

4º Trabalho: Interface gráfica do programa Cross

Desenho da configuração deformada e do diagrama de momentos fletores no programa do processo de Cross para vigas contínuas em MATLAB

Data da entrega: 20/out/2023

Este trabalho é continuação do programa que implementa o processo de Cross para viga contínuas com cargas uniformemente distribuídas em cada vão. O objetivo do trabalho é adicionar dois *canvases* e implementar as funções de desenho da configuração deformada e do diagrama de momentos fletores do modelo da viga contínua com apoios e cargas.

Complemente as classes `CrossDraw` e `CrossMember` do programa em MATLAB fornecido na homepage da disciplina no Moodle/CCEAD (procure por quarto trabalho):
https://www.tecgraf.puc-rio.br/ftp_pub/lfm/civ2801-232-trab4.zip.

Uma versão pré-compilada (pcode) da solução do trabalho também é fornecida no site da disciplina: https://www.tecgraf.puc-rio.br/ftp_pub/lfm/civ2801-232-trab4_pcode.zip.

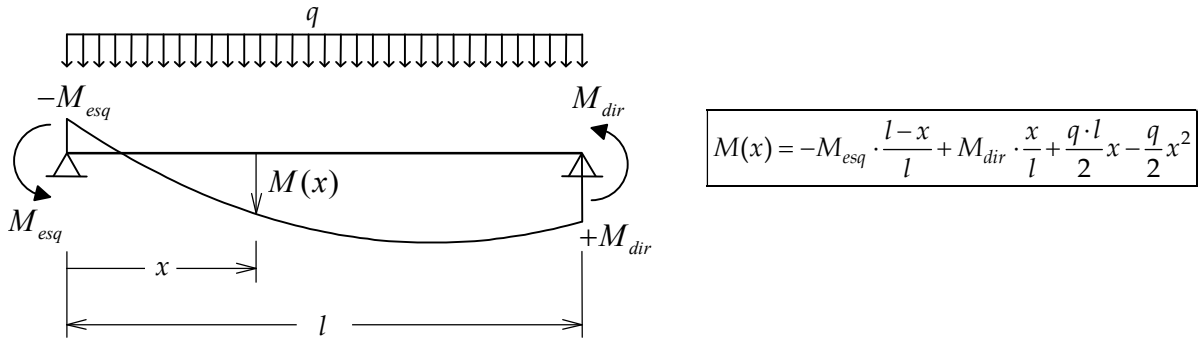
No link <https://www.tecgraf.puc-rio.br/~lfm/compgraf-232/cross03/main.html> tem uma documentação do código do programa do trabalho.

Especificação

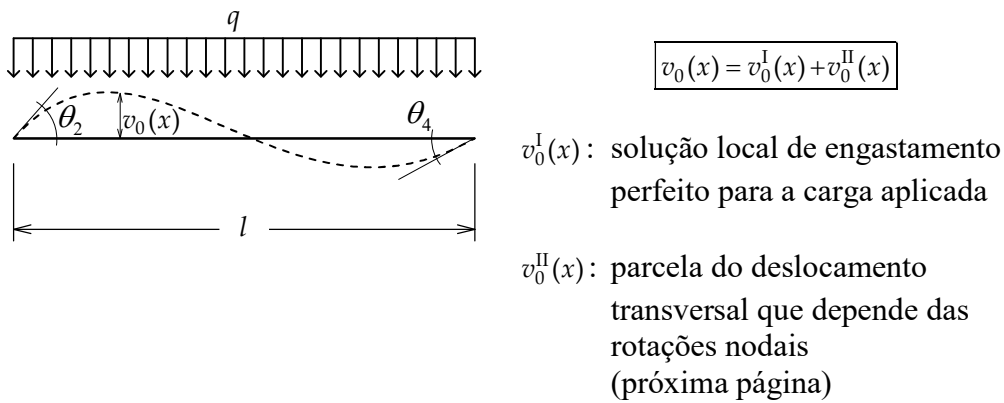
1. Modifique pelo aplicativo *App Designer* o arquivo `CrossGUI.mlapp` do terceiro trabalho acrescentando dois painéis, cada um com um *canvas* (eixo): um para a configuração deformada e outro para o diagrama de momentos fletores.
2. Os códigos implementados nos trabalhos anteriores devem ser copiados para os arquivos correspondentes deste trabalho. O arquivo `CrossSolver.m` com a solução do primeiro trabalho deve ser aproveitado. O arquivo `CrossControl.m` foi modificado em relação ao terceiro trabalho. Entretanto, para completá-lo basta copiar as soluções adotadas no segundo trabalho e no terceiro trabalho para este arquivo.
3. Os métodos públicos `CrossDraw.deformedConfig` e `CrossDraw.bendingMomDiagram` da classe `CrossDraw` devem ser implementados.
4. Os métodos privados `CrossMember.fixedEndUniformLoadDispl` e `CrossMember.displShapeFcnVector` e os métodos públicos `CrossMember.bendingMoment` e `CrossMember.momentRoots` da classe `CrossMember` devem ser implementados.
5. Enviar via e-mail os arquivos `CrossGUI.mlapp`, `CrossSolver.m`, `CrossDraw.m` e `CrossMember.m` com a solução do trabalho.
6. Enviar via e-mail um arquivo texto em PDF mostrando apenas as funções que foram complementadas nos arquivos `CrossDraw.m` e `CrossMember.m`.

Expressões para desenho do diagrama de momentos fletores e da configuração deformada

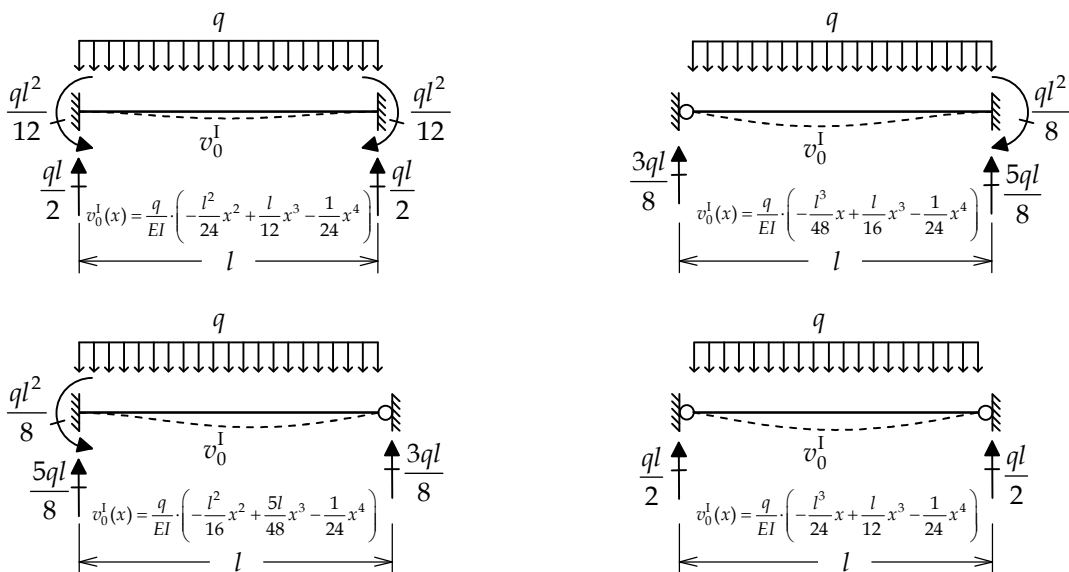
Para o desenho do diagrama de momentos fletores em cada vão da viga contínua, a seguinte expressão deve ser utilizada:



Para o desenho da elástica (configuração deformada) em cada vão da viga contínua, as seguintes expressões para o deslocamento transversal $v_0(x)$ devem ser utilizadas:



Reações de engastamento perfeito e correspondentes elásticas $v_0^I(x)$ de barra com e sem articulação nas extremidades para força transversal uniformemente distribuída:



A parcela $v_0^{\text{II}}(x)$ do deslocamento transversal que depende das rotações nas extremidades do vão é obtida utilizando as seguintes expressões para as **funções de forma** $N_2^v(x)$ e $N_4^v(x)$:

$$v_0^{\text{II}}(x) = \theta_2 \cdot N_2^v(x) + \theta_4 \cdot N_4^v(x)$$

Barra sem articulação

$$N_2^v(x) = x - \frac{2}{l}x^2 + \frac{1}{l^2}x^3 \quad N_4^v(x) = -\frac{1}{l}x^2 + \frac{1}{l^2}x^3$$

Barra com articulação na extremidade inicial

$$N_2^v(x) = 0 \quad N_4^v(x) = -\frac{1}{2}x + \frac{1}{2l^2}x^3$$

Barra com articulação na extremidade final

$$N_2^v(x) = x - \frac{3}{2l}x^2 + \frac{1}{2l^2}x^3 \quad N_4^v(x) = 0$$

Barra com duas extremidades articuladas

$$N_2^v(x) = 0 \quad N_4^v(x) = 0$$

Desenho da configuração deformada e do diagrama de momentos fletores utilizando expressões vetoriais (evitando loops)

Para criar as linhas poligonais que representam a configuração deformada e o diagrama de momentos fletores de cada vão da viga contínua, utilize o paradigma SIMD (*Single Instruction Multiple Data*), conforme ilustrado abaixo.

```

% Cria um vetor de pontos de uma parábola do segundo grau
% usando loop convencional (for)

clear;
clc;

x = -5:0.2:25;
n = size(x,2);
pt = zeros(1,n);

for i = 1:n
    pt(i) = -x(i)^2/5 + 4*x(i) + 5;
end

figure
plot(x, pt, 'Color', [0 0 1]);

% Cria um vetor de pontos de uma parábola do segundo grau
% usando o paradigma SIMD (Single Instruction Multiple Data)

clear;
clc;

x = -5:0.2:25;
pt = -x.^2/5 + 4*x + 5;

figure
plot(x, pt, 'Color', [0 0 1]);
    
```

