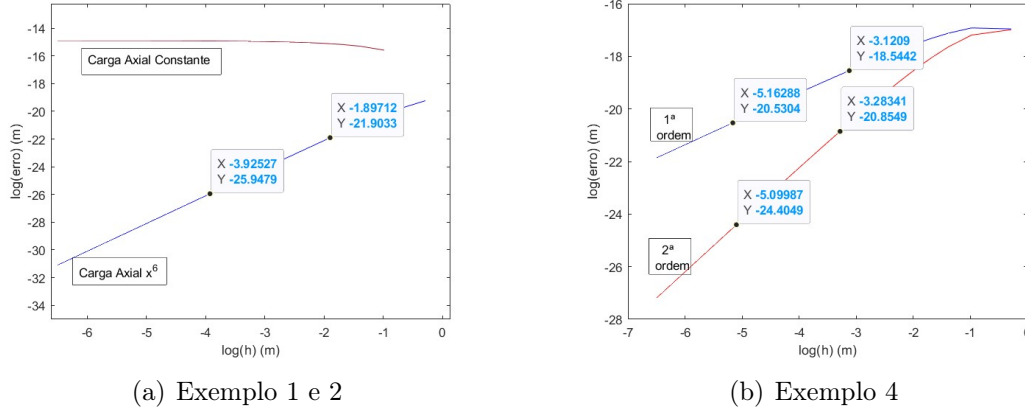


Figura 5: Evolução do erro com a diminuição do espaçamento dos nós para 500 iterações

6 Discussão dos resultados obtidos

Uma boa opção para visualizar o gráfico dos erros consiste em relacionar o valor do erro numa escala logarítmica em função de h , uma vez que se o erro $E(h)$ se comportar da forma:

$$E(h) \approx Ch^p \quad (11)$$

Então

$$\log | E(h) | \approx \log | C | + p \log(h) \quad (12)$$

Com o objetivo de confirmar esta relação, construímos no MATLAB um m-file (denominado errologSF.m e errologCF.m), onde fornecemos a função exata, o número de iterações que desejamos e todos parâmetros necessários à realização do problema com as diferenças finitas. O erro máximo terá em conta os erros de arredondamento e o erro inerente ao método numérico. Em geral, o erro de arredondamento aumentará com um aumento do número de cálculos (em cada iteração quando o N aumenta). Por outro lado, através dos gráficos é possível demonstrar que o erro inerente ao método numérico pode ser reduzido diminuindo o tamanho do h . Deste modo, numa escala logarítmica o erro evoluirá linearmente com um declive equivalente a p , a ordem de precisão do método.

No gráfico (a) da Fig.5, o gráfico do erro representado a azul tem um declive de aproximadamente 2, uma vez que essa foi a ordem da diferença finita utilizada para estimar o deslocamento da barra. Pela mesma razão (ordem do método utilizado), podemos explicar o declive de aproximadamente um para a reta azul no gráfico (b) da Fig.5, e o declive de cerca de 2, para a reta a vermelho no mesmo gráfico.

Ao analisarmos o declive da reta do erro para a mesma situação de uma força aplicada à direita, com carga axial distribuída igual a x^6 e a x^2 , obtivemos, para aproximações de 1ª ordem, declives de 0.9727 e de 0.9843, respetivamente.

Estes resultados vão de encontro ao esperado, uma vez que uma carga axial polinomial de maior ordem irá aumentar a ordem da função do deslocamento, o que pela expressão do erro associada à diferença finita regressiva de 1^a ordem (*Fig.6*), faz com que o erro associado a x^6 seja de ordem x^4 enquanto que o erro relativo a x^2 será de ordem 0, o que faz com que o declive da reta do erro para uma carga distribuída de x^2 seja superior ao de x^6 , devido a um menor erro associado, e portanto a uma maior ordem de precisão.

$$E_1'(x) = \frac{1}{2} \cdot h \cdot f''(\xi)$$

Figura 6: Erro associado à diferença finita regressiva de 1^a ordem

A uma ordem superior do método utilizado corresponde um maior declive da reta do erro e uma ordem de precisão superior. Isto acontece uma vez que o seu erro associado é proporcional a h^2 e a de 1^a ordem a h , ou seja quando h é muito pequeno a de 2^a ordem tem um erro muito inferior à de 1^a ordem. Uma vez que uma diminuição do h pode levar ao cancelamento subtrativo ou ao aumento do número de cálculos, os erros de truncamento diminuem à medida que os erros de arredondamento aumentam. Isto seria relevante caso os valores de deslocamento fossem muito pequenos, o que não sucede neste trabalho pois o Matlab tem um limite de algarismos significativos.

Referências

- [1] https://fenix.tecnico.ulisboa.pt/downloadFile/1126518382300922/Aula_07_2022.12.02_DiferencasFinitas.pdf
- [2] Jacquot, R. G., 38; Dewey, B. R. (n.d.). Solution of Static and Dynamic Beam Bending and Static Buckling Problems Using Finite Differences and MATLAB.
- [3] Zhao, S., 38; Pronina, Y. (n.d.). On the MATLAB Finite Element Modelling of an Elastic Plane with a Hole under Tension.
- [4] Chapra, S. C., Canale, R. P. (2010). *Numerical methods for engineers*. McGraw-Hill Higher Education ; London.

Apêndices

A Implementação de código em MatLab

Listings

1	Usemforca.m	9
2	Ucomforca.m	9
3	plotdeslocamento.m	10
4	errologSF.m	11
5	errologCF.m	12
6	ploterro.m	12

Listing 1: Usemforca.m

```

1 %Deslocamento de uma barra sujeito a carga axial a partir da Diferença
2 % Finita nos nós da barra (1,N) quando sabemos os deslocamentos
3 %das duas extremidades
4 function diferencafinita = Usemforca(E,A,L,N,P,dir,esq)
5     format long
6     h=L/(N-1);
7     l = -(E*A)/h^2;
8     M=zeros(N,N); % inicializa o da matriz M
9     b=zeros(N,1); % inicializa o do vector b
10    M(1,1)=1; b(1)=dir; % condições fronteira x=0
11    M(N,N)=1; b(N)=esq; % condições fronteira x=L
12    y=linspace(0,L,N);
13    for i=2:N-1
14        a1=1; a2=-2; a3=1;
15
16        M(i,i-1)=a1; % elemento esquerda da diagonal
17        M(i,i)=a2; % diagonal
18        M(i,i+1)=a3; % elemento direita da diagonal
19
20        % definição do interior do vector b
21        b(i)=P(y(i))/l;
22    end
23    %Resolve sistema
24    u = M\b;
25    diferencafinita = u;
26    plot(y,u, '-*')
27 end

```

Listing 2: Ucomforca.m

```

1 %Deslocamento de uma barra sujeito a carga axial a partir da Diferença
2 % Finita nos nós da barra (1,N) quando sabemos o deslocamentos

```

```

3 %da extremidade esq a for a aplicada direita
4 function diferencafinita = Ucomforca(e,A,L,N,P,esq,Fd)
5     h=L/(N-1);
6     l=(-e*A)/h^2;
7     M=zeros(N,N); % inicializa o da matriz Mde
8     b=zeros(N,1); % inicializa o do vector b
9
10    %%primeira ordem
11    M(N,N)=1;
12    M(N,N-1)=-1;
13    b(N)=( (P(L)+Fd)*h)/(A*e);
14
15    %%segunda ordem
16    %M(N,N)=3;
17    %M(N,N-1)=-4;
18    %M(N,N-2)=1;
19    %b(N)=( (P(L)+Fd)*2*h)/(A*e);
20
21    M(1,1)=1; b(1)=esq; % condi es fronteira x=0 deslocamentos
22    x=linspace(0,L,N);
23    for i=2:N-1
24        a1=1; a2=-2; a3=1;
25        M(i,i-1)=a1; % elemento esquerda da diagonal
26        M(i,i)=a2; % diagonal
27        M(i,i+1)=a3; % elemento direita da diagonal
28
29
30    % defini o do interior do vector b
31
32    b(i)=P(x(i))/l;
33    end
34    %Resolve sistema
35    u = M\b;
36    diferencafinita=u;
37    %plot(x,u,'-*)
38
39 end

```

Listing 3: plotdeslocamento.m

```

1 % Faz plot dos deslocamentos das diferen as finitas consoante quantos n s
2 % escolhermos e plot do dexato, para comparar
3
4 %Input do utilizador
5 E=input('M dulo de young (Pa) ');
6 A=input(' rea da sec o da barra (m2) ');

```

```

7 L=input('Comprimento da barra (m)');
8 ftxt=input('Carga axial distribu da=? N/m','s');
9 ftxt=strcat('@(x)',ftxt);
10 P=str2func(ftxt); %Carga axial
11 dir=input('Deslocamento extremidade direita (m) : ');
12 esq=input('Deslocamento extremidade esquerda (m) : ');
13 Fd=input('For a aplicada direita: ');
14
15
16
17 dtrue= @(x) (6*x^2 -40*x)/(-E*A); % colocar aqui a fun o exata!
18 Ucomforca(E,A,L,15,P,esq,Fd) % plot grafico com for a na extremidade com 15
    n s
19 hold on
20 Usemforca(E,A,L,4,P,dir,esq) % plot grafico sem for a na extremidade com 4
    n s
21 hold on
22 fplot(dtrue,[0,L],'g-') %plot da fun o exata
23
24 %%Nota para visualizar os graficos , ter a certeza que a linhas
25 %26 do Usemforca.m e 37 do Ucomforca.m n o estejam comentadas

```

Listing 4: errologSF.m

```

1 %Calcula o erro entre o dexato e os Df
2 function errologSF(n,dtrue,L,e,A,P,dir,esq)
3 format long
4 H=zeros(1,n-2);
5 E=zeros(1,n-2);
6 D=zeros(1,n-2);
7 nos=3;
8 for i=2:n-1
9 h=L/(nos-1);
10 H(i-1)=h; %guarda os h
11 x=linspace(0,L,nos);
12 Df=Usemforca(e,A,L,nos,P,dir,esq); % Df calculado pelo Usemforca.m
13 erro=-1;
14 Di=Df(1);
15 for j=2:(length(x)-1)
16     if erro<abs(dtrue(x(j))-Df(j)) %vai procurar o maior erro no
        deslocamento do Df
17         erro=abs(dtrue(x(j))-Df(j));
18     end
19 end
20 D(i-1)= Di;
21 E(i-1)=(erro); % o maior erro correspondente ao h

```

```

22     nos=nos+2; %atualizar os nos para atualizar o h
23 end
24 plot(log(H),log(E),'b-')
25 end

```

Listing 5: errologCF.m

```

1 %Calcula o erro entre o dexato e os Df
2 function errologCF(n,dtrue,L,e,A,P,Fd,esq)
3 format long
4 H=zeros(1,n-2);
5 E=zeros(1,n-2);
6 D=zeros(1,n-2);
7 nos=3;
8 for i=2:n-1
9     h=L/(nos-1);
10    H(i-1)=h;
11    x=linspace(0,L,nos);
12    Df=Ucomforca(e,A,L,nos,P,esq,Fd); %Df calculado a partir do Ucomforca.m
13    erro=-1;
14    Di=Df(1);
15    for j=2:(length(x)-1)
16        if erro<abs(dtrue(x(j))-Df(j))
17            erro=abs(dtrue(x(j))-Df(j));
18        end
19    end
20    D(i-1)= Di;
21    E(i-1)=(erro);
22    nos=nos+2;
23 end
24 plot(log(H),log(E),'b-')
25 end

```

Listing 6: ploterro.m

```

1 %Input do utilizador
2 E=input('M dulo de young (Pa)');
3 A=input(' rea da sec o da barra (m2) ');
4 L=input('Comprimento da barra (m)');
5 ftxt=input('Carga axial distribu da=? N/m','s');
6 ftxt=strcat('@(x)',ftxt);
7 P=str2func(ftxt);
8 dir=input('Deslocamento extermidade direita :');
9 esq=input('Deslocamento extermidade esquerda :');
10 n=input('N mero de itera es:');
11 Fd=input('For a aplicada direita: ');
12

```

```
13 |
14 | dtrue= @(x) (6*x^2 -40*x)/(-E*A); % colocar aqui a fun  o exata!
15 | errologSF(n,dtrue,L,E,A,P,dir,esq) % plot grafico sem for a
16 | hold on
17 | errologCF(n,dtrue,L,E,A,P,Fd,esq) % plot grafico com for a
18 |
19 | %%Nota para apenas visualizar o grafico do erro, ter a certeza que a linhas
20 | %26 do Usemforca.m e 37 do Ucomforca.m estejam comentadas
```