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

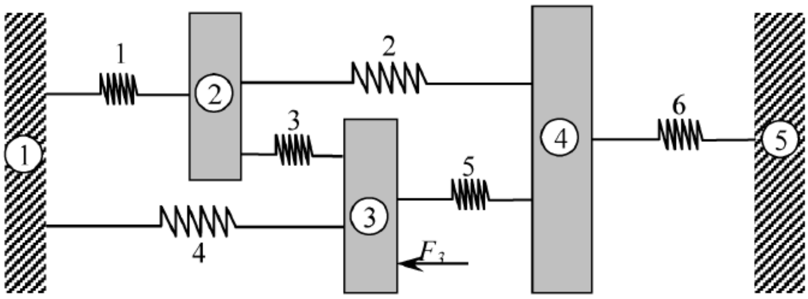


Figura 1: Sistema das questões 1 e 2

Matriz de Rigidez Global

Stiffness 1

	u1	u2	u3	u4	u5
u1	k1	-k1			
u2	-k1	k1			
u3					
u4					
u5					

Stiffness 2

	u1	u2	u3	u4	u5
u1					
u2		k2		-k2	
u3					
u4		-k2		k2	
u5					

Stiffness 3

	u1	u2	u3	u4	u5
u1					
u2		k3	-k3		
u3		-k3	k3		
u4					
u5					

Stiffness 4

	u1	u2	u3	u4	u5
u1	k4		-k4		
u2					
u3	-k4		k4		
u4					
u5					

Stiffness 5

	u1	u2	u3	u4	u5
u1					
u2					
u3			k5	-k5	
u4			-k5	k5	
u5					

Stiffness 6

	u1	u2	u3	u4	u5
u1					
u2					
u3					
u4				k6	-k6
u5				-k6	k6

Global Matrix

				u1	u2	u3	u4	u5	
	R12	R13	F1						u1
R21	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)

```
clear
close all
clc

%% Inputs

F2 = 0; % Force [N]
F3 = -1000; % Force [N]
F4 = 0; % Force [N]
u1 = 0; % Displacement [mm]
u5 = 0; % Displacement [mm]
k1 = 500; % Spring Constant [N/mm]
k2 = 400; % Spring Constant [N/mm]
k3 = 600; % Spring Constant [N/mm]
k4 = 200; % Spring Constant [N/mm]
k5 = 400; % Spring Constant [N/mm]
k6 = 300; % Spring Constant [N/mm]
n = 5; % Degrees of Freedom

%% Characterization

A = [1 -1; -1 1];

% Stiffness

K1 = k1*A;
K2 = k2*A;
K3 = k3*A;
K4 = k4*A;
K5 = k5*A;
K6 = k6*A;

% Global Matrix Equation

K = zeros(n);
K(1:2,1:2) = K(1:2,1:2) + K1;
K(2:2:4,2:2:4) = K(2:2:4,2:2:4) + K2;
K(2:3,2:3) = K(2:3,2:3) + K3;
K(1:2:3,1:2:3) = K(1:2:3,1:2:3) + K4;
K(3:4,3:4) = K(3:4,3:4) + K5;
K(4:5,4:5) = K(4:5,4:5) + K6;

% Force Vector

syms F1 F5
F = [F1; F2; F3; F4; F5];
```

```
% Displacement Vectors

syms u2 u3 u4
U = [u1; u2; u3; u4; u5];

%% Calculations

AN = solve(F-K*U);

F = double([AN.F1; F2; F3; F4; AN.F5]);
U = double([u1; AN.u2; AN.u3; AN.u4; u5]);

f1 = k1*(U(1) - U(2));
f2 = k2*(U(2) - U(4));
f3 = k3*(U(2) - U(3));
f4 = k4*(U(1) - U(3));
f5 = k5*(U(3) - U(4));
f6 = k6*(U(4) - U(5));

f = [f1; f2; f3; f4; f5; f6];

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)

```
clear
close all
clc

%% Inputs

F3 = -1000; % Force [N]
F4 = 0; % Force [N]
u1 = 0; % Displacement [mm]
u2 = 0; % Displacement [mm]
u5 = 0; % Displacement [mm]
k1 = 500; % Spring Constant [N/mm]
k2 = 400; % Spring Constant [N/mm]
k3 = 600; % Spring Constant [N/mm]
k4 = 200; % Spring Constant [N/mm]
k5 = 400; % Spring Constant [N/mm]
k6 = 300; % Spring Constant [N/mm]
n = 5; % Degrees of Freedom

%% Characterization

A = [1 -1; -1 1];

% Stiffness

K1 = k1*A;
K2 = k2*A;
K3 = k3*A;
K4 = k4*A;
K5 = k5*A;
K6 = k6*A;

% Global Matrix Equation

K = zeros(n);
K(1:2,1:2) = K(1:2,1:2) + K1;
K(2:2:4,2:2:4) = K(2:2:4,2:2:4) + K2;
K(2:3,2:3) = K(2:3,2:3) + K3;
K(1:2:3,1:2:3) = K(1:2:3,1:2:3) + K4;
K(3:4,3:4) = K(3:4,3:4) + K5;
K(4:5,4:5) = K(4:5,4:5) + K6;
```

```

% Force Vector

syms F1 F2 F5
F = [F1; F2; F3; F4; F5];

% Displacement Vectors

syms u3 u4
U = [u1; u2; u3; u4; u5];

%% Calculations

AN = solve(F-K*U);

F = double([AN.F1; AN.F2; F3; F4; AN.F5]);
U = double([u1; u2; AN.u3; AN.u4; u5]);

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.

Questão 3

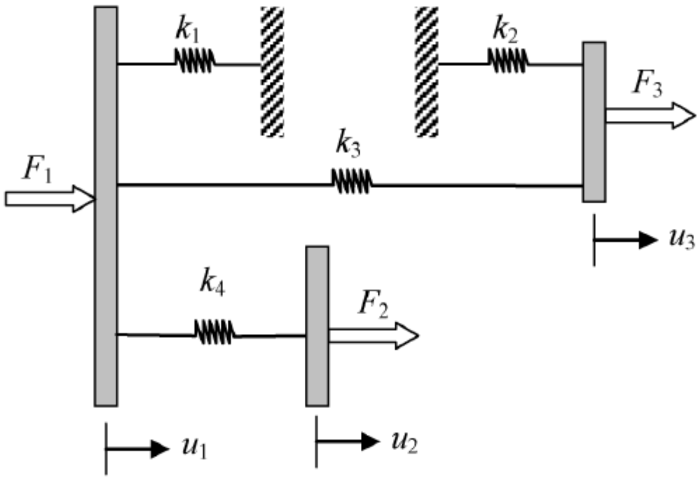
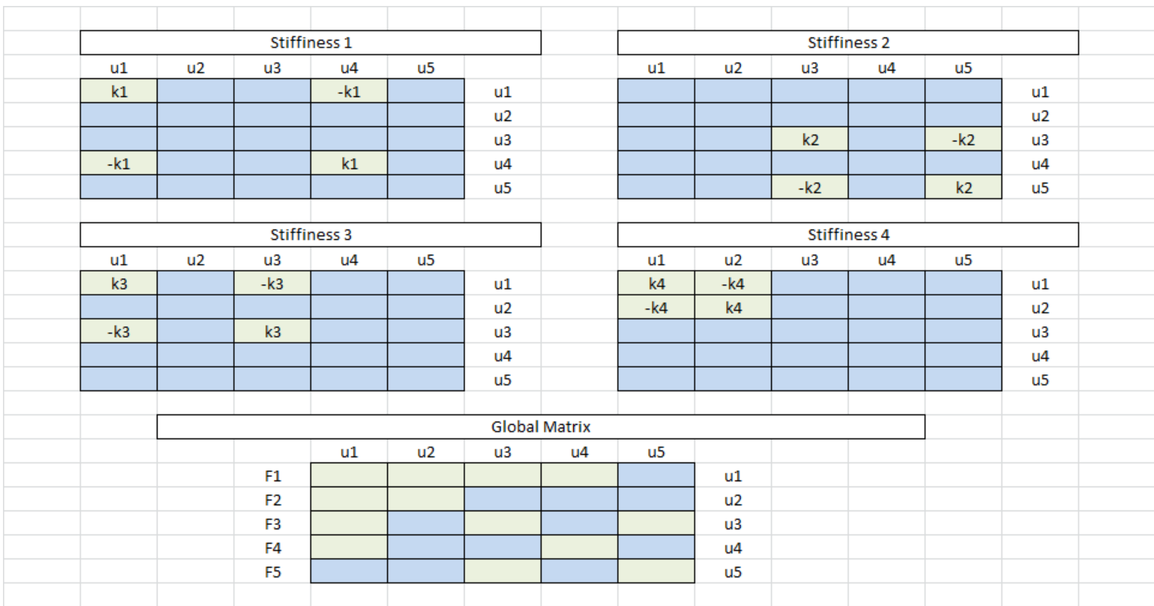


Figura 3: Sistema da questão 3

Matriz de Rigidez Global



Stiffiness 1	u1	u2	u3	u4	u5
u1	k1			-k1	
u2					
u3					
u4	-k1			k1	
u5					

Stiffiness 2	u1	u2	u3	u4	u5
u1					
u2					
u3			k2		-k2
u4					
u5			-k2		k2

Stiffiness 3	u1	u2	u3	u4	u5
u1	k3		-k3		
u2					
u3	-k3		k3		
u4					
u5					

Stiffiness 4	u1	u2	u3	u4	u5
u1	k4	-k4			
u2	-k4	k4			
u3					
u4					
u5					

Global Matrix	u1	u2	u3	u4	u5	
F1						u1
F2						u2
F3						u3
F4						u4
F5						u5

Figura 4: Obtenção da matriz de rigidez global da questão 3

Script (MATLAB)

```
clear
close all
clc

%% Inputs

syms k1 k2 k3 k4 F1 F2 F3 F4 F5 u1 u2 u3

u4 = 0; % Displacement [mm]
u5 = 0; % Displacement [mm]

%% Characterization

% Stiffness

K = [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];

% Force Vector

F = [F1; F2; F3; F4; F5];

% Displacement Vectors

U = [u1; u2; u3; u4; u5];

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

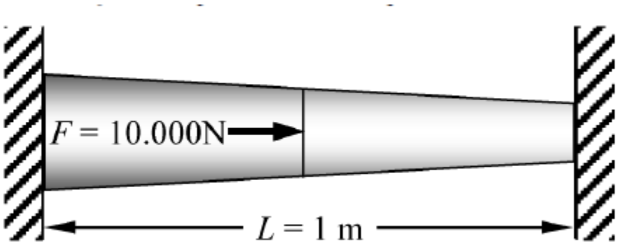


Figura 5: Sistema da questão 4

Script (MATLAB)

```
clear
close all
clc

%% Inputs

F2 = 0; % Force [N]
F3 = 10000; % Force [N]
F4 = 0; % Force [N]
u1 = 0; % Displacement [m]
u2 = 0; % Displacement [m]
L = 1; % Lenght [m]
E = 100*10^6; % Young's Modulus [Pa]
n = 5; % Degrees of Freedom

%% Characterization

M = [1 -1; -1 1];

% Stiffness

x = linspace(0,L,n);
r = 0.05 - 0.04.*(x); % Radius [m]
A = @(r) pi*r(1:end-1).*r(2:end); % Area [m^2]
A = A(r);

k = E.*A./(diff(x));

K1 = k(1)*M;
K2 = k(2)*M;
```

```
K3 = k(3)*M;
K4 = k(4)*M;

% Global Matrix Equation

K = zeros(n);
K(1:2,1:2) = K(1:2,1:2) + K1;
K(2:3,2:3) = K(2:3,2:3) + K2;
K(3:4,3:4) = K(3:4,3:4) + K3;
K(4:5,4:5) = K(4:5,4:5) + K4;

% Force Vector

syms F1 F5
F = [F1; F2; F3; F4; F5];

% Displacement Vectors

syms u3 u4 u5
U = [u1; u3; u4; u5; u2];

%% Calculations

AN = solve(F-K*U);

F = double([AN.F1; F2; F3; F4; AN.F5]);
U = double([u1; AN.u3; AN.u4; AN.u5; u2]);

f = k'.*diff(U);

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

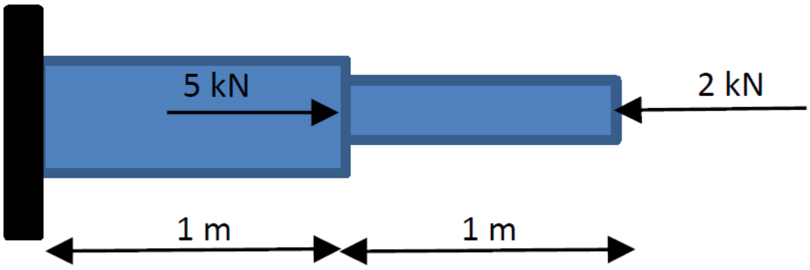


Figura 6: Sistema da questão 5

Script (MATLAB)

```
clear
close all
clc

%% Inputs

F2 = 5000; % Force [N]
F3 = -2000; % Force [N]
u1 = 0; % Displacement [m]
L1 = 1; % Lenght [m]
L2 = 1; % Lenght [m]
E = 100*10^9; % Young's Modulus [Pa]
A1 = 1*10^(-4); % Area [m^2]
A2 = 2*10^(-4); % Area [m^2]
n = 3; % Degrees of Freedom

%% Characterization

M = [1 -1; -1 1];

% Stiffness

A = [A1; A2];
x = cumsum([0; L1; L2]);

k = E.*A./(diff(x));

K1 = k(1)*M;
```

```
K2 = k(2)*M;

% Global Matrix Equation

K = zeros(n);
K(1:2,1:2) = K(1:2,1:2) + K1;
K(2:3,2:3) = K(2:3,2:3) + K2;

% Force Vector

syms F1
F = [F1; F2; F3];

% Displacement Vectors

syms u2 u3
U = [u1; u2; u3];

%% Calculations

AN = solve(F-K*U);

F = double([AN.F1; F2; F3]);
U = double([u1; AN.u2; AN.u3]);

f = k.*diff(U);

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)

```
clear
close all
clc

%% Inputs

F2 = 0; % Force [N]
F3 = 5000; % Force [N]
F4 = 0; % Force [N]
F5 = -2000; % Force [N]
u1 = 0; % Displacement [m]
L1 = 1; % Lenght [m]
L2 = 1; % Lenght [m]
E = 100*10^9; % Young's Modulus [Pa]
A1 = 1*10^(-4); % Area [m^2]
A2 = 2*10^(-4); % Area [m^2]
n = 5; % Degrees of Freedom

%% Characterization

M = [1 -1; -1 1];

% Stiffness

A = [A1; A1; A2; A2];
x = cumsum([0; L1/2; L1/2; L2/2; L2/2]);

k = E.*A./(diff(x));

K1 = k(1)*M;
K2 = k(2)*M;
K3 = k(3)*M;
K4 = k(4)*M;

% Global Matrix Equation

K = zeros(n);
K(1:2,1:2) = K(1:2,1:2) + K1;
K(2:3,2:3) = K(2:3,2:3) + K2;
K(3:4,3:4) = K(3:4,3:4) + K3;
K(4:5,4:5) = K(4:5,4:5) + K4;
```

```
% Force Vector

syms F1
F = [F1; F2; F3; F4; F5];

% Displacement Vectors

syms u2 u3 u4 u5
U = [u1; u2; u3; u4; u5];

%% Calculations

AN = solve(F-K*U);

F = double([AN.F1; F2; F3; F4; F5]);
U = double([u1; AN.u2; AN.u3; AN.u4; AN.u5]);

f = k.*diff(U);

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)

$$\begin{bmatrix} R1 \\ F2 \\ F3 \\ F4 \\ F5 \end{bmatrix} = \begin{bmatrix} k_1 & -k_1 & 0 & 0 & 0 \\ -k_1 & k_1+k_2 & -k_2 & 0 & 0 \\ 0 & -k_2 & k_2+k_3 & -k3 & 0 \\ 0 & 0 & -k_3 & k_3+k_4 & -k_4 \\ 0 & 0 & 0 & -k_4 & k_4 \end{bmatrix} \begin{bmatrix} u1 \\ u2 \\ u3 \\ u4 \\ u5 \end{bmatrix}$$

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

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

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

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

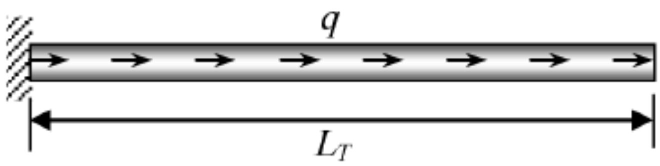


Figura 7: Sistema da questão 7

Script (MATLAB)

Item (a)

```
clear
close all
clc

%% Inputs

u1 = 0; % Boundary Condition
du15dx = 1; % Boundary Condition
p = 1000; % Load [N/m]
l = 1.5; % Length [m]
r = 0.1; % Ray [m]
E = 207*10^9; % Young's Modulus [Pa]
NE = 3; % Number of Elements

%% FEM - Direct Method

ND = NE + 1;
M = [1 -1; -1 1];

% Stiffness

A = pi*r^2;
X = linspace(0,l,ND);
L = diff(X);

k = E*A./L;

K1 = k(1)*M;
K2 = k(2)*M;
K3 = k(3)*M;
```

```
% Global Matrix Equation

K = zeros(ND);
K(1:2,1:2) = K(1:2,1:2) + K1;
K(2:3,2:3) = K(2:3,2:3) + K2;
K(3:4,3:4) = K(3:4,3:4) + K3;

% Force Vector

syms F1 F2 F3 F4

F = [F1; F2; F3; F4];
aux = p*X;
F(2:4) = aux(2:4);

% Displacement Vectors

syms u2 u3 u4
U = [u1; u2; u3; u4];

% Calculations

AN = solve(F-K*U);

F(1) = AN.F1;
U(2:4) = [AN.u2; AN.u3; AN.u4];

F = double(F);
U = double(U);

% Plots

figure
plot(X,F/A,'-ob','MarkerFaceColor','b')

figure
plot(X,U,'-ob','MarkerFaceColor','b')

%% Garlekin's Method

% Characterization

syms x F1 u2 u3 u4 du0dx
ND = NE + 1;

X = linspace(0,1,ND);
```

```
L = diff(X);

for i = 1:ND

if i == 1

phi(i,1) = (X(i+1) - x)/L(i);
phi(i,2) = 0;
phi(i,3) = 0;

elseif i == ND

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = (x - X(i-1))/L(i-1);

elseif i == 2

phi(i,1) = (x - X(i-1))/L(i-1);
phi(i,2) = (X(i+1) - x)/L(i);
phi(i,3) = 0;

else

phi(i,1) = 0;
phi(i,2) = (x - X(i-1))/L(i-1);
phi(i,3) = (X(i+1) - x)/L(i);

end

end

% Force

P = p*X;
P = [P(2:4); P(2:4); P(2:4); P(2:4)];
f = P.*phi;
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);
F(1) = F1;

% Stiffness

M = [1 -1; -1 1];

A = pi*r^2;
k = E*A./L;
```

```
K1 = k(1)*M;
K2 = k(2)*M;
K3 = k(3)*M;

K = zeros(ND);
K(1:2,1:2) = K(1:2,1:2) + K1;
K(2:3,2:3) = K(2:3,2:3) + K2;
K(3:4,3:4) = K(3:4,3:4) + K3;

% Displacement

U = [u1; u2; u3; u4];

% Calculations

AN = solve(F - K*U);
F(1) = AN.F1;
U(2:4) = [AN.u2; AN.u3; AN.u4];

% Plot

figure(1)
hold on
plot(X,F/A,'-or','MarkerFaceColor','r')
legend('Método Direto', 'Método de Garlekin','Location','southeast')
xlabel('Comprimento [m]')
ylabel('Tensão [Pa]')
grid minor

figure(2)
hold on
plot(X,U,'-or','MarkerFaceColor','r')
legend('Método Direto', 'Método de Garlekin','Location','southeast')
xlabel('Comprimento Inicial [m]')
ylabel('Deslocamento [m]')
grid minor
```

Item (b)

```
clear
close all
clc

%% Inputs
```

```
u1 = 0; % Boundary Condition
du15dx = 1; % Boundary Condition
p = 1000; % Load [N/m]
l = 1.5; % Length [m]
r = 0.1; % Ray [m]
E = 207*10^9; % Young's Modulus [Pa]

%% Garlekin's Method (4 Elements)

% Characterization

syms x F1 u2 u3 u4 u5 du0dx
NE = 4; % Number of Elements
ND = NE + 1;

X = linspace(0,l,ND);
L = diff(X);

for i = 1:ND

if i == 1

phi(i,1) = (X(i+1) - x)/L(i);
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = 0;

elseif i == 2

phi(i,1) = (x - X(i-1))/L(i-1);
phi(i,2) = (X(i+1) - x)/L(i);
phi(i,3) = 0;
phi(i,4) = 0;

elseif i == 3

phi(i,1) = 0;
phi(i,2) = (x - X(i-1))/L(i-1);
phi(i,3) = (X(i+1) - x)/L(i);
phi(i,4) = 0;

elseif i == 4

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = (x - X(i-1))/L(i-1);
```

```
phi(i,4) = (X(i+1) - x)/L(i);

else

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = (x - X(i-1))/L(i-1);

end

end

% Force

P = p*X;
P = [P(2:5); P(2:5); P(2:5); P(2:5); P(2:5)];
f = P.*phi;
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)) ...
+ int(f(:,4),x,X(4),X(5)) + du15dx*subs(phi(:,2),x,1.5) - ...
    du0dx*subs(phi(:,1),x,0);
F(1) = F1;

% Stiffness

M = [1 -1; -1 1];

A = pi*r^2;
k = E*A./L;

K1 = k(1)*M;
K2 = k(2)*M;
K3 = k(3)*M;
K4 = k(4)*M;

K = zeros(ND);
K(1:2,1:2) = K(1:2,1:2) + K1;
K(2:3,2:3) = K(2:3,2:3) + K2;
K(3:4,3:4) = K(3:4,3:4) + K3;
K(4:5,4:5) = K(4:5,4:5) + K4;

% Displacement

U = [u1; u2; u3; u4; u5];

% Calculations
```

```

AN = solve(F - K*U);
F(1) = AN.F1;
U(2:5) = [AN.u2; AN.u3; AN.u4; AN.u5];

% Plot

figure
plot(X,F/A,'-ob','MarkerFaceColor','b')

figure
plot(X,U,'-ob','MarkerFaceColor','b')

%% Garlekin's Method (6 Elements)

% Characterization

syms x F1 u2 u3 u4 u5 u6 u7 du0dx
NE = 6; % Number of Elements
ND = NE + 1;

X = linspace(0,l,ND);
L = diff(X);

for i = 1:ND

if i == 1

phi(i,1) = (X(i+1) - x)/L(i);
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = 0;
phi(i,6) = 0;

elseif i == 2

phi(i,1) = (x - X(i-1))/L(i-1);
phi(i,2) = (X(i+1) - x)/L(i);
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = 0;
phi(i,6) = 0;

elseif i == 3

phi(i,1) = 0;
```

```
phi(i,2) = (x - X(i-1))/L(i-1);
phi(i,3) = (X(i+1) - x)/L(i);
phi(i,4) = 0;
phi(i,5) = 0;
phi(i,6) = 0;

elseif i == 4

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = (x - X(i-1))/L(i-1);
phi(i,4) = (X(i+1) - x)/L(i);
phi(i,5) = 0;
phi(i,6) = 0;

elseif i == 5

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = (x - X(i-1))/L(i-1);
phi(i,5) = (X(i+1) - x)/L(i);
phi(i,6) = 0;

elseif i == 6

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = (x - X(i-1))/L(i-1);
phi(i,6) = (X(i+1) - x)/L(i);

else

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = 0;
phi(i,6) = (x - X(i-1))/L(i-1);

end

end

% Force
```

```
P = p*X;
P = [P(2:7); P(2:7); P(2:7); P(2:7); P(2:7); P(2:7); P(2:7)];
f = P.*phi;
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)) ...
+ int(f(:,4),x,X(4),X(5)) + int(f(:,5),x,X(5),X(6)) + ...
    int(f(:,6),x,X(6),X(7)) ...
+ du15dx*subs(phi(:,2),x,1.5) - du0dx*subs(phi(:,1),x,0);
F(1) = F1;

% Stiffness

M = [1 -1; -1 1];

A = pi*r^2;
k = E*A./L;

K1 = k(1)*M;
K2 = k(2)*M;
K3 = k(3)*M;
K4 = k(4)*M;
K5 = k(5)*M;
K6 = k(6)*M;

K = zeros(ND);
K(1:2,1:2) = K(1:2,1:2) + K1;
K(2:3,2:3) = K(2:3,2:3) + K2;
K(3:4,3:4) = K(3:4,3:4) + K3;
K(4:5,4:5) = K(4:5,4:5) + K4;
K(5:6,5:6) = K(5:6,5:6) + K5;
K(6:7,6:7) = K(6:7,6:7) + K6;

% Displacement

U = [u1; u2; u3; u4; u5; u6; u7];

% Calculations

AN = solve(F - K*U);
F(1) = AN.F1;
U(2:7) = [AN.u2; AN.u3; AN.u4; AN.u5; AN.u6; AN.u7];

% Plot

figure(1)
hold on
```

```
plot(X,F/A,'-or','MarkerFaceColor','r')

figure(2)
hold on
plot(X,U,'-or','MarkerFaceColor','r')

%% Garlekin's Method (8 Elements)

% Characterization

syms x F1 u2 u3 u4 u5 u6 u7 u8 u9 du0dx
NE = 8; % Number of Elements
ND = NE + 1;

X = linspace(0,1,ND);
L = diff(X);

for i = 1:ND

if i == 1

phi(i,1) = (X(i+1) - x)/L(i);
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = 0;
phi(i,6) = 0;
phi(i,7) = 0;
phi(i,8) = 0;

elseif i == 2

phi(i,1) = (x - X(i-1))/L(i-1);
phi(i,2) = (X(i+1) - x)/L(i);
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = 0;
phi(i,6) = 0;
phi(i,7) = 0;
phi(i,8) = 0;

elseif i == 3

phi(i,1) = 0;
phi(i,2) = (x - X(i-1))/L(i-1);
phi(i,3) = (X(i+1) - x)/L(i);
phi(i,4) = 0;
```

```
phi(i,5) = 0;
phi(i,6) = 0;
phi(i,7) = 0;
phi(i,8) = 0;

elseif i == 4

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = (x - X(i-1))/L(i-1);
phi(i,4) = (X(i+1) - x)/L(i);
phi(i,5) = 0;
phi(i,6) = 0;
phi(i,7) = 0;
phi(i,8) = 0;

elseif i == 5

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = (x - X(i-1))/L(i-1);
phi(i,5) = (X(i+1) - x)/L(i);
phi(i,6) = 0;
phi(i,7) = 0;
phi(i,8) = 0;

elseif i == 6

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = (x - X(i-1))/L(i-1);
phi(i,6) = (X(i+1) - x)/L(i);
phi(i,7) = 0;
phi(i,8) = 0;

elseif i == 7

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = 0;
phi(i,6) = (x - X(i-1))/L(i-1);
phi(i,7) = (X(i+1) - x)/L(i);
```

```
phi(i,8) = 0;

elseif i == 8

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = 0;
phi(i,6) = 0;
phi(i,7) = (x - X(i-1))/L(i-1);
phi(i,8) = (X(i+1) - x)/L(i);

else

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = 0;
phi(i,6) = 0;
phi(i,7) = 0;
phi(i,8) = (x - X(i-1))/L(i-1);

end

end

% Force

P = p*X;
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)];
f = P.*phi;
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)) ...
+ int(f(:,4),x,X(4),X(5)) + int(f(:,5),x,X(5),X(6)) + ...
    int(f(:,6),x,X(6),X(7)) ...
+ int(f(:,7),x,X(7),X(8)) + int(f(:,8),x,X(8),X(9)) + ...
    du15dx*subs(phi(:,2),x,1.5) ...
- du0dx*subs(phi(:,1),x,0);
F(1) = F1;

% Stiffness

M = [1 -1; -1 1];
```

```
A = pi*r^2;
k = E*A./L;

K1 = k(1)*M;
K2 = k(2)*M;
K3 = k(3)*M;
K4 = k(4)*M;
K5 = k(5)*M;
K6 = k(6)*M;
K7 = k(7)*M;
K8 = k(8)*M;

K = zeros(ND);
K(1:2,1:2) = K(1:2,1:2) + K1;
K(2:3,2:3) = K(2:3,2:3) + K2;
K(3:4,3:4) = K(3:4,3:4) + K3;
K(4:5,4:5) = K(4:5,4:5) + K4;
K(5:6,5:6) = K(5:6,5:6) + K5;
K(6:7,6:7) = K(6:7,6:7) + K6;
K(7:8,7:8) = K(7:8,7:8) + K7;
K(8:9,8:9) = K(8:9,8:9) + K8;

% Displacement

U = [u1; u2; u3; u4; u5; u6; u7; u8; u9];

% Calculations

AN = solve(F - K*U);
F(1) = AN.F1;
U(2:9) = [AN.u2; AN.u3; AN.u4; AN.u5; AN.u6; AN.u7; AN.u8; AN.u9];

% Plot

figure(1)
hold on
plot(X,F/A,'-og','MarkerFaceColor','g')

figure(2)
hold on
plot(X,U,'-og','MarkerFaceColor','g')

%% Garlekin's Method (10 Elements)

% Characterization

syms x F1 u2 u3 u4 u5 u6 u7 u8 u9 u10 u11 du0dx
```

```
NE = 10; % Number of Elements
ND = NE + 1;

X = linspace(0,1,ND);
L = diff(X);

for i = 1:ND

if i == 1

phi(i,1) = (X(i+1) - x)/L(i);
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = 0;
phi(i,6) = 0;
phi(i,7) = 0;
phi(i,8) = 0;
phi(i,9) = 0;
phi(i,10) = 0;

elseif i == 2

phi(i,1) = (x - X(i-1))/L(i-1);
phi(i,2) = (X(i+1) - x)/L(i);
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = 0;
phi(i,6) = 0;
phi(i,7) = 0;
phi(i,8) = 0;
phi(i,9) = 0;
phi(i,10) = 0;

elseif i == 3

phi(i,1) = 0;
phi(i,2) = (x - X(i-1))/L(i-1);
phi(i,3) = (X(i+1) - x)/L(i);
phi(i,4) = 0;
phi(i,5) = 0;
phi(i,6) = 0;
phi(i,7) = 0;
phi(i,8) = 0;
phi(i,9) = 0;
phi(i,10) = 0;

```

```
elseif i == 4

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = (x - X(i-1))/L(i-1);
phi(i,4) = (X(i+1) - x)/L(i);
phi(i,5) = 0;
phi(i,6) = 0;
phi(i,7) = 0;
phi(i,8) = 0;
phi(i,9) = 0;
phi(i,10) = 0;

elseif i == 5

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = (x - X(i-1))/L(i-1);
phi(i,5) = (X(i+1) - x)/L(i);
phi(i,6) = 0;
phi(i,7) = 0;
phi(i,8) = 0;
phi(i,9) = 0;
phi(i,10) = 0;

elseif i == 6

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = (x - X(i-1))/L(i-1);
phi(i,6) = (X(i+1) - x)/L(i);
phi(i,7) = 0;
phi(i,8) = 0;
phi(i,9) = 0;
phi(i,10) = 0;

elseif i == 7

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = 0;
phi(i,6) = (x - X(i-1))/L(i-1);
```

```
phi(i,7) = (X(i+1) - x)/L(i);
phi(i,8) = 0;
phi(i,9) = 0;
phi(i,10) = 0;

elseif i == 8

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = 0;
phi(i,6) = 0;
phi(i,7) = (x - X(i-1))/L(i-1);
phi(i,8) = (X(i+1) - x)/L(i);
phi(i,9) = 0;
phi(i,10) = 0;

elseif i == 9

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = 0;
phi(i,6) = 0;
phi(i,7) = 0;
phi(i,8) = (x - X(i-1))/L(i-1);
phi(i,9) = (X(i+1) - x)/L(i);
phi(i,10) = 0;

elseif i == 10

phi(i,1) = 0;
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = 0;
phi(i,6) = 0;
phi(i,7) = 0;
phi(i,8) = 0;
phi(i,9) = (x - X(i-1))/L(i-1);
phi(i,10) = (X(i+1) - x)/L(i);

else

phi(i,1) = 0;
```

```
phi(i,2) = 0;
phi(i,3) = 0;
phi(i,4) = 0;
phi(i,5) = 0;
phi(i,6) = 0;
phi(i,7) = 0;
phi(i,8) = 0;
phi(i,9) = 0;
phi(i,10) = (x - X(i-1))/L(i-1);

end

end

% Force

P = p*X;
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); ...
P(2:11); P(2:11); P(2:11)];
f = P.*phi;
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)) ...
+ int(f(:,4),x,X(4),X(5)) + int(f(:,5),x,X(5),X(6)) + ...
    int(f(:,6),x,X(6),X(7)) ...
+ int(f(:,7),x,X(7),X(8)) + int(f(:,8),x,X(8),X(9)) + ...
    int(f(:,9),x,X(9),X(10)) ...
+ int(f(:,10),x,X(10),X(11)) + du15dx*subs(phi(:,2),x,1.5) - ...
    du0dx*subs(phi(:,1),x,0);
F(1) = F1;

% Stiffness

M = [1 -1; -1 1];

A = pi*r^2;
k = E*A./L;

K1 = k(1)*M;
K2 = k(2)*M;
K3 = k(3)*M;
K4 = k(4)*M;
K5 = k(5)*M;
K6 = k(6)*M;
K7 = k(7)*M;
K8 = k(8)*M;
K9 = k(9)*M;
```

```
K10 = k(10)*M;

K = zeros(ND);
K(1:2,1:2) = K(1:2,1:2) + K1;
K(2:3,2:3) = K(2:3,2:3) + K2;
K(3:4,3:4) = K(3:4,3:4) + K3;
K(4:5,4:5) = K(4:5,4:5) + K4;
K(5:6,5:6) = K(5:6,5:6) + K5;
K(6:7,6:7) = K(6:7,6:7) + K6;
K(7:8,7:8) = K(7:8,7:8) + K7;
K(8:9,8:9) = K(8:9,8:9) + K8;
K(9:10,9:10) = K(9:10,9:10) + K9;
K(10:11,10:11) = K(10:11,10:11) + K10;

% Displacement

U = [u1; u2; u3; u4; u5; u6; u7; u8; u9; u10; u11];

% Calculations

AN = solve(F - K*U);
F(1) = AN.F1;
U(2:11) = [AN.u2; AN.u3; AN.u4; AN.u5; AN.u6; AN.u7; AN.u8; AN.u9; ...
    AN.u10; AN.u11];

% Plot

figure(1)
hold on
plot(X,F/A,'-oc','MarkerFaceColor','c')
legend('4 Elementos', '6 Elementos', '8 Elementos', '10 ...
    Elementos','Location','southeast')
xlabel('Comprimento [m]')
ylabel('Tensão [Pa]')
grid minor

figure(2)
hold on
plot(X,U,'-oc','MarkerFaceColor','c')
legend('4 Elementos', '6 Elementos', '8 Elementos', '10 ...
    Elementos','Location','southeast')
xlabel('Comprimento Inicial [m]')
ylabel('Deslocamento [m]')
grid minor
```

Resultados

Item (a)

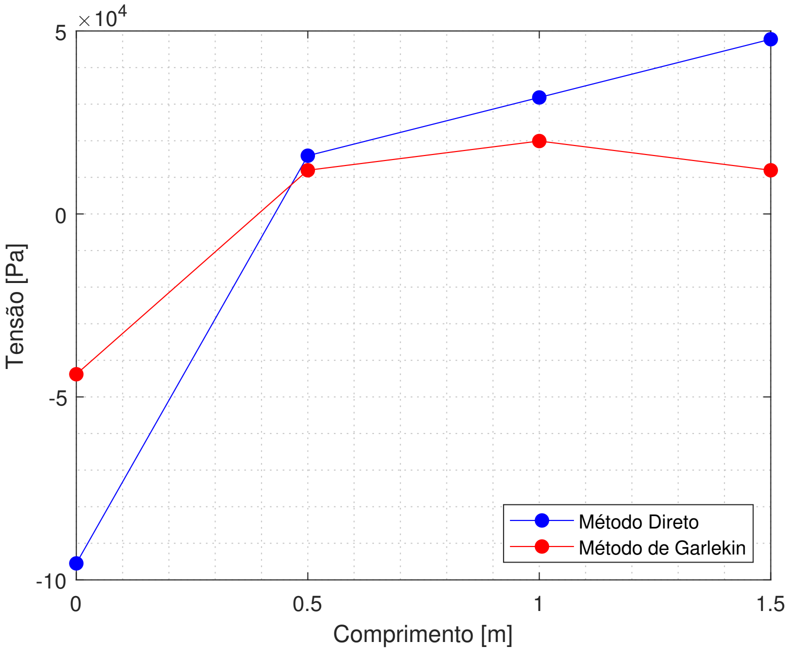


Figura 8: Tensões na barra

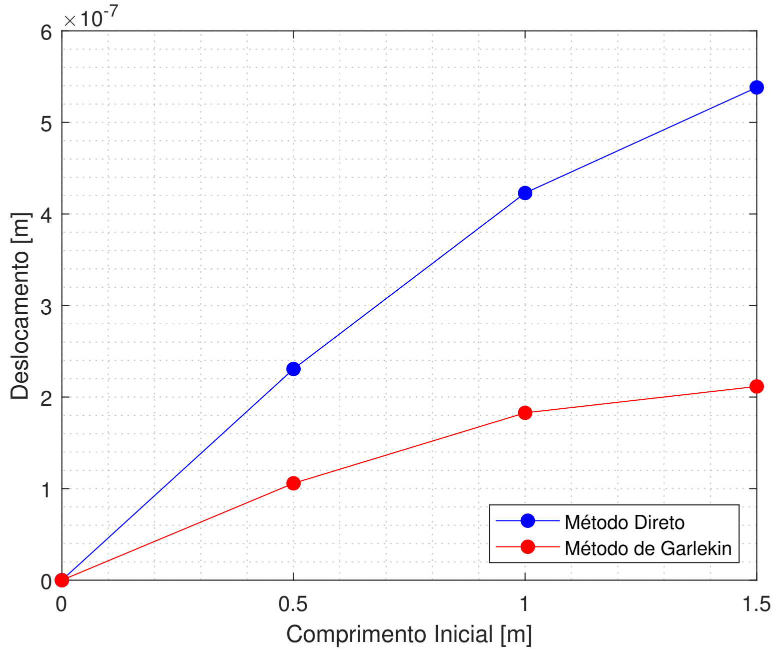


Figura 9: Deslocamentos dos nós

Item (b)

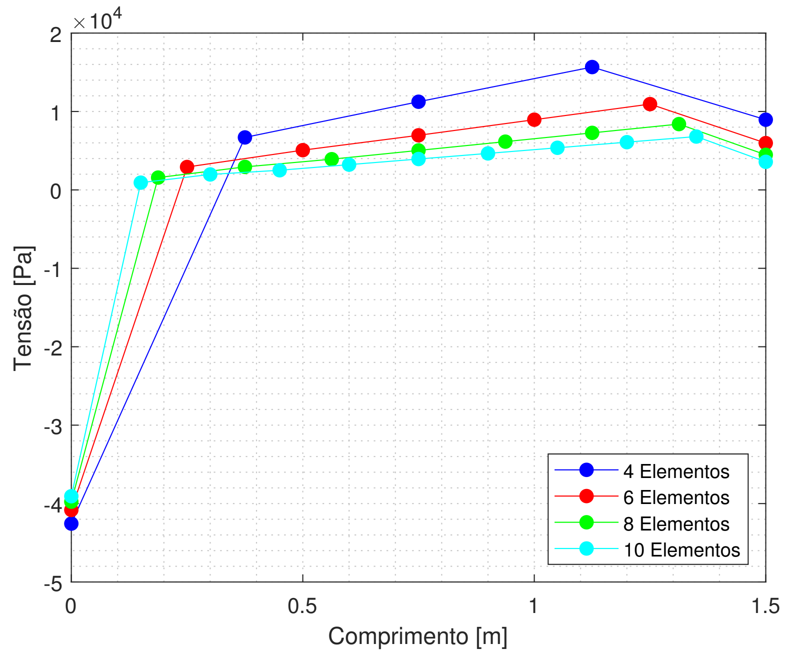


Figura 10: Comparativo de tensões na barra

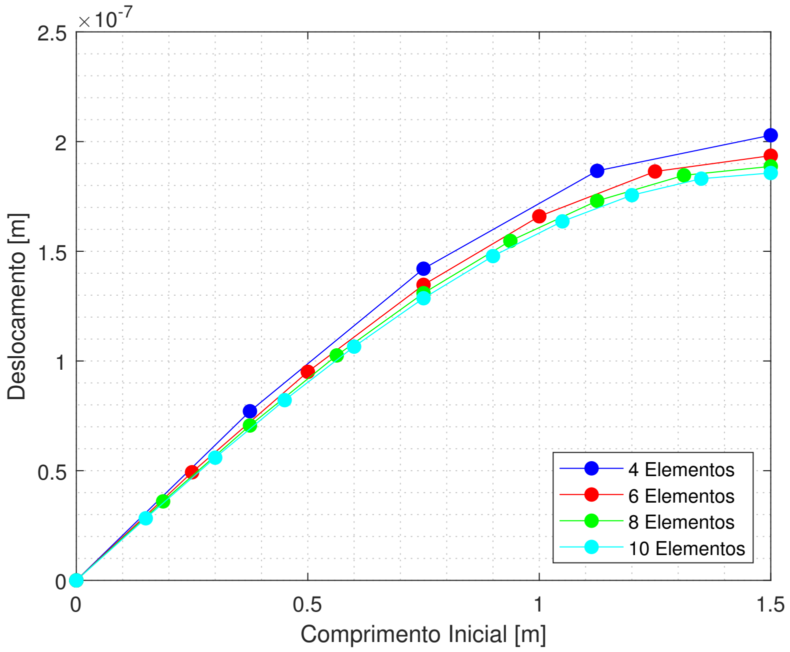


Figura 11: Comparativo de deslocamentos dos nós