

# CIV 2118 – Introdução ao Método dos Elementos Finitos

## 2º Semestre – 2009

### Trab1: Método da Rigidez Direta

#### Treliças planas

O objetivo do trabalho é complementar um programa em MATLAB (mostrado abaixo) para análise de treliças planas. O programa considera que a solicitação externa da treliça é constituída de forças aplicadas nos nós e recalques de apoio.

Para a complementação do programa é necessário o entendimento do programa incompleto fornecido. Esse entendimento faz parte do trabalho. Os trechos do programa que devem ser complementados são identificados pela linha de comentário do tipo:

```
%%% COMPLETE AQUI XX %%%
```

O trabalho deve ser entregue na forma de um relatório, que descreve os procedimentos que foram utilizados para complementar o programa. O código MATLAB completo deve ser enviado via e-mail para os professores. Tal relatório deve conter, para cada trecho complementado, a(s) linha(s) de código MATLAB introduzidas. Por exemplo,

```
%%% COMPLETE AQUI 01 %%%
```

```
TRECHO DE CÓDIGO MATLAB
```

```
%%% COMPLETE AQUI 02 %%%
```

```
TRECHO DE CÓDIGO MATLAB
```

...

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% Pontifícia Universidade Católica do Rio de Janeiro - PUC-Rio
%% Disciplina: CIV 2118 - Introdução ao Método dos Elementos Finitos
%% Profa. Deane Roehl
%% Prof. Luiz Fernando Martha
%% Programa para Analise de Treliças Planas
%% pelo Método da Rigidez Direta
%% Versão 1.0 - 31/ago/2009
%% Versão original desenvolvida por João da Costa Pantoja
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;
fprintf(1,'PROGRAMA PARA ANÁLISE DE TRELIÇAS PLANAS\n\n');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1 ETAPA - LEITURA DE DADOS VIA ARQUIVO DE ENTRADA
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fprintf(1,'ARQUIVO DE ENTRADA: TrelicaTeste.dat\n\n');
fid = fopen('TrelicaTeste.dat','rt');
% Leitura da primeira linha do arquivo com 4 posições
dados = fscanf(fid,'%d',4);
nnos = dados(1);
nelm = dados(2);
nnr = dados(3);
nnc = dados(4);
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Impressão dos dados de entrada e dados globais
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(1, 'DADOS GLOBAIS DA ESTRUTURA\n\n');
fprintf(1, 'Número de Nós:\n');
fprintf(1, 'nnos = %d\n', nnos);
nen = 2;
fprintf(1, 'Número de Nós por Elemento:\n');
fprintf(1, 'nen = %d\n', nen);
ndof = 2;
fprintf(1, 'Número de Graus de Liberdade por Nó:\n');
fprintf(1, 'ndof = %d\n', ndof);
ngl = ndof*nnos;
fprintf(1, 'Número Total de Graus de Liberdade da Estrutura:\n');
fprintf(1, 'ngl = %d\n', ngl);
fprintf(1, 'Número de Elementos da Treliza:\n');
fprintf(1, 'nelm = %d\n', nelm);
fprintf(1, 'Número de Nós com Restrições de Apoio:\n');
fprintf(1, 'nnr = %d\n', nnr);
fprintf(1, 'Número de Nós Carregados:\n');
fprintf(1, 'nnc = %d\n\n', nnc);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      Leitura de uma matriz coord(k,n), sendo:
%      n - Número do Nó (n = 1, ..., nnos)
%      coord(1,n) - Coordenada x
%      coord(2,n) - Coordenada y
%      Cria um vetor para cada variável: x,y
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
coord = fscanf(fid, '%10e', [2, nnos]);
x = coord(1, :);
y = coord(2, :);

% % % Impressão das Coodenadas Nodais
fprintf(1, 'NÓS E RESPECTIVAS COORDENADAS NODAIS\n\n');
fprintf(1, 'Nó      Coord_x      Coord_y\n\n');
for n = 1:nnos
    fprintf(1, ' %d      %8.3f      %8.3f\n', ...
            n, x(n), y(n));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      Leitura de uma matriz restr(k,nn), sendo:
%      nn - Número de nós com restrições (nn = 1, ..., nnc)
%      restr(1,nn) - Número do nó com restrição
%      restr(2,nn) - Restrição na direção x (0 => livre; 1 => fixo)
%      restr(3,nn) - Restrição na direção y (0 => livre; 1 => fixo)
%      restr(4,nn) - Valor do recalque na direção da restrição x
%      restr(5,nn) - Valor do recalque na direção da restrição y
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
restr = fscanf(fid, '%10e', [5, nnr]);

% % % Impressão dos Nós Restritos e suas Restrições Nodais
fprintf(1, '\n\n');
fprintf(1, 'NÓS RESTRITOS E RESPECTIVAS RESTRIÇÕES NODAIS\n\n');
fprintf(1, 'Nó  RESTR_X  RESTR_Y  RECALQUE_X  RECALQUE_Y\n\n');
for nn = 1:nnr
    fprintf(1, ' %d      %d      %d      %8.2e      %8.2e\n', ...
            restr(1, nn), restr(2, nn), restr(3, nn), restr(4, nn), restr(5, nn));
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Definição da matriz GL com os números das equações dos GLs nodais.
% Inicialmente esta matriz contém as informações das restrições de apoio:
% se GL(k,n) = 0 o GL k do nó n é livre;
% se GL(k,n) = 1 o GL k do nó n está restringido
% Contabiliza o número de GLs com restrição de apoio (nglf).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
GL = zeros(ndof,nnos);
nglf = 0;
for nn = 1:nnr
  n = restr(1,nn);
  for k = 1:ndof
    if(restr(k+1,nn) == 1)
      nglf = nglf + 1;
      GL(k,n) = 1;
    end
  end
end

fprintf(1, '\n\n');
fprintf(1, 'Número Total de Graus de Liberdade Com Restrições de Apoio:\n');
fprintf(1, 'nglf = %d\n', nglf);
ngll = ngl - nglf;
fprintf(1, 'Número Total de Graus de Liberdade Livres:\n');
fprintf(1, 'ngll = %d\n', ngll);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Preenchimento da matriz GL com os números das equações dos GLs nodais
% de tal maneira que os graus de liberdade livres são numerados
% inicialmente e depois os graus de liberdade fixos.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
countl = 0;
countf = ngll;
for n = 1:nnos
  for k = 1:ndof
    if(GL(k,n) == 0)
      %%% COMPLETE AQUI 01 %%%
    else
      %%% COMPLETE AQUI 02 %%%
    end
  end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      Leitura de uma matriz prop(k,e), sendo:
%      e - Número do Elemento(e = 1, ..., nelm)
%      prop(1,e) - Nó1 da Barra (Incidência Nodal)
%      prop(2,e) - Nó2 da Barra (Incidência Nodal)
%      prop(3,e) - Área da Seção Transversal do Elemento e
%      prop(4,e) - Módulo de Elasticidade do Elemento e
%      Cria um vetor para cada variável: no1, no2, area e mod
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
prop = fscanf(fid, '%10e', [4, nelm]);
no1 = prop(1, :);
no2 = prop(2, :);
area = prop(3, :);
mod = prop(4, :);

% % % Impressão da Incidência Nodal e Propriedades de Material por Barra
fprintf(1, '\n\n')
fprintf(1, 'CONNECTIVIDADES E PROPRIEDADES DO MATERIAL\n\n');
fprintf(1, 'Elemento  Nó_1  Nó_2  Área  Mód. de Elast.\n\n');
for e = 1:nelm
  fprintf(1, '  %d      %d      %d      %2.5f      %8.2e\n', ...

```

```

                e, no1(e), no2(e), area(e), mod(e));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      Leitura de uma matriz forcas(k,nn), sendo:
%      nn - Número de nós carregados (nn = 1,...,nnc)
%      forcas(1,nn)- Número do nó carregado
%      forcas(2,nn)- Força na direção x (positivo esq/direita)
%      forcas(3,nn)- Força na direção y (positivo p/ cima)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
forcas = fscanf(fid, '%10e', [3,nnc]);

% % % Impressão do Vetor de Forças Aplicadas
fprintf(1, '\n\n');
fprintf(1, 'VETOR DAS FORÇAS APLICADAS NA TRELIÇA\n\n');
fprintf(1, 'Nó      Direção_x      Direção_y\n');
for nn = 1:nnc
    fprintf(1, ' %d      %6.2f      %6.2f\n', ...
            forcas(1,nn), forcas(2,nn), forcas(3,nn));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Montagem do Vetor de Forças Aplicadas
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
FG = zeros(ngl,1);
for nn = 1:nnc
    n = forcas(1,nn);
    for k = 1:ndof
        if(GL(k,n) <= ngl1)
            FG(GL(k,n)) = forcas(k+1,nn);
        end
    end
end

% Fecha o arquivo de leitura de dados
st = fclose(fid);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      2 ETAPA - PLOTA A ESTRUTURA NA SUA CONFIGURAÇÃO INICIAL      %
%                                                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fprintf(1, '\n\n');
% Desenhando a Estrutura na sua Configuração Inicial
fprintf(1, 'DESENHANDO A CONFIGURAÇÃO INDEFORMADA DA ESTRUTURA\n\n');
fprintf(1, '(VER FIGURA 1 NA TELA)\n\n');
figure(1);
clf;

for e = 1:nelm
    line([x(no1(e)) x(no2(e))], [y(no1(e)) y(no2(e))]);
end
axis equal;
axis off;
title('Modelo Estrutural da Treliza');
for e = 1:nelm
    text((x(no1(e))+x(no2(e)))/2, (y(no1(e))+y(no2(e)))/2, num2str(e));
end
for n = 1:nnos
    texto = text(x(n), y(n), num2str(n));
    set(texto, 'Color', 'red');
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           3 ETAPA - OBTENÇÃO DA MATRIZ DE RIGIDEZ GLOBAL           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fprintf(1,'MATRIZ DE RIGIDEZ GLOBAL DA ESTRUTURA\n');
fprintf(1,'           MÉTODO DA RIGIDEZ DIRETA\n\n');

% Inicializando a Matriz de Rigidez global
KG = zeros(ngl,ngl);

% Inicializando vetor de espalhamento que é reutilizado para cada elemento
gle = zeros(nen*ndof,1);

% Loop nos elementos para montagem da Matriz de Rigidez Global
for e = 1:nelm

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Montagem da Matriz de Rigidez de cada Elemento
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Projeção da barra nos eixos globais
dx = x(no2(e)) - x(no1(e));
dy = y(no2(e)) - y(no1(e));
% Cálculo do comprimento do elemento
L = sqrt(dx^2 + dy^2);
% Cálculo dos cossenos diretores
cx = dx/L;
cy = dy/L;
% Definição da Matriz de Rotação
R = [ cx  cy  0.  0.;
      -cy  cx  0.  0.;
        0.  0.  cx  cy;
        0.  0. -cy  cx];
% Definição da Matriz de Rigidez do elemento de treliça kel (eixos locais)
E = mod(e);
A = area(e);
kel = [ E*A/L  0. -E*A/L  0.;
        0.  0.  0.  0.;
       -E*A/L  0.  E*A/L  0.;
        0.  0.  0.  0.];
% Cálculo da Matriz de Rigidez do elemento no Sistema Global
%%% COMPLETE AQUI 03 %%%

% Montagem do vetor gle com o n° dos GLs do elemento
gle(1) = GL(1,no1(e));
gle(2) = GL(2,no1(e));
gle(3) = GL(1,no2(e));
gle(4) = GL(2,no2(e));
% Espalhamento na matriz de rigidez global KG
%%% COMPLETE AQUI 04 %%%

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Imposição das restrições de apoio na matriz de rigidez global
% Método utilizado: adição de uma mola com rigidez grande (10000 vezes
% maior valor da diagonal da matriz de rigidez global) e aplicação de
% uma força com valor igual da kigidez (grande) da mola vezes o valor
% do recalque correspondente.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
KGmax = 0.0;
for i = 1:ngl
    if(KG(i,i) > KGmax)

```

```

    KGmax = KG(i,i);
end
end
KGmax = KGmax * 10000.0;

for nn = 1:nnr
    for k = 1:ndof
        n = restr(1,nn);
        if(GL(k,n) > ngl1)
%%% COMPLETE AQUI 05 %%%
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           4 ETAPA - RESOLUÇÃO DO SISTEMA DE EQUAÇÕES DE EQUILÍBRIO           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
UG = KG \ FG;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plota a estrutura deformada
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xd(nnos) = 0.0;
yd(nnos) = 0.0;
escala = input('Digite a escala a ser utilizada(500):');
if (isempty(escala) == 1)
    escala = 500;
end
for n = 1:nnos
    i = GL(1,n);
    xd(n) = x(n) + escala*UG(i);
    i = GL(2,n);
    yd(n) = y(n) + escala*UG(i);
end
fprintf('%s\n','Desenhando a estrutura deformada...');
figure(2);
clf;
for e = 1:nelm
    linha = line([x(no1(e)) x(no2(e))],[y(no1(e)) y(no2(e))]);
    set(linha,'Color','blue')
    set(linha,'LineStyle','--')
    linha = line([xd(no1(e)) xd(no2(e))],[yd(no1(e)) yd(no2(e))]);
    set(linha,'Color','red')
end
axis equal;
axis off;
title('Configuração Deformada da Estrutura');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           5 ETAPA - OBTENÇÃO DOS ESFORÇOS NOS ELEMENTOS E REAÇÕES DE APOIO           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Def. vetor dos deslocamentos locais ueg paralelos aos eixos globais
ueg = zeros(nen*ndof,1);
% Def. vetor dos deslocamentos locais uel paralelos aos eixos locais
uel = zeros(nen*ndof,1);
% Def. vetor do esforços locais feg paralelos aos eixos globais
feg = zeros(nen*ndof,1);
% Def. matrix do esforços locais fel paralelos aos eixos locais
fel = zeros(nen*ndof,nelm);

```

```

% Inicializa termos do vetor das forças globais da reações de apoio
for i = ngl1+1:ngl
    FG(i) = 0.0;
end

% Loop nos elementos para obtenção dos esforços nos elementos e reações
for e = 1:nelm
    % Projeção da barra nos eixos globais
    dx = x(no2(e)) - x(no1(e));
    dy = y(no2(e)) - y(no1(e));
    % Cálculo do comprimento do elemento
    L = sqrt(dx^2 + dy^2);
    % Cálculo dos cossenos diretores
    cx = dx/L;
    cy = dy/L;
    % Definição da Matriz de Rotação
    R = [ cx  cy  0.  0.;
         -cy  cx  0.  0.;
          0.  0.  cx  cy;
          0.  0. -cy  cx];
    % Definição da Matriz de Rigidez do elemento de treliça kel (eixos locais)
    E = mod(e);
    A = area(e);
    kel = [ E*A/L  0. -E*A/L  0.;
            0.  0.  0.  0.;
           -E*A/L  0.  E*A/L  0.;
            0.  0.  0.  0.];
    % Montagem do vetor gle com o n° dos GLs do elemento
    gle(1) = GL(1,no1(e));
    gle(2) = GL(2,no1(e));
    gle(3) = GL(1,no2(e));
    gle(4) = GL(2,no2(e));
    % Obtenção do vetor dos deslocamentos locais ueg (eixos globais)
    for jj = 1:nen*ndof
        ueg(jj) = UG(gle(jj));
    end
    % Cálculo do vetor dos deslocamentos locais uel paralelos aos eixos locais
    %%% COMPLETE AQUI 06 %%%
    % Cálculo dos esforços locais fel paralelos aos eixos locais
    fel(:,e) = kel*uel;
    % Cálculo dos esforços locais feg paralelos aos eixos globais
    %%% COMPLETE AQUI 07 %%%
    % Considera contribuição do elemento nas reações de apoio
    % (usa as últimas posições do vetor F para armazenar reações)
    for jj = 1:nen*ndof
        if(gle(jj) > ngl1)
    %%% COMPLETE AQUI 08 %%%
        end
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           6 ETAPA - PLOTA O DIAGRAMA DE ESFORÇOS NORMAIS E           %
%           REAÇÕES DE APOIO                                           %
%                                                                           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf('%s\n', 'Desenhando o DEN...');
figure(3);
clf;
% Desenha os esforços normais
for e = 1:nelm
    linha = line([x(no1(e)) x(no2(e))], [y(no1(e)) y(no2(e))]);
    set(linha, 'Color', 'blue')
end

```

```

texto = text((x(no1(e))+x(no2(e)))/2, (y(no1(e))+y(no2(e)))/2, ...
            num2str(-fel(1,e),4));
set(texto, 'Color', 'red')
end
% Desenha as reações de apoio
for nn = 1:nnr
    n = restr(1,nn);
    if(GL(1,n) > ngl1)
        valorx = num2str(FG(GL(1,n)));
    else
        valorx = strcat('none');
    end
    if(GL(2,n) > ngl1)
        valory = num2str(FG(GL(2,n)));
    else
        valory = strcat('none');
    end
    valor = strcat('(', valorx, ', ', valory, ')');
    texto = text(x(n), y(n), valor);
    set(texto, 'Color', 'black')
end
axis equal;
axis off;
title('Diagrama de Esforços Normais e Reações de Apoio');

```

## Arquivo de dados para teste

```

4 5 2 2
0.0      0.0
3.0      0.0
6.0      0.0
3.0      3.0
1 1 1 0.000 0.000
3 1 1 0.001 0.000
1 2 0.001 1.0E8
1 4 0.001 1.0E8
2 3 0.001 1.0E8
3 4 0.001 1.0E8
4 2 0.001 1.0E8
2 0 -10.0
4 5.0 0.0
    
```

