Trabalho de Implementação Computacional 1; Análise de treliças planas

Utilizando matlab foi desenvolvido um script para calcular os deslocamentos e tensões das treliças apresentada na figura abaixo.

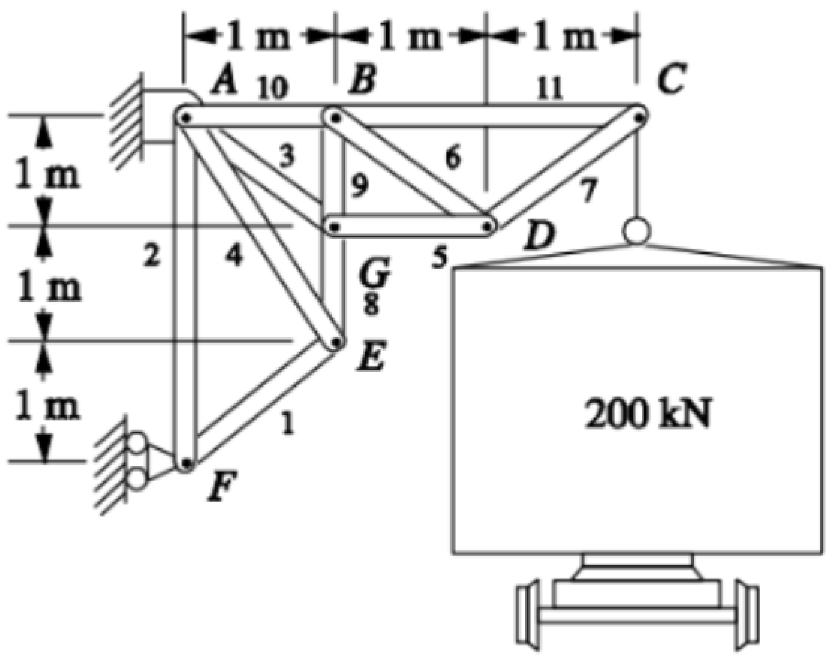

Figure 1: Treliças

1. Deslocamentos nodais da treliça

Utilizando o método de rigidez direto obtemos os deslocamentos nodais da treliça.

Nó	Deslocamento X (m)	Deslocamento Y (m)
A(1)	0.00000000	0.00000000
B(2)	0.00015748	-0.00076138
C(3)	0.00031496	-0.00425034
D(4)	-0.00128555	-0.00242712
E(5)	-0.00001251	-0.00044642
F(6)	0.00000000	-0.00023622
G(7)	-0.00112807	-0.00068264

2. Tensões das barras

Utilizando o método de rigidez direto obtemos as tensões das barras da treliça.

Barra	Tensão (MPa)
1	-22.27
2	15.75
3	-44.54
4	35.21
5	-31.50
6	22.27
7	-22.27
8	-47.24
9	-15.75
10	31.50
11	15.75

Onde os valores negativos indicam que a barras estão em compressão e os valores positivos indicam que as barras estão em tração.

3. Comparação com resultados disponiveis

Ao comparar os valores obtidos e mencionados nas tabelas acima com os disponíveis no link, https://drive.google.com/file/d/1uwsdMndz3jo5VpDi6VDPcPgXZZUvXpn-/view, percebemos que os valores são iguais, indicando que o script desenvolvido está correto.

4. Grafico com configuração deformada da treliça

Para comparação visual da deformação da treliça, foi plotado um gráfico com a configuração inicial e a configuração deformada da treliça.

5. Codigo computacional implementado

Segue abaixo o codigo implementado em matlab. A priori o codigo define os valores das constantes, como o modulo de elasticidade, a area da seção transversal, os angulos e os comprimentos iniciais das barras.

Em seguida é especifico a matriz de equivalencia, que relaciona os nós da treliça com as barras.

O codigo então calcula a matriz de rigidez de cada barra e a soma na matriz de rigidez global.

Em seguida são definidas as condições de contorno, onde são aplicadas as forças e os deslocamentos conhecidos.

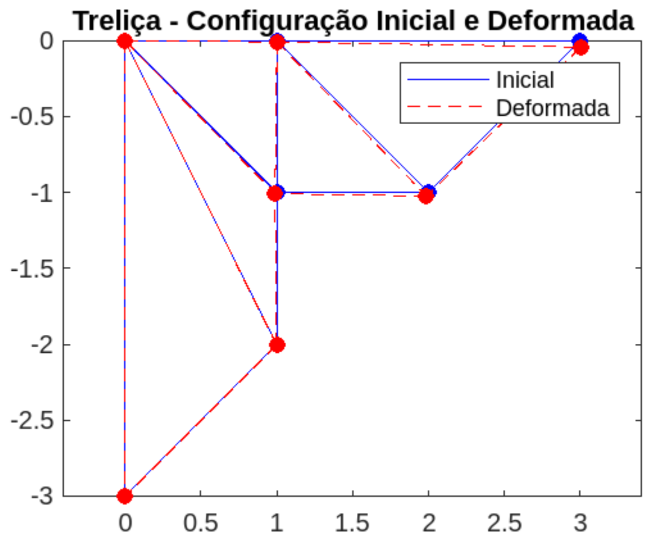

Figure 2: alt text

A matriz de rigidez global é reduzida para os graus de liberdade desconhecidos e os deslocamentos são calculados.

Por fim, são calculadas as tensões de cada barra.

Codigo também disponivel em:

https://github.com/freedxmgxd/metodos-computacionais-para-analise-estrutural/blob/main/Trabalho_1/main.m

```
clear all
close all
clc

E = 200e9;
A = 0.0127;

beta=[45 90 135 116.56505 0 135 45 90 90 0 0]*pi/180;
L = [sqrt(2) 3 sqrt(2) sqrt(5) 1 sqrt(2) sqrt(2) 1 1 1 2];

K = zeros(14,14);

% Matrix de equivalencia

ME = [11 12 9 10
    11 12 1 2
    13 14 1 2
```

```
    9 10 1 2
    13 14 7 8
    7 8 3 4
    7 8 5 6
    9 10 13 14
    13 14 3 4
    1 2 3 4
    3 4 5 6];

for n=1:11
    k = A*E/L(n);
    c = cos(beta(n));
    s = sin(beta(n));
    ke= k * [c*c c*s -c*c -c*s
        c*s s*s -c*s -s*s
        -c*c -c*s c*c c*s
        -c*s -s*s c*s s*s];
    disp(ke);
    for i=1:4
        for j=1:4
            c=ke(i,j);
            I=ME(n,i);
            J=ME(n,j);
            K(I,J) = K(I,J) + c;
        end
    end
end

disp(K);

% Condições de contorno

% Forças
R = zeros(14,1);
R(6) = -200000;

% D1 = D2 = D11 = 0
D0 = [1 2 11];

K_reduzido = K;
R_reduzido = R;

K_reduzido(D0,:) = [];
K_reduzido(:,D0) = [];
R_reduzido(D0,:) = [];
```

```
D_reduzido = inv(K_reduzido)*R_reduzido;

% Reinserindo os valores de D

D = zeros(14,1);
indices_restantes = setdiff(1:14, D0);
D(indices_restantes) = D_reduzido;

disp(D);

tensoes = zeros(11,1);

for i= 1:11
  ui = D(ME(i,1));
  vi = D(ME(i,2));
  uj = D(ME(i,3));
  vj = D(ME(i,4));
  c = cos(beta(i));
  s = sin(beta(i));

  deformation = (uj - ui)*c + (vj - vi)*s;
  tensoes(i) = (E/L(i))*deformation;
end

disp(tensoes);


% Coordenadas dos nós
x = [0, 1, 3,  2,  1,  0,  1];
y = [0, 0, 0, -1, -2, -3, -1];

% Deslocamentos nodais
Ux = D(1:2:end);
Uy = D(2:2:end);

% Coordenadas dos nós deformados
fator_escala = 10; % Exagera a deformação
x_deformado = x + fator_escala * Ux';
y_deformado = y + fator_escala * Uy';

% Conectividade das barras
barras = [1 2; 1 7; 1 5; 1 6; 2 3; 2 4; 2 7; 3 4; 4 7; 5 6; 5 7;]

% Plot da treliça na mesma figura
figure;
```

```
% Plot da configuração inicial
for i = 1:size(barras, 1)
  no1 = barras(i, 1);
  no2 = barras(i, 2);
  h_inicial = plot([x(no1), x(no2)], [y(no1), y(no2)], 'b-'); % Linhas azuis para a configu
  hold on;
end
h_inicial_nos = plot(x, y, 'bo', 'MarkerFaceColor', 'b'); % Nós azuis

% Plot da configuração deformada
for i = 1:size(barras, 1)
  no1 = barras(i, 1);
  no2 = barras(i, 2);
  h_deformada = plot([x_deformado(no1), x_deformado(no2)], [y_deformado(no1), y_deformado(no
end
h_deformada_nos = plot(x_deformado, y_deformado, 'ro', 'MarkerFaceColor', 'r'); % Nós verme

title('Treliça - Configuração Inicial e Deformada');
axis equal;
legend([h_inicial, h_deformada], 'Inicial', 'Deformada');
```