



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA
DISCIPLINA DE ANÁLISE MATRICIAL E MODELAGEM DE ESTRUTURAS

1º Trabalho de Avaliação de Aprendizagem

Thiago Ney Evaristo Rodrigues - 11502852
Professor: Dr. Marcelo Cavalcanti Rodrigues

João Pessoa, PB
Novembro de 2019

Questão 1

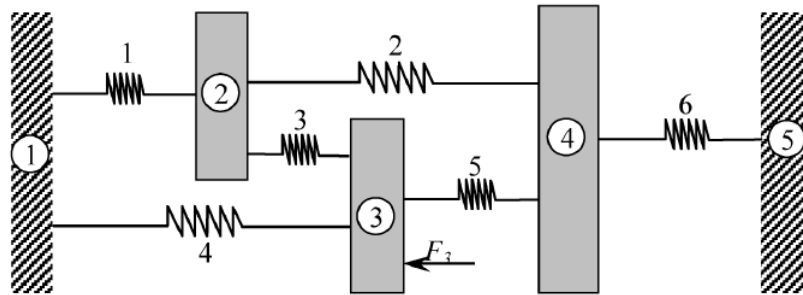


Figura 1: Sistema das questões 1 e 2

Matriz de Rigidez Global

Stiffness 1								
u1	u2	u3	u4	u5				
k1	-k1				u1			
-k1	k1				u2			
					u3			
					u4			
					u5			
Stiffness 2								
u1	u2	u3	u4	u5				
					u1			
	k2		-k2		u2			
					u3			
	-k2		k2		u4			
					u5			
Stiffness 3								
u1	u2	u3	u4	u5				
					u1			
	k3	-k3			u2			
	-k3	k3			u3			
					u4			
					u5			
Stiffness 4								
u1	u2	u3	u4	u5				
k4		-k4			u1			
					u2			
-k4		k4			u3			
					u4			
					u5			
Stiffness 5								
u1	u2	u3	u4	u5				
					u1			
					u2			
		k5	-k5		u3			
		-k5	k5		u4			
					u5			
Stiffness 6								
u1	u2	u3	u4	u5				
					u1			
					u2			
					u3			
			k6	-k6	u4			
			-k6	k6	u5			
Global Matrix								
			u1	u2	u3	u4	u5	
R21	R12	R13	F1					u1
	R23	R24	F2					u2
			F3					u3
	R42	R43	F4					u4
		R54	F5					u5

Figura 2: Obtenção da matriz de rigidez global da questão 1

Script (MATLAB)

```

1 clear
2 close all
3 clc
4
5 %% Inputs
6
7 F2 = 0; % Force [N]
8 F3 = -1000; % Force [N]
9 F4 = 0; % Force [N]
10 u1 = 0; % Displacement [mm]
11 u5 = 0; % Displacement [mm]
12 k1 = 500; % Spring Constant [N/mm]
13 k2 = 400; % Spring Constant [N/mm]
14 k3 = 600; % Spring Constant [N/mm]
15 k4 = 200; % Spring Constant [N/mm]
16 k5 = 400; % Spring Constant [N/mm]
17 k6 = 300; % Spring Constant [N/mm]
18 n = 5; % Degrees of Freedom
19
20 %% Characterization
21
22 A = [1 -1; -1 1];
23
24 % Stiffness
25
26 K1 = k1*A;
27 K2 = k2*A;
28 K3 = k3*A;
29 K4 = k4*A;
30 K5 = k5*A;
31 K6 = k6*A;
32
33 % Global Matrix Equation
34
35 K = zeros(n);
36 K(1:2,1:2) = K(1:2,1:2) + K1;
37 K(2:2:4,2:2:4) = K(2:2:4,2:2:4) + K2;
38 K(2:3,2:3) = K(2:3,2:3) + K3;
39 K(1:2:3,1:2:3) = K(1:2:3,1:2:3) + K4;
40 K(3:4,3:4) = K(3:4,3:4) + K5;
41 K(4:5,4:5) = K(4:5,4:5) + K6;
42
43 % Force Vector
44
45 syms F1 F5
46 F = [F1; F2; F3; F4; F5];

```

```

47
48 % Displacement Vectors
49
50 syms u2 u3 u4
51 U = [u1; u2; u3; u4; u5];
52
53 %% Calculations
54
55 AN = solve(F-K*U);
56
57 F = double([AN.F1; F2; F3; F4; AN.F5]);
58 U = double([u1; AN.u2; AN.u3; AN.u4; u5]);
59
60 f1 = k1*(U(1) - U(2));
61 f2 = k2*(U(2) - U(4));
62 f3 = k3*(U(2) - U(3));
63 f4 = k4*(U(1) - U(3));
64 f5 = k5*(U(3) - U(4));
65 f6 = k6*(U(4) - U(5));
66
67 f = [f1; f2; f3; f4; f5; f6];
68
69 clear A AN f1 f2 f3 f4 f5 f6 F1 F2 F3 F4 F5 K1 K2 K3 K4 K5 K6 n u1 u2 ...
    u3 u4 u5

```

Resultados

Obteve-se, a partir do script apresentado, os seguintes resultados:

$$[F] = \begin{bmatrix} R_1 \\ 0 \\ F_3 \\ 0 \\ R_5 \end{bmatrix} = \begin{bmatrix} 737.5 \\ 0 \\ -1000 \\ 0 \\ 262.5 \end{bmatrix} N$$

$$[K] = \begin{bmatrix} 700 & -500 & -200 & 0 & 0 \\ -500 & 1500 & -600 & -400 & 0 \\ -200 & -600 & 1200 & -400 & 0 \\ 0 & -400 & -400 & 1100 & -300 \\ 0 & 0 & 0 & -300 & 300 \end{bmatrix} N/mm$$

$$[U] = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} 0 \\ -0.8541667 \\ -1.55208333 \\ -0.875 \\ 0 \end{bmatrix} mm$$

$$[f] = \begin{bmatrix} 427.08333 \\ 8.333 \\ 418.75 \\ 310.41667 \\ -270.8333 \\ -262.5 \end{bmatrix} N$$

sendo $[F]$ o vetor força (carregamento e reações nos nós), $[K]$ a matriz de rigidez global, $[U]$ o vetor deslocamento e $[f]$ o vetor força (tração/compressão nos elementos de mola).

Questão 2

Script (MATLAB)

```
1 clear
2 close all
3 clc
4
5 %% Inputs
6
7 F3 = -1000; % Force [N]
8 F4 = 0; % Force [N]
9 u1 = 0; % Displacement [mm]
10 u2 = 0; % Displacement [mm]
11 u5 = 0; % Displacement [mm]
12 k1 = 500; % Spring Constant [N/mm]
13 k2 = 400; % Spring Constant [N/mm]
14 k3 = 600; % Spring Constant [N/mm]
15 k4 = 200; % Spring Constant [N/mm]
16 k5 = 400; % Spring Constant [N/mm]
17 k6 = 300; % Spring Constant [N/mm]
18 n = 5; % Degrees of Freedom
19
20 %% Characterization
21
22 A = [1 -1; -1 1];
23
24 % Stiffness
25
26 K1 = k1*A;
27 K2 = k2*A;
28 K3 = k3*A;
29 K4 = k4*A;
30 K5 = k5*A;
31 K6 = k6*A;
32
33 % Global Matrix Equation
34
35 K = zeros(n);
36 K(1:2,1:2) = K(1:2,1:2) + K1;
37 K(2:2:4,2:2:4) = K(2:2:4,2:2:4) + K2;
38 K(2:3,2:3) = K(2:3,2:3) + K3;
39 K(1:2:3,1:2:3) = K(1:2:3,1:2:3) + K4;
40 K(3:4,3:4) = K(3:4,3:4) + K5;
41 K(4:5,4:5) = K(4:5,4:5) + K6;
```

```

42
43 % Force Vector
44
45 syms F1 F2 F5
46 F = [F1; F2; F3; F4; F5];
47
48 % Displacement Vectors
49
50 syms u3 u4
51 U = [u1; u2; u3; u4; u5];
52
53 %% Calculations
54
55 AN = solve(F-K*U);
56
57 F = double([AN.F1; AN.F2; F3; F4; AN.F5]);
58 U = double([u1; u2; AN.u3; AN.u4; u5]);
59
60 clear A AN F1 F2 F3 F4 F5 K1 K2 K3 K4 K5 K6 n u1 u2 u3 u4 u5

```

Resultados

Obteve-se, a partir do script apresentado, os seguintes resultados:

$$[F] = \begin{bmatrix} R_1 \\ F_2 \\ F_3 \\ 0 \\ R_5 \end{bmatrix} = \begin{bmatrix} 189.65517 \\ 706.89655 \\ -1000 \\ 0 \\ 103.44828 \end{bmatrix} N$$

$$[U] = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -0.94828 \\ -0.34483 \\ 0 \end{bmatrix} mm$$

sendo $[F]$ o novo vetor força (carregamento e reações nos nós) e $[U]$ o novo vetor deslocamento.

Script (MATLAB)


```

1 clear
2 close all
3 clc
4
5 %% Inputs
6
7 syms k1 k2 k3 k4 F1 F2 F3 F4 F5 u1 u2 u3
8
9 u4 = 0; % Displacement [mm]
10 u5 = 0; % Displacement [mm]
11
12 %% Characterization
13
14 % Stiffness
15
16 K = [k1+k3+k4 -k4 -k3 -k1 0; ...
17 -k4 k4 0 0 0; ...
18 -k3 0 k2+k3 0 -k2; ...
19 -k1 0 0 k1 0; ...
20 0 0 k2 0 -k2];
21
22 % Force Vector
23
24 F = [F1; F2; F3; F4; F5];
25
26 % Displacement Vectors
27
28 U = [u1; u2; u3; u4; u5];
29
30 clear F1 F2 F3 F4 F5 u1 u2 u3 u4 u5

```

Resultados

Obteve-se, a partir do script apresentado, os seguintes resultados:

$$[F] = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ R_4 \\ R_5 \end{bmatrix} N$$

$$[K] = \begin{bmatrix} k1 + k3 + k4 & -k4 & -k3 & -k1 & 0 \\ -k4 & k4 & 0 & 0 & 0 \\ -k3 & 0 & k2 + k3 & 0 & -k2 \\ -k1 & 0 & 0 & k1 & 0 \\ 0 & 0 & -k2 & 0 & k2 \end{bmatrix} N/mm$$

$$[U] = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ 0 \\ 0 \end{bmatrix} mm$$

sendo $[F]$ o vetor força (carregamento e reações nos nós), $[K]$ a matriz de rigidez global e $[U]$ o vetor deslocamento.

Questão 4

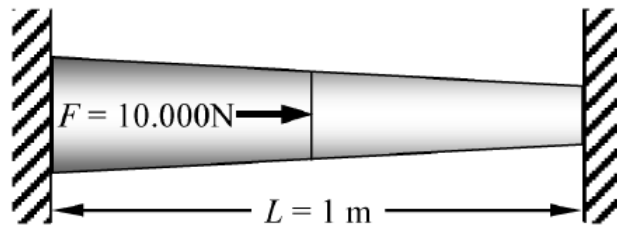


Figura 5: Sistema da questão 4

Script (MATLAB)

```

1 clear
2 close all
3 clc
4
5 %% Inputs
6
7 F2 = 0; % Force [N]
8 F3 = 10000; % Force [N]
9 F4 = 0; % Force [N]
10 u1 = 0; % Displacement [m]
11 u2 = 0; % Displacement [m]
12 L = 1; % Length [m]
13 E = 100*10^6; % Young's Modulus [Pa]
14 n = 5; % Degrees of Freedom
15
16 %% Characterization
17
18 M = [1 -1; -1 1];
19
20 % Stiffness
21
22 x = linspace(0,L,n);
23 r = 0.05 - 0.04.*(x); % Radius [m]
24 A = @ (r) pi*r(1:end-1).*r(2:end); % Area [m^2]
25 A = A(r);
26
27 k = E.*A./(diff(x));
28
29 K1 = k(1)*M;
30 K2 = k(2)*M;

```

```

31 K3 = k(3)*M;
32 K4 = k(4)*M;
33
34 % Global Matrix Equation
35
36 K = zeros(n);
37 K(1:2,1:2) = K(1:2,1:2) + K1;
38 K(2:3,2:3) = K(2:3,2:3) + K2;
39 K(3:4,3:4) = K(3:4,3:4) + K3;
40 K(4:5,4:5) = K(4:5,4:5) + K4;
41
42 % Force Vector
43
44 syms F1 F5
45 F = [F1; F2; F3; F4; F5];
46
47 % Displacement Vectors
48
49 syms u3 u4 u5
50 U = [u1; u3; u4; u5; u2];
51
52 %% Calculations
53
54 AN = solve(F-K*U);
55
56 F = double([AN.F1; F2; F3; F4; AN.F5]);
57 U = double([u1; AN.u3; AN.u4; AN.u5; u2]);
58
59 f = k'.*diff(U);
60
61 clear AN F1 F2 F3 F4 F5 K1 K2 K3 K4 M n u1 u2 u3 u4 u5

```

Resultados

Diante da característica do sistema, o mesmo foi discretizado a fim de simplificar a análise, obtendo-se:

$$[x] = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.25 \\ 0.5 \\ 0.75 \\ 1 \end{bmatrix} m$$

$$[r] = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{bmatrix} = \begin{bmatrix} 0.05 \\ 0.04 \\ 0.03 \\ 0.02 \\ 0.01 \end{bmatrix} m$$

$$[A] = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix} = \begin{bmatrix} 0.0062832 \\ 0.0037699 \\ 0.001885 \\ 0.00062839 \end{bmatrix} m^2$$

$$[k] = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{bmatrix} = 10^6 \times \begin{bmatrix} 2.5133 \\ 1.5080 \\ 0.7540 \\ 0.2513 \end{bmatrix} N/m$$

sendo $[X]$ o vetor comprimento (distância entre origem e nós), $[r]$ o vetor raio (raio da seção transversal de cada nó), $[A]$ o vetor área (área da seção transversal de cada nó) e $[k]$ o vetor rigidez de cada elemento.

Obteve-se, a partir do script apresentado, os seguintes resultados:

$$[F] = \begin{bmatrix} R_1 \\ F_2 \\ F_3 \\ F_4 \\ R_5 \end{bmatrix} = 10^4 \times \begin{bmatrix} -0.8333 \\ 0 \\ 1.0000 \\ 0 \\ -0.1667 \end{bmatrix} N$$

$$[K] = 10^6 \times \begin{bmatrix} 2.51327 & -2.51327 & 0 & 0 & 0 \\ -2.51327 & 4.02124 & -1.50796 & 0 & 0 \\ 0 & -1.50796 & 2.26195 & -0.75398 & 0 \\ 0 & 0 & -0.75398 & 1.00531 & -0.25133 \\ 0 & 0 & 0 & -0.25133 & 0.25133 \end{bmatrix} N/m$$

$$[U] = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.0033 \\ 0.0088 \\ 0.0066 \\ 0 \end{bmatrix} m$$

$$[f] = 10^3 \times \begin{bmatrix} 8.3333 \\ 8.3333 \\ -1.6667 \\ -1.6667 \end{bmatrix} N$$

sendo $[F]$ o vetor força (carregamento e reações nos nós), $[K]$ a matriz de rigidez global, $[U]$ o vetor deslocamento e $[f]$ o vetor força (tração/compressão nos elementos de mola).

Questão 5

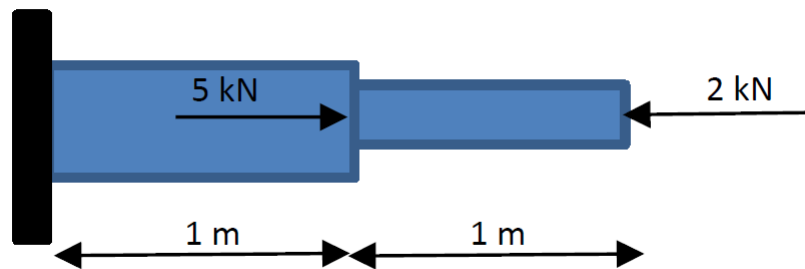


Figura 6: Sistema da questão 5

Script (MATLAB)

```

1 clear
2 close all
3 clc
4
5 %% Inputs
6
7 F2 = 5000; % Force [N]
8 F3 = -2000; % Force [N]
9 u1 = 0; % Displacement [m]
10 L1 = 1; % Length [m]
11 L2 = 1; % Length [m]
12 E = 100*10^9; % Young's Modulus [Pa]
13 A1 = 1*10^(-4); % Area [m^2]
14 A2 = 2*10^(-4); % Area [m^2]
15 n = 3; % Degrees of Freedom
16
17 %% Characterization
18
19 M = [1 -1; -1 1];
20
21 % Stiffness
22
23 A = [A1; A2];
24 x = cumsum([0; L1; L2]);
25
26 k = E.*A./(diff(x));
27
28 K1 = k(1)*M;
```

```

29 K2 = k(2)*M;
30
31 % Global Matrix Equation
32
33 K = zeros(n);
34 K(1:2,1:2) = K(1:2,1:2) + K1;
35 K(2:3,2:3) = K(2:3,2:3) + K2;
36
37 % Force Vector
38
39 syms F1
40 F = [F1; F2; F3];
41
42 % Displacement Vectors
43
44 syms u2 u3
45 U = [u1; u2; u3];
46
47 %% Calculations
48
49 AN = solve(F-K*U);
50
51 F = double([AN.F1; F2; F3]);
52 U = double([u1; AN.u2; AN.u3]);
53
54 f = k.*diff(U);
55
56 clear AN A1 A2 F1 F2 F3 L1 L2 M n u1 u2 u3

```

Resultados

Obteve-se, a partir do script apresentado, os seguintes resultados:

Item (a)

$$[K_1] = 10^7 \times \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} N/m$$

$$[K_2] = 10^7 \times \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix} N/m$$

sendo $[K_1]$ a matriz de rigidez do elemento de mola 1 e $[K_2]$ a matriz de rigidez do elemento de mola 2.

Item (b)

$$\begin{bmatrix} R1 \\ F2 \\ F3 \end{bmatrix} = \begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} \begin{bmatrix} u1 \\ u2 \\ u3 \end{bmatrix}$$

$$[F] = \begin{bmatrix} -3000 \\ 5000 \\ -2000 \end{bmatrix} N$$

$$[K] = 10^7 \times \begin{bmatrix} 1 & -1 & 0 \\ -1 & 3 & -2 \\ 0 & -2 & 2 \end{bmatrix} N/m$$

$$[U] = 10^{-3} \times \begin{bmatrix} 0 \\ 0.3000 \\ 0.2000 \end{bmatrix} m$$

sendo $[F]$ o vetor força (carregamento e reações nos nós), $[K]$ a matriz de rigidez global e $[U]$ o vetor deslocamento.

Item (c)

$$[U] = 10^{-3} \times \begin{bmatrix} 0 \\ 0.3000 \\ 0.2000 \end{bmatrix} m$$

sendo $[U]$ o vetor deslocamento nodal.

Item (d)

$$[f] = \begin{bmatrix} 3000 \\ -2000 \end{bmatrix} N$$

sendo $[f]$ o vetor força (tração/compressão nos elementos de mola).

Questão 6

Script (MATLAB)

```

1 clear
2 close all
3 clc
4
5 %% Inputs
6
7 F2 = 0; % Force [N]
8 F3 = 5000; % Force [N]
9 F4 = 0; % Force [N]
10 F5 = -2000; % Force [N]
11 u1 = 0; % Displacement [m]
12 L1 = 1; % Lenght [m]
13 L2 = 1; % Lenght [m]
14 E = 100*10^9; % Young's Modulus [Pa]
15 A1 = 1*10^(-4); % Area [m^2]
16 A2 = 2*10^(-4); % Area [m^2]
17 n = 5; % Degrees of Freedom
18
19 %% Characterization
20
21 M = [1 -1; -1 1];
22
23 % Stiffness
24
25 A = [A1; A1; A2; A2];
26 x = cumsum([0; L1/2; L1/2; L2/2; L2/2]);
27
28 k = E.*A./(diff(x));
29
30 K1 = k(1)*M;
31 K2 = k(2)*M;
32 K3 = k(3)*M;
33 K4 = k(4)*M;
34
35 % Global Matrix Equation
36
37 K = zeros(n);
38 K(1:2,1:2) = K(1:2,1:2) + K1;
39 K(2:3,2:3) = K(2:3,2:3) + K2;
40 K(3:4,3:4) = K(3:4,3:4) + K3;
41 K(4:5,4:5) = K(4:5,4:5) + K4;

```

```

42
43 % Force Vector
44
45 syms F1
46 F = [F1; F2; F3; F4; F5];
47
48 % Displacement Vectors
49
50 syms u2 u3 u4 u5
51 U = [u1; u2; u3; u4; u5];
52
53 %% Calculations
54
55 AN = solve(F-K*U);
56
57 F = double([AN.F1; F2; F3; F4; F5]);
58 U = double([u1; AN.u2; AN.u3; AN.u4; AN.u5]);
59
60 f = k.*diff(U);
61
62 clear AN A1 A2 F1 F2 F3 F4 F5 L1 L2 M n u1 u2 u3 u4 u5

```

Resultados

Obteve-se, a partir do script apresentado, os seguintes resultados:

Item (a)

$$[K_1] = 10^7 \times \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix} N/m$$

$$[K_2] = 10^7 \times \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix} N/m$$

$$[K_3] = 10^7 \times \begin{bmatrix} 4 & -4 \\ -4 & 4 \end{bmatrix} N/m$$

$$[K_4] = 10^7 \times \begin{bmatrix} 4 & -4 \\ -4 & 4 \end{bmatrix} N/m$$

sendo $[K_i]$ a matriz de rigidez do elemento de mola i .

Item (b)

$$\begin{bmatrix} R1 \\ F2 \\ F3 \\ F4 \\ F5 \end{bmatrix} = \begin{bmatrix} k_1 & -k_1 & 0 & 0 & 0 \\ -k_1 & k_1 + k_2 & -k_2 & 0 & 0 \\ 0 & -k_2 & k_2 + k_3 & -k_3 & 0 \\ 0 & 0 & -k_3 & k_3 + k_4 & -k_4 \\ 0 & 0 & 0 & -k_4 & k_4 \end{bmatrix} \begin{bmatrix} u1 \\ u2 \\ u3 \\ u4 \\ u5 \end{bmatrix}$$

$$[F] = \begin{bmatrix} -3000 \\ 0 \\ 5000 \\ 0 \\ -2000 \end{bmatrix} N$$

$$[K] = 10^7 \times \begin{bmatrix} 2 & -2 & 0 & 0 & 0 \\ -2 & 4 & -2 & 0 & 0 \\ 0 & -2 & 6 & -4 & 0 \\ 0 & 0 & -4 & 8 & -4 \\ 0 & 0 & 0 & -4 & 4 \end{bmatrix} N/m$$

$$[U] = 10^{-3} \times \begin{bmatrix} 0 \\ 0.15 \\ 0.3 \\ 0.25 \\ 0.2 \end{bmatrix} m$$

sendo $[F]$ o vetor força (carregamento e reações nos nós), $[K]$ a matriz de rigidez global e $[U]$ o vetor deslocamento.

Item (c)

$$[U] = 10^{-3} \times \begin{bmatrix} 0 \\ 0.15 \\ 0.3 \\ 0.25 \\ 0.2 \end{bmatrix} m$$

sendo $[U]$ o vetor deslocamento nodal.

Item (d)

$$[f] = \begin{bmatrix} 3000 \\ 3000 \\ -2000 \\ -2000 \end{bmatrix} N$$

sendo $[f]$ o vetor força (tração/compressão nos elementos de mola).

Questão 7

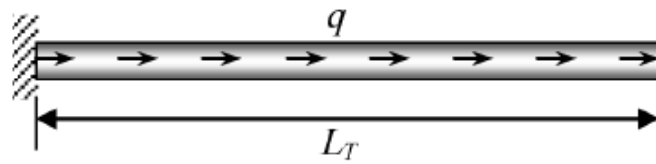


Figura 7: Sistema da questão 7

Script (MATLAB)

Item (a)

```

1 clear
2 close all
3 clc
4
5 %% Inputs
6
7 u1 = 0; % Boundary Condition
8 du15dx = 1; % Boundary Condition
9 p = 1000; % Load [N/m]
10 l = 1.5; % Length [m]
11 r = 0.1; % Ray [m]
12 E = 207*10^9; % Young's Modulus [Pa]
13 NE = 3; % Number of Elements
14
15 %% FEM - Direct Method
16
17 ND = NE + 1;
18 M = [1 -1; -1 1];
19
20 % Stiffness
21
22 A = pi*r^2;
23 X = linspace(0,l,ND);
24 L = diff(X);
25
26 k = E*A./L;
27
28 K1 = k(1)*M;
29 K2 = k(2)*M;
30 K3 = k(3)*M;

```

```

31
32 % Global Matrix Equation
33
34 K = zeros(ND);
35 K(1:2,1:2) = K(1:2,1:2) + K1;
36 K(2:3,2:3) = K(2:3,2:3) + K2;
37 K(3:4,3:4) = K(3:4,3:4) + K3;
38
39 % Force Vector
40
41 syms F1 F2 F3 F4
42
43 F = [F1; F2; F3; F4];
44 aux = p*X;
45 F(2:4) = aux(2:4);
46
47 % Displacement Vectors
48
49 syms u2 u3 u4
50 U = [u1; u2; u3; u4];
51
52 % Calculations
53
54 AN = solve(F-K*U);
55
56 F(1) = AN.F1;
57 U(2:4) = [AN.u2; AN.u3; AN.u4];
58
59 F = double(F);
60 U = double(U);
61
62 % Plots
63
64 figure
65 plot(X,F/A,'-ob','MarkerFaceColor','b')
66
67 figure
68 plot(X,U,'-ob','MarkerFaceColor','b')
69
70 %% Garlekin's Method
71
72 % Characterization
73
74 syms x F1 u2 u3 u4 du0dx
75 ND = NE + 1;
76
77 X = linspace(0,1,ND);

```

```

78 L = diff(X);
79
80 for i = 1:ND
81
82     if i == 1
83
84         phi(i,1) = (X(i+1) - x)/L(i);
85         phi(i,2) = 0;
86         phi(i,3) = 0;
87
88     elseif i == ND
89
90         phi(i,1) = 0;
91         phi(i,2) = 0;
92         phi(i,3) = (x - X(i-1))/L(i-1);
93
94     elseif i == 2
95
96         phi(i,1) = (x - X(i-1))/L(i-1);
97         phi(i,2) = (X(i+1) - x)/L(i);
98         phi(i,3) = 0;
99
100    else
101
102        phi(i,1) = 0;
103        phi(i,2) = (x - X(i-1))/L(i-1);
104        phi(i,3) = (X(i+1) - x)/L(i);
105
106    end
107
108 end
109
110 % Force
111
112 P = p*X;
113 P = [P(2:4); P(2:4); P(2:4); P(2:4)];
114 f = P.*phi;
115 F = int(f(:,1),x,0,0.5) + int(f(:,2),x,0.5,1) + int(f(:,3),x,1,1.5) + ...
        du15dx*subs(phi(:,2),x,1.5) - du0dx*subs(phi(:,1),x,0);
116 F(1) = F1;
117
118 % Stiffness
119
120 M = [1 -1; -1 1];
121
122 A = pi*r^2;
123 k = E*A./L;

```



```

124
125 K1 = k(1)*M;
126 K2 = k(2)*M;
127 K3 = k(3)*M;
128
129 K = zeros(ND);
130 K(1:2,1:2) = K(1:2,1:2) + K1;
131 K(2:3,2:3) = K(2:3,2:3) + K2;
132 K(3:4,3:4) = K(3:4,3:4) + K3;
133
134 % Displacement
135
136 U = [u1; u2; u3; u4];
137
138 % Calculations
139
140 AN = solve(F - K*U);
141 F(1) = AN.F1;
142 U(2:4) = [AN.u2; AN.u3; AN.u4];
143
144 % Plot
145
146 figure(1)
147 hold on
148 plot(X,F/A,'-or','MarkerFaceColor','r')
149 legend('Método Direto', 'Método de Garlekin','Location','southeast')
150 xlabel('Comprimento [m]')
151 ylabel('Tensão [Pa]')
152 grid minor
153
154 figure(2)
155 hold on
156 plot(X,U,'-or','MarkerFaceColor','r')
157 legend('Método Direto', 'Método de Garlekin','Location','southeast')
158 xlabel('Comprimento Inicial [m]')
159 ylabel('Deslocamento [m]')
160 grid minor

```

Item (b)

```

1 clear
2 close all
3 clc
4
5 %% Inputs

```

```

6
7 u1 = 0; % Boundary Condition
8 du15dx = 1; % Boundary Condition
9 p = 1000; % Load [N/m]
10 l = 1.5; % Length [m]
11 r = 0.1; % Ray [m]
12 E = 207*10^9; % Young's Modulus [Pa]
13
14 %% Garlekin's Method (4 Elements)
15
16 % Characterization
17
18 syms x F1 u2 u3 u4 u5 du0dx
19 NE = 4; % Number of Elements
20 ND = NE + 1;
21
22 X = linspace(0,l,ND);
23 L = diff(X);
24
25 for i = 1:ND
26
27     if i == 1
28
29         phi(i,1) = (X(i+1) - x)/L(i);
30         phi(i,2) = 0;
31         phi(i,3) = 0;
32         phi(i,4) = 0;
33
34     elseif i == 2
35
36         phi(i,1) = (x - X(i-1))/L(i-1);
37         phi(i,2) = (X(i+1) - x)/L(i);
38         phi(i,3) = 0;
39         phi(i,4) = 0;
40
41     elseif i == 3
42
43         phi(i,1) = 0;
44         phi(i,2) = (x - X(i-1))/L(i-1);
45         phi(i,3) = (X(i+1) - x)/L(i);
46         phi(i,4) = 0;
47
48     elseif i == 4
49
50         phi(i,1) = 0;
51         phi(i,2) = 0;
52         phi(i,3) = (x - X(i-1))/L(i-1);

```

```

53 phi(i,4) = (X(i+1) - x)/L(i);
54
55 else
56
57 phi(i,1) = 0;
58 phi(i,2) = 0;
59 phi(i,3) = 0;
60 phi(i,4) = (x - X(i-1))/L(i-1);
61
62 end
63
64 end
65
66 % Force
67
68 P = p*X;
69 P = [P(2:5); P(2:5); P(2:5); P(2:5); P(2:5)];
70 f = P.*phi;
71 F = int(f(:,1),x,X(1),X(2)) + int(f(:,2),x,X(2),X(3)) + ...
      int(f(:,3),x,X(3),X(4)) ...
72 + int(f(:,4),x,X(4),X(5)) + du15dx*subs(phi(:,2),x,1.5) - ...
      du0dx*subs(phi(:,1),x,0);
73 F(1) = F1;
74
75 % Stiffness
76
77 M = [1 -1; -1 1];
78
79 A = pi*r^2;
80 k = E*A./L;
81
82 K1 = k(1)*M;
83 K2 = k(2)*M;
84 K3 = k(3)*M;
85 K4 = k(4)*M;
86
87 K = zeros(ND);
88 K(1:2,1:2) = K(1:2,1:2) + K1;
89 K(2:3,2:3) = K(2:3,2:3) + K2;
90 K(3:4,3:4) = K(3:4,3:4) + K3;
91 K(4:5,4:5) = K(4:5,4:5) + K4;
92
93 % Displacement
94
95 U = [u1; u2; u3; u4; u5];
96
97 % Calculations

```

```

98
99 AN = solve(F - K*U);
100 F(1) = AN.F1;
101 U(2:5) = [AN.u2; AN.u3; AN.u4; AN.u5];
102
103 % Plot
104
105 figure
106 plot(X,F/A, '-ob', 'MarkerFaceColor', 'b')
107
108 figure
109 plot(X,U, '-ob', 'MarkerFaceColor', 'b')
110
111 %% Garlekin's Method (6 Elements)
112
113 % Characterization
114
115 syms x F1 u2 u3 u4 u5 u6 u7 du0dx
116 NE = 6; % Number of Elements
117 ND = NE + 1;
118
119 X = linspace(0,1,ND);
120 L = diff(X);
121
122 for i = 1:ND
123
124     if i == 1
125
126         phi(i,1) = (X(i+1) - x)/L(i);
127         phi(i,2) = 0;
128         phi(i,3) = 0;
129         phi(i,4) = 0;
130         phi(i,5) = 0;
131         phi(i,6) = 0;
132
133     elseif i == 2
134
135         phi(i,1) = (x - X(i-1))/L(i-1);
136         phi(i,2) = (X(i+1) - x)/L(i);
137         phi(i,3) = 0;
138         phi(i,4) = 0;
139         phi(i,5) = 0;
140         phi(i,6) = 0;
141
142     elseif i == 3
143
144         phi(i,1) = 0;

```

```

145 phi(i,2) = (x - X(i-1))/L(i-1);
146 phi(i,3) = (X(i+1) - x)/L(i);
147 phi(i,4) = 0;
148 phi(i,5) = 0;
149 phi(i,6) = 0;
150
151 elseif i == 4
152
153 phi(i,1) = 0;
154 phi(i,2) = 0;
155 phi(i,3) = (x - X(i-1))/L(i-1);
156 phi(i,4) = (X(i+1) - x)/L(i);
157 phi(i,5) = 0;
158 phi(i,6) = 0;
159
160 elseif i == 5
161
162 phi(i,1) = 0;
163 phi(i,2) = 0;
164 phi(i,3) = 0;
165 phi(i,4) = (x - X(i-1))/L(i-1);
166 phi(i,5) = (X(i+1) - x)/L(i);
167 phi(i,6) = 0;
168
169 elseif i == 6
170
171 phi(i,1) = 0;
172 phi(i,2) = 0;
173 phi(i,3) = 0;
174 phi(i,4) = 0;
175 phi(i,5) = (x - X(i-1))/L(i-1);
176 phi(i,6) = (X(i+1) - x)/L(i);
177
178 else
179
180 phi(i,1) = 0;
181 phi(i,2) = 0;
182 phi(i,3) = 0;
183 phi(i,4) = 0;
184 phi(i,5) = 0;
185 phi(i,6) = (x - X(i-1))/L(i-1);
186
187 end
188
189 end
190
191 % Force

```

```

192
193 P = p*X;
194 P = [P(2:7); P(2:7); P(2:7); P(2:7); P(2:7); P(2:7); P(2:7)];
195 f = P.*phi;
196 F = int(f(:,1),x,X(1),X(2)) + int(f(:,2),x,X(2),X(3)) + ...
      int(f(:,3),x,X(3),X(4)) ...
197 + int(f(:,4),x,X(4),X(5)) + int(f(:,5),x,X(5),X(6)) + ...
      int(f(:,6),x,X(6),X(7)) ...
198 + du15dx*subs(phi(:,2),x,1.5) - du0dx*subs(phi(:,1),x,0);
199 F(1) = F1;
200
201 % Stiffness
202
203 M = [1 -1; -1 1];
204
205 A = pi*r^2;
206 k = E*A./L;
207
208 K1 = k(1)*M;
209 K2 = k(2)*M;
210 K3 = k(3)*M;
211 K4 = k(4)*M;
212 K5 = k(5)*M;
213 K6 = k(6)*M;
214
215 K = zeros(ND);
216 K(1:2,1:2) = K(1:2,1:2) + K1;
217 K(2:3,2:3) = K(2:3,2:3) + K2;
218 K(3:4,3:4) = K(3:4,3:4) + K3;
219 K(4:5,4:5) = K(4:5,4:5) + K4;
220 K(5:6,5:6) = K(5:6,5:6) + K5;
221 K(6:7,6:7) = K(6:7,6:7) + K6;
222
223 % Displacement
224
225 U = [u1; u2; u3; u4; u5; u6; u7];
226
227 % Calculations
228
229 AN = solve(F - K*U);
230 F(1) = AN.F1;
231 U(2:7) = [AN.u2; AN.u3; AN.u4; AN.u5; AN.u6; AN.u7];
232
233 % Plot
234
235 figure(1)
236 hold on

```

```

237 plot(X,F/A, '-or', 'MarkerFaceColor', 'r')
238
239 figure(2)
240 hold on
241 plot(X,U, '-or', 'MarkerFaceColor', 'r')
242
243 %% Garlekin's Method (8 Elements)
244
245 % Characterization
246
247 syms x F1 u2 u3 u4 u5 u6 u7 u8 u9 du0dx
248 NE = 8; % Number of Elements
249 ND = NE + 1;
250
251 X = linspace(0,1,ND);
252 L = diff(X);
253
254 for i = 1:ND
255
256     if i == 1
257
258         phi(i,1) = (X(i+1) - x)/L(i);
259         phi(i,2) = 0;
260         phi(i,3) = 0;
261         phi(i,4) = 0;
262         phi(i,5) = 0;
263         phi(i,6) = 0;
264         phi(i,7) = 0;
265         phi(i,8) = 0;
266
267     elseif i == 2
268
269         phi(i,1) = (x - X(i-1))/L(i-1);
270         phi(i,2) = (X(i+1) - x)/L(i);
271         phi(i,3) = 0;
272         phi(i,4) = 0;
273         phi(i,5) = 0;
274         phi(i,6) = 0;
275         phi(i,7) = 0;
276         phi(i,8) = 0;
277
278     elseif i == 3
279
280         phi(i,1) = 0;
281         phi(i,2) = (x - X(i-1))/L(i-1);
282         phi(i,3) = (X(i+1) - x)/L(i);
283         phi(i,4) = 0;

```

```

284 phi(i,5) = 0;
285 phi(i,6) = 0;
286 phi(i,7) = 0;
287 phi(i,8) = 0;
288
289 elseif i == 4
290
291 phi(i,1) = 0;
292 phi(i,2) = 0;
293 phi(i,3) = (x - X(i-1))/L(i-1);
294 phi(i,4) = (X(i+1) - x)/L(i);
295 phi(i,5) = 0;
296 phi(i,6) = 0;
297 phi(i,7) = 0;
298 phi(i,8) = 0;
299
300 elseif i == 5
301
302 phi(i,1) = 0;
303 phi(i,2) = 0;
304 phi(i,3) = 0;
305 phi(i,4) = (x - X(i-1))/L(i-1);
306 phi(i,5) = (X(i+1) - x)/L(i);
307 phi(i,6) = 0;
308 phi(i,7) = 0;
309 phi(i,8) = 0;
310
311 elseif i == 6
312
313 phi(i,1) = 0;
314 phi(i,2) = 0;
315 phi(i,3) = 0;
316 phi(i,4) = 0;
317 phi(i,5) = (x - X(i-1))/L(i-1);
318 phi(i,6) = (X(i+1) - x)/L(i);
319 phi(i,7) = 0;
320 phi(i,8) = 0;
321
322 elseif i == 7
323
324 phi(i,1) = 0;
325 phi(i,2) = 0;
326 phi(i,3) = 0;
327 phi(i,4) = 0;
328 phi(i,5) = 0;
329 phi(i,6) = (x - X(i-1))/L(i-1);
330 phi(i,7) = (X(i+1) - x)/L(i);

```



```

331 phi(i,8) = 0;
332
333 elseif i == 8
334
335 phi(i,1) = 0;
336 phi(i,2) = 0;
337 phi(i,3) = 0;
338 phi(i,4) = 0;
339 phi(i,5) = 0;
340 phi(i,6) = 0;
341 phi(i,7) = (x - X(i-1))/L(i-1);
342 phi(i,8) = (X(i+1) - x)/L(i);
343
344 else
345
346 phi(i,1) = 0;
347 phi(i,2) = 0;
348 phi(i,3) = 0;
349 phi(i,4) = 0;
350 phi(i,5) = 0;
351 phi(i,6) = 0;
352 phi(i,7) = 0;
353 phi(i,8) = (x - X(i-1))/L(i-1);
354
355 end
356
357 end
358
359 % Force
360
361 P = p*X;
362 P = [P(2:9); P(2:9); P(2:9); P(2:9); P(2:9); P(2:9); P(2:9); P(2:9); ...
      P(2:9)];
363 f = P.*phi;
364 F = int(f(:,1),x,X(1),X(2)) + int(f(:,2),x,X(2),X(3)) + ...
      int(f(:,3),x,X(3),X(4)) ...
365 + int(f(:,4),x,X(4),X(5)) + int(f(:,5),x,X(5),X(6)) + ...
      int(f(:,6),x,X(6),X(7)) ...
366 + int(f(:,7),x,X(7),X(8)) + int(f(:,8),x,X(8),X(9)) + ...
      du15dx*subs(phi(:,2),x,1.5) ...
367 - du0dx*subs(phi(:,1),x,0);
368 F(1) = F1;
369
370 % Stiffness
371
372 M = [1 -1; -1 1];
373

```

```

374 A = pi*r^2;
375 k = E*A./L;
376
377 K1 = k(1)*M;
378 K2 = k(2)*M;
379 K3 = k(3)*M;
380 K4 = k(4)*M;
381 K5 = k(5)*M;
382 K6 = k(6)*M;
383 K7 = k(7)*M;
384 K8 = k(8)*M;
385
386 K = zeros (ND) ;
387 K(1:2,1:2) = K(1:2,1:2) + K1;
388 K(2:3,2:3) = K(2:3,2:3) + K2;
389 K(3:4,3:4) = K(3:4,3:4) + K3;
390 K(4:5,4:5) = K(4:5,4:5) + K4;
391 K(5:6,5:6) = K(5:6,5:6) + K5;
392 K(6:7,6:7) = K(6:7,6:7) + K6;
393 K(7:8,7:8) = K(7:8,7:8) + K7;
394 K(8:9,8:9) = K(8:9,8:9) + K8;
395
396 % Displacement
397
398 U = [u1; u2; u3; u4; u5; u6; u7; u8; u9];
399
400 % Calculations
401
402 AN = solve(F - K*U);
403 F(1) = AN.F1;
404 U(2:9) = [AN.u2; AN.u3; AN.u4; AN.u5; AN.u6; AN.u7; AN.u8; AN.u9];
405
406 % Plot
407
408 figure(1)
409 hold on
410 plot(X,F/A,'-og','MarkerFaceColor','g')
411
412 figure(2)
413 hold on
414 plot(X,U,'-og','MarkerFaceColor','g')
415
416 %% Garlekin's Method (10 Elements)
417
418 % Characterization
419
420 syms x F1 u2 u3 u4 u5 u6 u7 u8 u9 u10 u11 du0dx

```

```

421 NE = 10; % Number of Elements
422 ND = NE + 1;
423
424 X = linspace(0,1,ND);
425 L = diff(X);
426
427 for i = 1:ND
428
429     if i == 1
430
431         phi(i,1) = (X(i+1) - x)/L(i);
432         phi(i,2) = 0;
433         phi(i,3) = 0;
434         phi(i,4) = 0;
435         phi(i,5) = 0;
436         phi(i,6) = 0;
437         phi(i,7) = 0;
438         phi(i,8) = 0;
439         phi(i,9) = 0;
440         phi(i,10) = 0;
441
442     elseif i == 2
443
444         phi(i,1) = (x - X(i-1))/L(i-1);
445         phi(i,2) = (X(i+1) - x)/L(i);
446         phi(i,3) = 0;
447         phi(i,4) = 0;
448         phi(i,5) = 0;
449         phi(i,6) = 0;
450         phi(i,7) = 0;
451         phi(i,8) = 0;
452         phi(i,9) = 0;
453         phi(i,10) = 0;
454
455     elseif i == 3
456
457         phi(i,1) = 0;
458         phi(i,2) = (x - X(i-1))/L(i-1);
459         phi(i,3) = (X(i+1) - x)/L(i);
460         phi(i,4) = 0;
461         phi(i,5) = 0;
462         phi(i,6) = 0;
463         phi(i,7) = 0;
464         phi(i,8) = 0;
465         phi(i,9) = 0;
466         phi(i,10) = 0;
467

```

```
468 elseif i == 4
469
470 phi(i,1) = 0;
471 phi(i,2) = 0;
472 phi(i,3) = (x - X(i-1))/L(i-1);
473 phi(i,4) = (X(i+1) - x)/L(i);
474 phi(i,5) = 0;
475 phi(i,6) = 0;
476 phi(i,7) = 0;
477 phi(i,8) = 0;
478 phi(i,9) = 0;
479 phi(i,10) = 0;
480
481 elseif i == 5
482
483 phi(i,1) = 0;
484 phi(i,2) = 0;
485 phi(i,3) = 0;
486 phi(i,4) = (x - X(i-1))/L(i-1);
487 phi(i,5) = (X(i+1) - x)/L(i);
488 phi(i,6) = 0;
489 phi(i,7) = 0;
490 phi(i,8) = 0;
491 phi(i,9) = 0;
492 phi(i,10) = 0;
493
494 elseif i == 6
495
496 phi(i,1) = 0;
497 phi(i,2) = 0;
498 phi(i,3) = 0;
499 phi(i,4) = 0;
500 phi(i,5) = (x - X(i-1))/L(i-1);
501 phi(i,6) = (X(i+1) - x)/L(i);
502 phi(i,7) = 0;
503 phi(i,8) = 0;
504 phi(i,9) = 0;
505 phi(i,10) = 0;
506
507 elseif i == 7
508
509 phi(i,1) = 0;
510 phi(i,2) = 0;
511 phi(i,3) = 0;
512 phi(i,4) = 0;
513 phi(i,5) = 0;
514 phi(i,6) = (x - X(i-1))/L(i-1);
```

```

515 phi(i,7) = (X(i+1) - x)/L(i);
516 phi(i,8) = 0;
517 phi(i,9) = 0;
518 phi(i,10) = 0;
519
520 elseif i == 8
521
522 phi(i,1) = 0;
523 phi(i,2) = 0;
524 phi(i,3) = 0;
525 phi(i,4) = 0;
526 phi(i,5) = 0;
527 phi(i,6) = 0;
528 phi(i,7) = (x - X(i-1))/L(i-1);
529 phi(i,8) = (X(i+1) - x)/L(i);
530 phi(i,9) = 0;
531 phi(i,10) = 0;
532
533 elseif i == 9
534
535 phi(i,1) = 0;
536 phi(i,2) = 0;
537 phi(i,3) = 0;
538 phi(i,4) = 0;
539 phi(i,5) = 0;
540 phi(i,6) = 0;
541 phi(i,7) = 0;
542 phi(i,8) = (x - X(i-1))/L(i-1);
543 phi(i,9) = (X(i+1) - x)/L(i);
544 phi(i,10) = 0;
545
546 elseif i == 10
547
548 phi(i,1) = 0;
549 phi(i,2) = 0;
550 phi(i,3) = 0;
551 phi(i,4) = 0;
552 phi(i,5) = 0;
553 phi(i,6) = 0;
554 phi(i,7) = 0;
555 phi(i,8) = 0;
556 phi(i,9) = (x - X(i-1))/L(i-1);
557 phi(i,10) = (X(i+1) - x)/L(i);
558
559 else
560
561 phi(i,1) = 0;

```

```

562 phi(i,2) = 0;
563 phi(i,3) = 0;
564 phi(i,4) = 0;
565 phi(i,5) = 0;
566 phi(i,6) = 0;
567 phi(i,7) = 0;
568 phi(i,8) = 0;
569 phi(i,9) = 0;
570 phi(i,10) = (x - X(i-1))/L(i-1);
571
572 end
573
574 end
575
576 % Force
577
578 P = p*X;
579 P = [P(2:11); P(2:11); P(2:11); P(2:11); P(2:11); P(2:11); P(2:11); ...
      P(2:11); ...
580 P(2:11); P(2:11); P(2:11)];
581 f = P.*phi;
582 F = int(f(:,1),x,X(1),X(2)) + int(f(:,2),x,X(2),X(3)) + ...
      int(f(:,3),x,X(3),X(4)) ...
583 + int(f(:,4),x,X(4),X(5)) + int(f(:,5),x,X(5),X(6)) + ...
      int(f(:,6),x,X(6),X(7)) ...
584 + int(f(:,7),x,X(7),X(8)) + int(f(:,8),x,X(8),X(9)) + ...
      int(f(:,9),x,X(9),X(10)) ...
585 + int(f(:,10),x,X(10),X(11)) + du15dx*subs(phi(:,2),x,1.5) - ...
      du0dx*subs(phi(:,1),x,0);
586 F(1) = F1;
587
588 % Stiffness
589
590 M = [1 -1; -1 1];
591
592 A = pi*r^2;
593 k = E*A./L;
594
595 K1 = k(1)*M;
596 K2 = k(2)*M;
597 K3 = k(3)*M;
598 K4 = k(4)*M;
599 K5 = k(5)*M;
600 K6 = k(6)*M;
601 K7 = k(7)*M;
602 K8 = k(8)*M;
603 K9 = k(9)*M;

```

```

604 K10 = k(10)*M;
605
606 K = zeros (ND);
607 K(1:2,1:2) = K(1:2,1:2) + K1;
608 K(2:3,2:3) = K(2:3,2:3) + K2;
609 K(3:4,3:4) = K(3:4,3:4) + K3;
610 K(4:5,4:5) = K(4:5,4:5) + K4;
611 K(5:6,5:6) = K(5:6,5:6) + K5;
612 K(6:7,6:7) = K(6:7,6:7) + K6;
613 K(7:8,7:8) = K(7:8,7:8) + K7;
614 K(8:9,8:9) = K(8:9,8:9) + K8;
615 K(9:10,9:10) = K(9:10,9:10) + K9;
616 K(10:11,10:11) = K(10:11,10:11) + K10;
617
618 % Displacement
619
620 U = [u1; u2; u3; u4; u5; u6; u7; u8; u9; u10; u11];
621
622 % Calculations
623
624 AN = solve(F - K*U);
625 F(1) = AN.F1;
626 U(2:11) = [AN.u2; AN.u3; AN.u4; AN.u5; AN.u6; AN.u7; AN.u8; AN.u9; ...
            AN.u10; AN.u11];
627
628 % Plot
629
630 figure(1)
631 hold on
632 plot(X,F/A,'-oc','MarkerFaceColor','c')
633 legend('4 Elementos', '6 Elementos', '8 Elementos', '10 ...
        Elementos','Location','southeast')
634 xlabel('Comprimento [m]')
635 ylabel('Tensão [Pa]')
636 grid minor
637
638 figure(2)
639 hold on
640 plot(X,U,'-oc','MarkerFaceColor','c')
641 legend('4 Elementos', '6 Elementos', '8 Elementos', '10 ...
        Elementos','Location','southeast')
642 xlabel('Comprimento Inicial [m]')
643 ylabel('Deslocamento [m]')
644 grid minor

```

Resultados

Item (a)

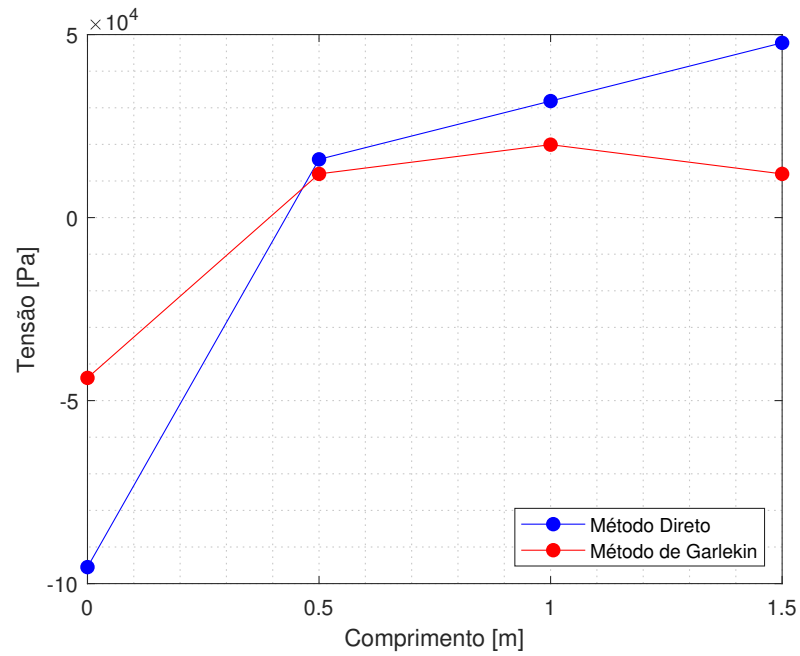


Figura 8: Tensões na barra

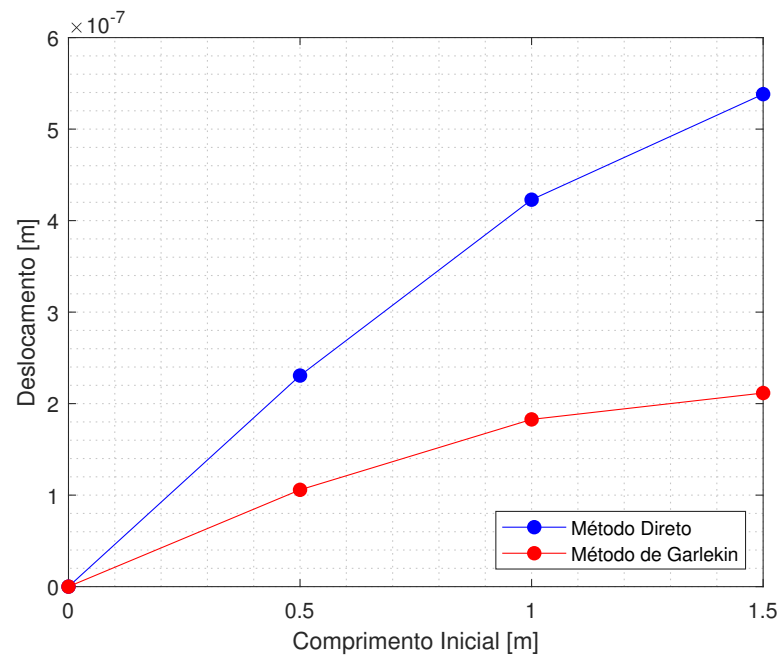


Figura 9: Deslocamentos dos nós

Item (b)

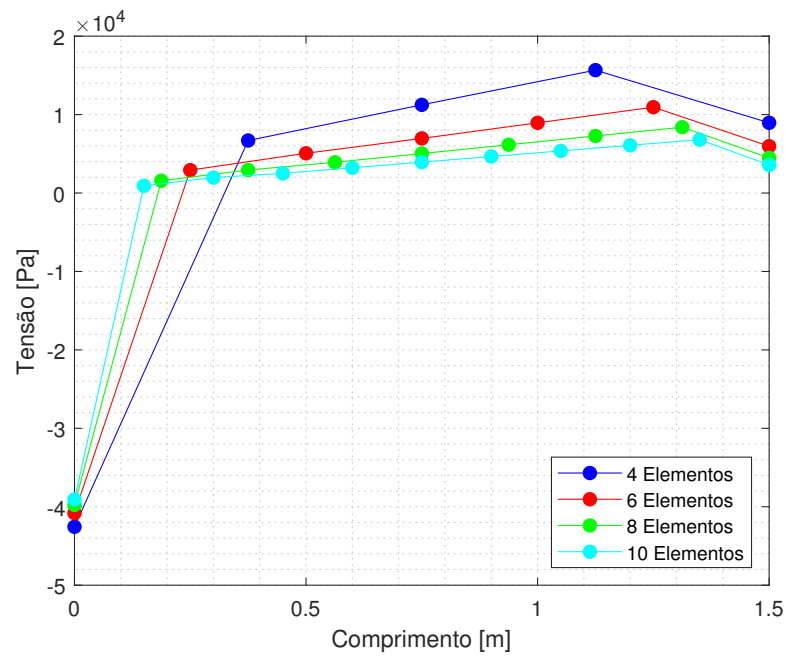


Figura 10: Comparativo de tensões na barra

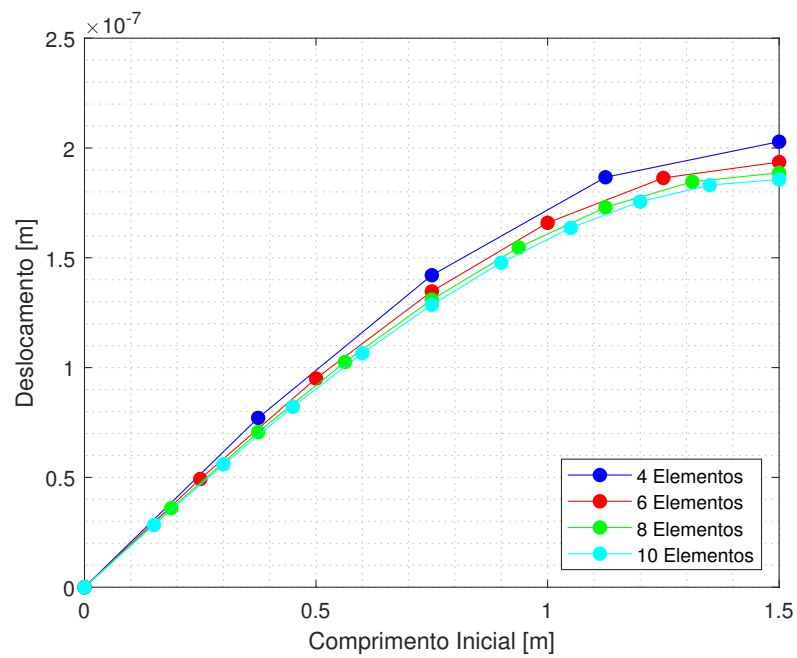


Figura 11: Comparativo de deslocamentos dos nós