RELATÓRIO DE ESTÁGIO SUPERVISIONADO

TÍTULO: ESTUDO E APLICAÇÃO DE ELEMENTOS FINITOS UTILIZANDO O SOFTWARE MATLAB

ALUNA: SUSANE RIBEIRO
MATRÍCULA:29411461

ORIENTADOR: ÍTALO ATAÍDE NOTARO

LOCAL: UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

Biblioteca Setorial do CDSA. Maio de 2021.

Sumé - PB

ÍNDICE:

1. OBJETIVO

Adquirir conhecimentos e habilidade na utilização do soft MATLAB, aplicando o método de elementos finitos na resolução de problemas de treliça plana.

2. RESUMO

O método de elementos finitos é uma ferramenta de muitas utilidades e aplicações na resolução de problemas numéricos relacionados ao estudo de mecânica dos sólidos e resistência dos materiais. Trata-se de uma ferramenta muito importante para os engenheiros no projeto de estruturas.

Na Engenharia Agrícola possibilita muitas aplicações, principalmente nas áreas de mecânica agrícola, construções rurais e armazenamento. Com o aprimoramento e desenvolvimento de novos softwares, a utilização e aplicação dos métodos numéricos tornaram-se mais rápido, eficiente e prático. Neste trabalho foi dada ênfase à aplicação do soft MATLAB na elaboração de programa computacional para cálculo de esforços e deslocamentos em barras de uma treliça plana pelo método numérico de elementos finitos.

3. INTRODUÇÃO

O método dos elementos finitos é uma importante ferramenta computacional para executar cálculos que na prática seriam muito difíceis ou mesmo impossível. A sua concepção não é das mais recentes, data de 1943.

Até a década de 70 seu processamento só podia ser feito nos caros mainframes (antigos computadores) e, por isso, seu uso era restrito a grandes empresas, centros de pesquisa, instalações militares, dentre outros. Com a evolução da capacidade e a redução de custos dos computadores, as aplicações do método se expandiram e se tornaram cada vez mais precisas e sofisticadas.

De início era usado quase sempre no cálculo de estruturas de engenharia, mas atualmente é aplicado em diversas áreas, como transferência de calor, escoamento de fluidos, eletromagnetismo e muitas outras.

3.1. Concepção básica

Métodos computacionais de cálculo geralmente fazem a aproximação de um resultado, que teoricamente seria contínuo, em valores discretos.

A figura 1 dá exemplo de um resultado representado pela função contínua f(x) (curva contínua) e sua aproximação discreta, indicada pelos segmentos de retas. É evidente que a aproximação será tanto melhor quanto maior for o número de pontos discretos.

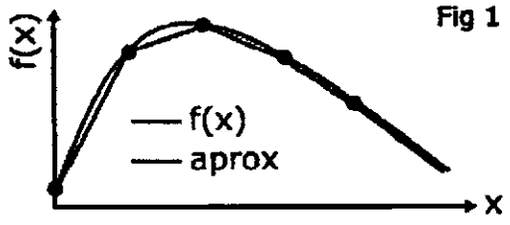

Fig 1

FIGURA 1

Um exemplo prático é dado pela figura 2:

Fig 2

FIGURA 2

Uma viga de seção retangular engastada em uma extremidade e com uma carga concentrada na outra.

Para quem conhece a teoria, é bastante fácil obter o resultado analítico e contínuo para, por exemplo, as tensões e deformações na mesma.

Mas se a viga tiver um furo redondo conforme figura 3?

Fig 3

FIGURA 3

Provavelmente ainda será possível uma solução analítica, mas esta será bem mais complexa.

É possível imaginar que, em muitos casos práticos, as formas geométricas não são tão simples como estas e as soluções analíticas são virtualmente impossíveis.

A figura 4 dá uma idéia básica do método dos elementos finitos:

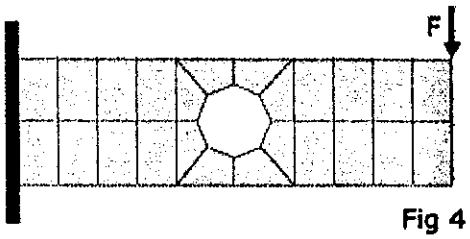

Fig 4

FIGURA 4

A viga é considerada como uma malha na qual as interseções (nós) adjacentes formam elementos (no caso, quadriláteros).

Aos elementos são atribuídas propriedades de deformação compatíveis com o material da viga e os cálculos se fazem por um sistema de equações lineares em forma de matrizes, proporcionando uma aproximação discreta, similar à figura 1.

Naturalmente, este exemplo é apenas ilustrativo. Na prática as malhas são bem mais finas, ou seja, o número de elementos é grande. Isto exige uma capacidade e velocidade de cálculo que torna indispensável o uso de computadores para a quase totalidade das aplicações.

Embora, conforme já foi dito, a abrangência e sofisticação do método tenham evoluído bastante, os princípios básicos permanecem. O exemplo abaixo mostra como o método funciona.

EXEMPLO:
Uma treliça isostática da mais simples possível. Com 3 barras e uma carga **W** conforme figura 5. Deseja-se obter o deslocamento das barras em função da carga **W** aplicada.

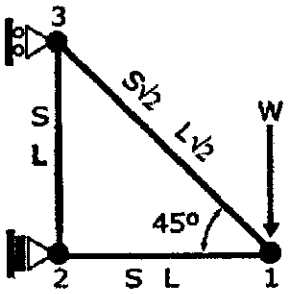

Fig 5

FIGURA 5

A lógica do problema sugere que os elementos devem ser as barras da treliça.

E, desde que as barras de uma treliça só são submetidas a esforços axiais pelas extremidades, não há qualquer necessidade de subdivisões adicionais.

Assim, a malha será formada por 3 elementos unidimensionais correspondentes às barras, e os nós serão as articulações da treliça, ou seja, elementos 1-2, 2-3 e 3-1 e nós 1, 2 e 3.

3.2. Matriz de rigidez

Para continuar a solução do problema anterior, é necessário introduzir este conceito, um dos fundamentais do método dos elementos finitos.

Matriz de rigidez nada é mais do que uma matriz que indica relações entre propriedades do elemento. No caso da barra de treliça, as relações entre forças aplicadas nas extremidades e os deslocamentos.

Conforme a figura 6, 1-2 a barra genérica da treliça, de comprimento **L**, que faz um ângulo a com a horizontal do sistema de coordenadas **xy** (que será denominado sistema global de coordenadas).

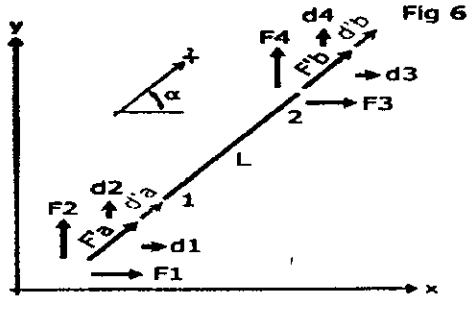

FIGURA 6

A direção **x'** será denominada sistema local de coordenadas para a barra.

Considera-se a barra em regime elástico conforme lei de Hooke. Assim, a relação entre uma força **F** aplicada no sentido longitudinal e o deslocamento **d**, também longitudinal, é dada por:

F = (SE/L) d ,onde: **S** - é a área da seção transversal,
E - o módulo de elasticidade e
L - o comprimento.

Nas extremidades são aplicadas as forças **F'a** e **F'b** e os deslocamentos são **d'a** e **d'b**, tudo em coordenadas locais.

Considerando-se a extremidade 2 fixa (deslocamento nulo), temos:

F'a = (SE/L) d'a e **F'b = - (SE/L) d'a** (o sinal negativo é para atender a condição de equilíbrio estático).

De forma análoga, com 1 fixo:

$$F'a = -(SE/L)\, d'b \quad e \quad F'b = (SE/L)\, d'b.$$

Combinando as equações, temos a forma genérica para deslocamentos em ambas extremidades:

$$F'a = (SE/L)\,(d'a - d'b)$$
$$F'b = (SE/L)\,(-d'a + d'b)$$

Em forma matricial;

$$\begin{bmatrix} F'a \\ F'a \end{bmatrix} = \begin{bmatrix} SE/L & -SE/L \\ -SE/L & SE/L \end{bmatrix} \begin{bmatrix} d'a \\ d'b \end{bmatrix}$$

e, assim, fica definida a matriz de rigidez k para o elemento. Ou se pode escrever **F' = k' d'** pois estão em coordenadas locais.

3.3. Transformação de coordenadas

Em uma treliça os elementos têm várias inclinações, isto é, existe mais de um sistema de coordenadas locais. Assim, é necessária a transformação para o sistema global.

Na figura 6, **F1-F2** e **F3-F4** são as componentes no sistema global de **F'a** e **F'b** respectivamente. E, chamando **c = cos a** e **s = sen a**, chega-se, por relações trigonométricas, à seguinte matriz de transformação:

$$\begin{bmatrix} F1 \\ F2 \\ F3 \\ F4 \end{bmatrix} = \begin{bmatrix} c & 0 \\ s & 0 \\ 0 & c \\ 0 & s \end{bmatrix} \begin{bmatrix} F'a \\ F'b \end{bmatrix}$$

ou **F = TF F'**. E, de forma similar, os deslocamentos **d1-d2** e **d3-d4** correspondem aos sistemas de coordenadas locais **d'a** e **d'b**. E com a seguinte matriz de transformação:

$$\begin{bmatrix} d'a \\ d'b \end{bmatrix} = \begin{bmatrix} c & s & 0 & 0 \\ 0 & 0 & c & s \end{bmatrix} \begin{bmatrix} d1 \\ d2 \\ d3 \\ d4 \end{bmatrix}$$

ou **d' = Td d.**

Combinando as igualdades: **F = TF F' = TF k' d' = TF k' Td d** ou **F = kd.**
A matriz **k = TF k' Td** tem a forma:

$$(SE/L)\begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix}$$

e é denominada matriz de rigidez do elemento em coordenadas globais.

3.4. Matriz de rigidez para a treliça do exemplo

Na figura 7 abaixo, esta indicada o esquema das forças e deslocamentos dos elementos da treliça.

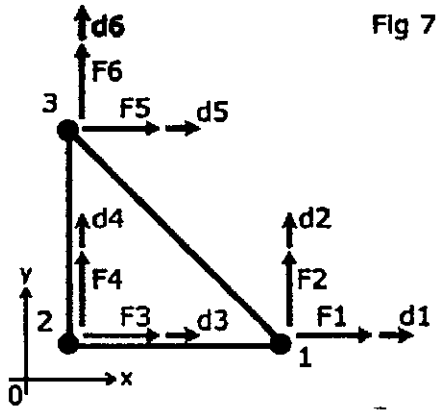

Fig 7

FIGURA 7

O sistema global de coordenadas **xy** está indicado em uma posição deslocada por uma questão de clareza. Na realidade, é considerado que a origem **0** coincide com o nó 2 da malha.

Os ângulos de inclinação em relação ao eixo **x** para uso na matriz do item anterior são: 0° para o elemento 1-2, 90° para 2-3 e 135° para 3-1.

O elemento 3-1 (figura 7) é suposto ter uma área $S\sqrt{2}$, onde **S** é a área dos demais. Isto não é fato comum, apenas para facilitar o cálculo.

Matriz de rigidez para elemento 1-2:

F1		1	0	-1	0	d1
F2	= SE/L	0	0	0	0	d2
F3		-1	0	1	0	d3
F4		0	0	0	0	d4

Matriz de rigidez para elemento 2-3:

F3		0	0	0	0	d3
F4	= SE/L	0	1	0	-1	d4
F5		0	0	0	0	d5
F6		0	-1	0	1	d6

Matriz de rigidez para elemento 3-1:

F1		1	-1	-1	1	d1
F2	= SE/	-1	1	1	-1	d2
F5	2L	-1	1	1	-1	d5
F6		1	-1	-1	1	d6

A matriz de rigidez global é formada pela composição das matrizes de cada elemento.

Onde houver superposição de células, aplica-se a soma.

Matriz de rigidez global:

F1		3/2	-1/2	-1	0	-1/2	1/2	d1
F2		-1/2	1/2	0	0	1/2	-1/2	d2
F3	= SE/	-1	0	1	0	0	0	d3
F4	L	0	0	0	1	0	-1	d4
F5		-1/2	1/2	0	0	1/2	-1/2	d5
F6		1/2	-1/2	0	-1	-1/2	3/2	d6

3.5. Condições de contorno

Alguns dados a mais são necessários, quando o problema não pode ser resolvido apenas pela matriz de rigidez na forma do item anterior.

Pelo tipo de apoio pode-se concluir que **d3 = d4 = d5 = 0.** E, considerando somente as forças externas, **F1 = 0**, **F2 = -W** e **F6 = 0**. Tais dados são chamados de condições de contorno e sempre ocorrerão na aplicação do método.

Substituindo na matriz anterior:

0		3/2	-1/2	-1	0	-1/2	1/2	d1
-W		-1/2	1/2	0	0	1/2	-1/2	d2
F3	= SE/L	-1	0	1	0	0	0	0
F4		0	0	0	1	0	-1	0
F5		-1/2	1/2	0	0	1/2	-1/2	0
0		1/2	-1/2	0	-1	-1/2	3/2	d6

Eliminando as linhas e colunas nulas:

$$\begin{bmatrix} 0 \\ W \\ 0 \end{bmatrix} = SE/L \begin{bmatrix} 3/2 & -1/2 & 1/2 \\ -1/2 & 1/2 & -1/2 \\ 1/2 & -1/2 & 3/2 \end{bmatrix} \begin{bmatrix} d1 \\ d2 \\ d6 \end{bmatrix}$$

Portanto, **d1 = -WL/SE, d2 = -4WL/SE e d6 = -WL/SE.**

Este exemplo mostra de forma genérica o funcionamento do método. Aplicações práticas são em geral bem mais complexas, com elementos bi ou tridimensionais e em grande número. Isto requer o uso de computadores e de programas específicos.

4. O SOFTWARE MATLAB

4.1. Introdução

O MATLAB é uma ferramenta software que pode funcionar como uma simples calculadora ou até como uma linguagem de programação científica (Fortran, C, etc.) para soluções de complicadas expressões algébricas. Entretanto, o MATLAB apresenta diversas vantagens em relação às calculadoras e linguagens de programação: simplicidade e uma interface gráfica bastante completa para visualização e análise dos resultados.

4.2. Características Básicas

O MATLAB é uma ferramenta software que tem por característica básica a simplicidade de utilização e uma poderosa interface gráfica. Como qualquer software ou linguagem de programação é necessário haver uma adaptação à ferramenta.

4.3. Área de Trabalho do MATLAB

A área de trabalho do MATLAB é onde ficam residentes os diversos comandos e valores de quaisquer variáveis que foram digitados na janela de comandos. Como aqueles comandos e variáveis estão residentes na área de trabalho do MATLAB, podem ser invocados sempre que for preciso ou desejado. Por exemplo, se quiser verificar o valor da variável **a** basta que se entre com o nome desta variável no prompt.

4.4. Formato dos Números

Como default, se um resultado é inteiro, o MATLAB mostra o número como inteiro. Igualmente, quando o resultado é real, o MATLAB mostra o número com 4 dígitos a direita do ponto decimal. Se os dígitos do resultado estiverem fora desta faixa, o MATLAB mostra o resultado usando a notação científica como numa calculadora

científica. Este default pode, entretanto, ser modificado usando-se o **Numeric Format** da pasta **general** na linha **Preferences** do item **Files** na barra de menus.

4.5. CONSTRUÇÃO DE VETORES

Nas construções das funções implementadas até agora, utilizou-se da construção de vetores. Agora, mostrar-se-á algumas outras formas de manipular vetores no MATLAB. Na tabela 1, tem-se um resumo das diversas formas de se construir um vetor no MATLAB.

Tabela 1- Construção de Vetores

x=[2 2*pi sqrt(2) 2-3j]	Cria um vetor **x** contendo os elementos especificados
x=primero : último	Cria um vetor **x** começando com o valor **primeiro**, incrementando-se de 1(um) em 1(um) até atingir o valor **último** ou o valor mais próximo possível de **último**
x=primeiro:incremento:último	Cria um vetor **x** começando com o valor **primeiro**, incrementando-se do valor **incremento** até atingir o valor **último** ou o valor mais próximo possível de último.

4.6. Arquivos Script (M-files)

Para resolver problemas simples, é cômodo e eficiente utilizar o MATLAB como se fosse uma calculadora, entrando-se com os comandos diretamente no prompt. Entretanto, à medida que o número de comandos aumenta, ou quando se deseja mudar o valor de uma ou mais variáveis e executar novamente os comandos, o uso do MATLAB simplesmente como calculadora torna-se ineficiente e tedioso. Nestes casos, o melhor é utilizar o MATLAB como uma linguagem de programação de alto nível, isto é, escrever um arquivo "script" ou M-files. Os M-files são os programas fontes do MATLAB e levam a extensão.**m**, por exemplo, **exemplo1.m**.

Para escrever um programa no MATLAB, escolha **File** na barra de menu. Dentro do menu **File** escolha **New** e selecione **M-file**. Abre-se, então, um editor de textos, onde se pode escrever os comandos do MATLAB.

4.7. Matrizes

Os vetores vistos até agora se tratam de vetores linha, pois, possuem apenas uma linha com várias colunas. Também se pode obter vetores coluna, isto é, vetores com apenas uma coluna e várias linhas:

Se um vetor passa a consistir de várias linhas e colunas, denominamo-lo de Matrizes:

Portanto, um vetor linha é um caso particular de uma matriz $1 \times N$, e um vetor coluna é um caso particular de matriz $N \times 1$.

4.7.1. Operações Escalares com Matrizes no MATLAB

Uma operação de adição, subtração, multiplicação e divisão de uma matriz com um valor escalar, é obtida simplesmente aplicado-se a respectiva operação em cada um dos elementos da matriz.

4.7.2. Operações entre Matrizes no MATLAB

As operações entre matrizes requerem que as mesmas tenham as mesmas dimensões, e as operações de adição, subtração, multiplicação e divisão são aplicadas elemento-por-elemento.

Observe que a operação g.*h é diferente da operação g*h. Enquanto que a primeira executa uma multiplicação elemento-por-elemento de duas matrizes, a segunda executa uma multiplicação entre duas matrizes.

4.8. Manipulação de Gráficos

Nesta secção mostrar-se-ão alguns comandos úteis para manipulação de gráficos. Pode-se adicionar curvas a um gráfico já plotado usando o comando hold.

Duas outras formas úteis de se construir gráficos são utilizando-se os comandos hist e stem. O Comando hist(y) desenha um histograma com 10 bins para os dados de um vetor y. hist(y,n) cria um histograma com n bins. hist(y,x), onde x é um vetor cria um histograma usando os bins especificados no vetor x. O exemplo seguinte ilustra o uso do hist.

Para a representação de seqüências discretas, é útil a plotagem de gráficos com stem. stem(y) plota os dados do vetor y. stem(x,y) plota os valores do vetor y dados por x. O exemplo seguinte ilustra o uso do stem.

4.9. Estilo de Linhas e Cores

Nos exemplos anteriores, utilizamos diferentes estilos de linhas e cores. Os estilos de linhas e as cores podem ser especificados nos comandos de plotagem como um argumento do tipo caracter string (entre apóstrofes – por exemplo, 'c'), consistindo de 1,2 ou 3 caracteres. A tabela 2 apresenta estes caracteres.

Tabela 2: Estilos de linhas e cores

Símbolo	Cor	Símbolo	Estilo de Linha
y	amarelo	.	pontos
m	magenta	o	círculos
c	cian	x	marcas x
r	vermelho	+	marcas mais
g	verde	*	marcas asterístico
b	azul	-	linha sólida
w	branco	:	linha pontilhada
k	preto	-.	linha com traço e ponto
		--	linha com segmentos de traço

4.10. Construção de Vetores

Nas construções das funções implementadas até agora, utilizou-se da construção de vetores. Agora, mostrar-se-á algumas outras formas de manipular vetores no MATLAB. Na tabela 3, tem-se um resumo das diversas formas de se construir um vetor no MATLAB.

Tabela 3 - Construção de Vetores

x=[2 2*pi sqrt(2) 2-3j]	Cria um vetor **x** contendo os elementos especificados
x=primero : último	Cria um vetor **x** começando com o valor **primeiro**, incrementando-se de 1(um) em 1(um) até atingir o valor **último** ou o valor mais próximo possível de **último**
x=primeiro: incremento: último	Cria um vetor **x** começando com o valor **primeiro**, incrementando-se do valor **incremento** até atingir o valor **último** ou o valor mais próximo possível de último.
x=linspace (primeiro, último, n).	Cria um vetor **x** começando com o valor **primeiro** e terminado-se no valor **último**, contendo **n** elementos.
x=logspace (primeiro, último, n).	Cria um vetor com os elementos espaçado logaritmicamente, começando-se com o valor $10^{primeiro}$ e terminando-se no valor $10^{último}$, contendo **n** elementos.

5. PROGRAMA COMPUTACIONAL

Desenvolvido utilizando elementos finitos através do software MATLAB para resolução de problemas de treliça plana.

Início do programa

```
clear

% Base de Dados - Formato do arquivo de entrada de dados

     %ngln=2;
     %nel=3;     % numero de elementos
     %nnos=3;
     %ncn=1;     %

% Matriz dos Nos
%           x  y res alfa  u   v
     % nos=[0  0  2   0    0   0;
     %      1  0  1  pi/4  0   0;
     %      0  1  0   0    0   0];

     %ee=250000;
     %aa=0.1;

% Matriz dos Elementos
%         no1 no2  E   A    fx  fy
     % el=[1   3   ee  aa   .5   .5;
     %     1   2   ee  aa   .5   .5;
     %     2   3   ee  aa   .5   .5];

% Matriz das cargas nodais
%         no  P1   P2
     %cn=[3   500  500];

% Meta Programa
% Alocaçao de memoria
Dadostrelicamodificada
kb=zeros(4,4);
fb=zeros(4,1);
ub=zeros(4,1);
reac=zeros(2,1);
r=zeros(4,1);


soma=0;
for i=1:nnos
   soma=soma+nos(i,3);     % Calcula o numero total de
                               restriçoes da treliça
end
```

```
tam=nnos*ngln-soma;          % Calcula o tamanho das matrizes
                               de rigidez e força
kg=zeros(tam,tam);           % Alocaçao da matriz de rigidez
fg=zeros(tam,1);             % % Alocaçao da matriz de força

% Inicio do Bloco

for i=1:nel
   no1=el(i,1);
   no2=el(i,2);
   e=el(i,3);
   a=el(i,4);
   fb(1)=el(i,5);
   fb(2)=el(i,6);
   x1=nos(no1,1);
   y1=nos(no1,2);
   x2=nos(no2,1);
   y2=nos(no2,2);
   rest1=nos(no1,3);
   rest2=nos(no2,3);
   alf1=nos(no1,4);
   alf2=nos(no2,4);
   dx=x2-x1;
   dy=y2-y1;
   tet=calcang(dx,dy);
   gam1=tet-alf1;
   gam2=tet-alf2;
   beta=[cos(gam1) sin(gam1) 0 0; 0 0 cos(gam2) sin(gam2)];
   lb=sqrt(dx*dx+dy*dy);
   ke=e*a*[1 -1; -1 1]/lb;
   kb=(beta')*ke*beta;
   fb(3:4)=fb(1:2);
   fb=fb*lb/2;
   r1=[cos(alf1) sin(alf1); -sin(alf1) cos(alf1)];
   r2=[cos(alf2) sin(alf2); -sin(alf2) cos(alf2)];
   fb(1:2)=r1*fb(1:2);
   fb(3:4)=r2*fb(3:4);

   soma=0;
   for j=1:no1-1
      soma=soma+nos(j,3);
   end
   pos1=(no1-1)*ngln-soma+1;

   soma=0;
   for j=1:no2-1
      soma=soma+nos(j,3);
```

```
end
pos2=(no2-1)*ngln-soma+1;

if rest1==0
   m=pos1+1;
   kg(pos1:m,pos1:m)=kg(pos1:m,pos1:m)+kb(1:2,1:2);
   fg(pos1:m)=fg(pos1:m)+fb(1:2);
   n=pos2+1;
   if rest2==0
      kg(pos1:m,pos2:n)=kg(pos1:m,pos2:n)+kb(1:2,3:4);
      kg(pos2:n,pos1:m)=kg(pos2:n,pos1:m)+kb(3:4,1:2);
   end
   if rest2==1
      kg(pos1:m,pos2)=kg(pos1:m,pos2)+kb(1:2,3);
      kg(pos2,pos1:m)=kg(pos2,pos1:m)+kb(3,1:2);
      u4=nos(no2,6);
      fg(pos1:m)=fg(pos1:m)-kb(1:2,4)*u4;
   end
   if rest2==2
      fg(pos1:m)=fg(pos1:m)-kb(1:2,3:4)*nos(no2,5:6)';
   end
end

if rest1==1
   kg(pos1,pos1)=kg(pos1,pos1)+kb(1,1);
   u2=nos(no1,6);
   fg(pos1)=fg(pos1)+fb(1)-kb(1,2)*u2;
   n=pos2+1;
   if rest2==0
      kg(pos1,pos2:n)=kg(pos1,pos2:n)+kb(1,3:4);
      kg(pos2:n,pos1)=kg(pos2:n,pos1)+kb(3:4,1);
   end
   if rest2==1
      kg(pos1,pos2)=kg(pos1,pos2)+kb(1,3);
      kg(pos2,pos1)=kg(pos2,pos1)+kb(3,1);
      u4=nos(no2,6);
      fg(pos1)=fg(pos1)-kb(1,4)*u4;
   end
   if rest2==2
      fg(pos1)=fg(pos1)-kb(1,3:4)*nos(no2,5:6)';
   end
end


if rest2==0
   n=pos2+1;
   kg(pos2:n,pos2:n)=kg(pos2:n,pos2:n)+kb(3:4,3:4);
```

```
      fg(pos2:n)=fg(pos2:n)+fb(3:4);
      if rest1==1
         u2=nos(no1,6);
         fg(pos2:n)=fg(pos2:n)-kb(3:4,2)*u2;
      end
      if rest1==2
         fg(pos2:n)=fg(pos2:n)-kb(3:4,1:2)*nos(no1,5:6)';
      end
   end

   if rest2==1
      kg(pos2,pos2)=kg(pos2,pos2)+kb(3,3);
      fg(pos2)=fg(pos2)+fb(3);
      u4=nos(no2,6);
      fg(pos2)=fg(pos2)-kb(3,4)*u4;
      if rest1==1
         u2=nos(no1,6);
         fg(pos2)=fg(pos2)-kb(3,2)*u2;
      end
      if rest1==2
         fg(pos2)=fg(pos2)-kb(3,1:2)*nos(no1,5:6)';
      end
   end
end

for i=1:ncn
   no=cn(i,1);
   soma=0;
   for j=1:no-1
      soma=soma+nos(j,3);
   end
   pos=(no-1)*ngln-soma+1;
   rest=nos(no,3);
   if rest==0
      m=pos+1;
      fg(pos:m)=fg(pos:m)+cn(i,2:3)';
   end
   if rest==1
      fg(pos)=fg(pos)+cn(i,2);
   end
end
fg
u=kg\fg

for s=1:nnos
   rest=nos(s,3);
   if rest ~= 0
```

```
reac=zeros(2,1);
for i=1:nel
   no1=el(i,1);
   no2=el(i,2);
   if no1==s | no2==s

      % inicio do bloco ##############################

      no1=el(i,1);
      no2=el(i,2);
      e=el(i,3);
      a=el(i,4);
      fb(1)=el(i,5);
      fb(2)=el(i,6);
      x1=nos(no1,1);
      y1=nos(no1,2);
      x2=nos(no2,1);
      y2=nos(no2,2);
      rest1=nos(no1,3);
      rest2=nos(no2,3);
      alf1=nos(no1,4);
      alf2=nos(no2,4);
      dx=x2-x1;
      dy=y2-y1;
      tet=calcang(dx,dy);
      gam1=tet-alf1;
      gam2=tet-alf2;
      beta=[cos(gam1) sin(gam1) 0 0; 0 0 cos(gam2)
sin(gam2)];
      lb=sqrt(dx*dx+dy*dy);
      ke=e*a*[1 -1; -1 1]/lb;
      kb=(beta')*ke*beta;
      fb(3:4)=fb(1:2);
      fb=fb*lb/2;
      r1=[cos(alf1) sin(alf1); -sin(alf1) cos(alf1)];
      r2=[cos(alf2) sin(alf2); -sin(alf2) cos(alf2)];
      fb(1:2)=r1*fb(1:2);
      fb(3:4)=r2*fb(3:4);

      soma=0;
      for j=1:no1-1
         soma=soma+nos(j,3);
      end
      pos1=(no1-1)*ngln-soma+1;
      soma=0;
      for j=1:no2-1
         soma=soma+nos(j,3);
```

```
            end
            pos2=(no2-1)*ngln-soma+1;

            % fim do bloco ####################################

            if rest1==0
               ub(1:2)=u(pos1:(pos1+1));
            end
            if rest1==1
               ub(1)=u(pos1);
               ub(2)=nos(no1,6);
            end
            if rest1==2

               ub(1:2)=nos(no1,5:6)';
            end
            if rest2==0
               ub(3:4)=u(pos2:(pos2+1));
            end
            if rest2==1
               ub(3)=u(pos2);
               ub(4)=nos(no2,6);
            end
            if rest2==2
               ub(3:4)=nos(no2,5:6)';
            end
            r=kb*ub-fb;
            if no1==s
               reac=reac+r(1:2);
            else
               reac=reac+r(3:4);
            end
         end
      end
      s
      reac
   end
end

lb1=1;
lb2=1;
lb3=sqrt(2);

ww1=el(1,5)*lb1;
ww2=el(2,5)*lb2;
ww3=el(3,5)*lb3;
```

```
p2=cn(1,2);
p3=cn(1,3);

alfa=nos(2,4);

rr3=(ww1+ww2+2*p2)/(2*cos(alfa))
rr2=-(p3+p2)+(ww1+ww2+2*ww3)/2
rr1=rr3*sin(alfa)-p2


% Fim do programa
```

PROGRAMA DE ENTRADA DE DADOS

```
Ngln = 2; % número de graus de liberdade nodais
Nnos = 3; % número total de nós
Nel = 3;  % número total de Elementos
NCn = 1;  % número de nós com condições de força concentradas
Ncm = 0;  % número de nós com molas concentradas

% Nós - xi - yi - rest - alfa - Ui - Vi
Nos=[0.0 0.0 2 0.0  0.0 0.0;
     1.0 0.0 1 pi/4 0.0 0.0;
     0.0 1.0 0 0.0  0.0 0.0];

% Elementos No1 No2 E A Fx Fy(forcas por unidade de comp.no
meio do elemento)
  El=[1 2 2.1e11 0.01 0 0.0 0.0;
      2 3 2.1e11 0.01 0 0.0 0.0;
      3 1 2.1e11 0.01 0 0.0 0.0];

% Condições de Newmman (Carregamento Concentrado)N0-P2 -p3
Cn=[3 1000 0.0];

% Condições mixtas (molas concentradas)
```

6. CRONOGRAMA

Atividades	Meses					
Estudo de noções de Elementos Finitos	x					
Estudo de noções do soft MATLAB		x				
Estudo de um algoritmo de programa de Elemento Finito Treliça			x	x		
Implantação com o MATLAB					x	
Redação					x	x
Apresentação do Trabalho de Estágio						x

7. CONCLUSÕES

Pode-se concluir que ao término deste estágio tive um ganho de conhecimentos, primeiro em relação ao método de elementos finitos, que apesar de superficial é um começo, quanto ao software MATLAB é uma ferramenta indispensável no dia a dia do engenheiro e também do pesquisador, neste consegui , poderia dizer, dominar mais ou menos o software - será sem dúvida uma ferramenta muito importante no meu dia a dia daqui para frente.

Observação – Este estágio teve início com as atividades do período 2003.1, tendo sido interrompido por ocasião da greve dos servidores federais.

8. REFERÊNCIAS BIBLIOGRÁFICAS

1. Fried, I. and Malkus, D. S., "Finite Element Mass Matrix Lumping by Numerical Integration With the Convergence Rate Loss", International Journal of Solids and Structures, Vol. 11, 1975, pp. 461-465.

2. Bang, H. and Kwon, Y. W., "Boundary Force Feedback for Flexible Structure Maneuver and Vibration Control", Proceedings for 1994 ASME Winter Meeting, Chicago, IL, November 6-11, pp. 59-70.

3. Site: www.dsc.ufcg.edu.br/~cnum

Susane Ribeiro
Aluna

Ítalo Ataíde Notaro
Orientador