

# Análise matricial de estruturas usando Matlab

## Carregando o MATLAB

O Matlab pode ser utilizado em qualquer maquina da Escola Politécnica, todos os computadores da pró-aluno o tem instalado.

No Gerenciador de Programas do Microsoft Windows deve-se abrir o grupo de programas do MATLAB for Windows, que contém o ícone do aplicativo MATLAB. Um duplo clique no ícone MATLAB carrega o aplicativo MATLAB.

## Ambiente de trabalho

Quando o MATLAB é carregado, dependendo da versão e das preferências do usuário, três janelas são exibidas: no caso apresentado na Figura 1 ,a Janela de Comando (Command Windows) e mais duas janelas auxiliares, por enquanto iremos precisar apenas da janela de comando.

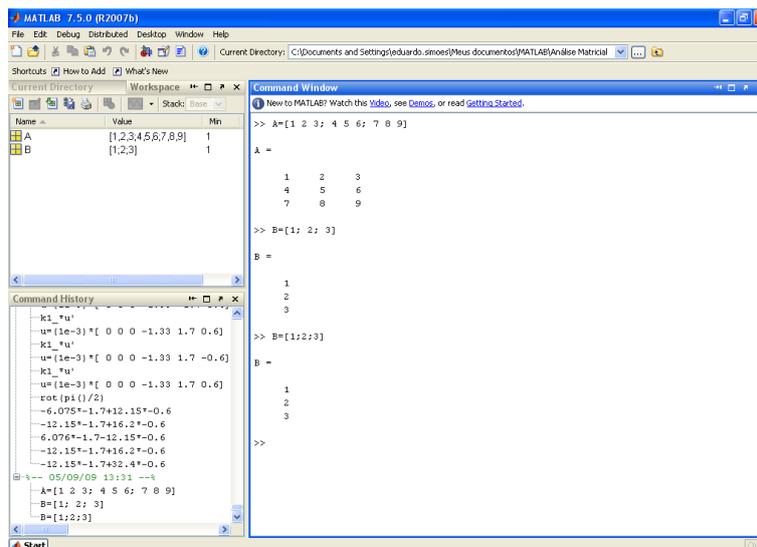


Figura 1

Quando o Matlab é aberto na janela de comando aparece o "prompt" padrão (>>) , é nele que iremos, inicialmente, fazer as operações numéricas.

## Matrizes no Matlab

### Entrando com matrizes

Uma das forças do Matlab é o número de operações disponíveis com matrizes, o que o torna bastante atrativo para o uso no ensino da análise matricial. Como exemplo inicial vamos definir uma matriz no matlab, na janela de comando digite no prompt:

```
>> A=[1 2 3; 4 5 6; 7 8 9]
```

Ao digitar esse comando aparecerá na janela de comando:

A =

1	2	3
4	5	6
7	8	9

Se olharmos na janela workspace (Figura 2) veremos que existe agora uma matriz A definida.

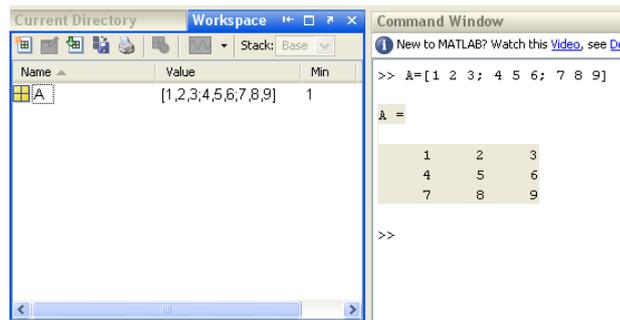


Figura 2

Para o Matlab não mostrar a matriz A, após digita-lá na tela de prompt, bastava adicionar “;” depois da matriz A.

```
>> A=[1 2 3; 4 5 6; 7 8 9];
```

O Matlab usa espaços para separar colunas e “;” para separar linhas. Outros exemplos:

Matriz linha

```
>> B=[1 2 3]
```

Matriz coluna

```
>> B=[1; 2; 3]
```

### ***Soma, transposição e multiplicação de matrizes***

Vamos começar definindo uma nova matriz C para somar com a matriz A anteriormente definida.

```
>>C=[22 22 33; 0 1 0; 1 20 0];
```

Para somar, basta fazer:

```
>>A+C
```

ans =

23	24	36
4	6	6
8	28	9

A matriz `ans` apresentada acima guarda a soma. Poderia ter se definido um matriz `D` que seria a soma das matrizes `A` e `C`.

```
>>D=A+C;
```

Para transpor uma matriz usa-se o comando `'` após a matriz, exemplo:

```
>> B=[1 2 3]
```

```
>> B =
```

```
1 2 3
```

```
>> B'
```

```
ans =
```

```
1
```

```
2
```

```
3
```

Para multiplicar `A` por  $B^T$  basta fazer:

```
>> A*B'
```

```
ans =
```

```
14
```

```
32
```

```
50
```

Também se pode definir matriz em função de variáveis previamente definidas, exemplo:

```
a =
```

```
50
```

```
>> b=30
```

```
b =
```

```
30
```

```
>> A=[ a b]
```

```
A =
```

```
50    30
```

### Índices de uma matriz

Uma outra vantagem do Matlab é eu poder facilmente acessar informações dos coeficientes de uma matriz usando os índices, exemplo:

```
>> B=[9 5 7]
```

```
>> B(1)
```

```
ans =
```

```
9
```

```
>>B(2)
```

```
ans =
```

```
5
```

Ou seja, eu digo a linha e a coluna na matriz e ele me diz qual é o valor naquele coeficiente, exemplo:

```
>> A=[1 5 7 ; 3 4 7; 9 16 53]
```

```
>> A =
```

```
1     5     7
```

```
3     4     7
```

```
9    16    53
```

```
>> A(1,3)
```

```
ans =
```

```
7
```

Ou a obtenção de submatrizes em bloco ou em qualquer ordem de linhas e colunas:

```
>> A([1 2],[1 2])
```

```
>> ans =
```

```
1     5
```

```
3     4
```

```
>> B=[11 12 13; 21 21 23; 31 32 33]
```

```
B =
```

```
11 12 13
21 21 23
31 32 33
```

```
>> B([1 3],[1 3])
```

```
ans =
```

```
11 13
31 33
```

No exemplo acima, foi montada uma matriz nova a partir de dois vetores que informam as colunas e as linhas de interesse. Essa função será muito usada daqui em diante. Os comandos apresentados até agora são os mínimos necessários para partirmos para um primeiro exemplo.

### Problema 1.

Dada um viga em balanço com  $EA = 3 \times 10^5 \text{ kN}$ ,  $EI = 3,24 \times 10^4 \text{ kNm}^2$  e  $l = 4\text{m}$ , como apresentada na Figura 3, e submetida aos esforços  $F = 10\text{kN}$  e  $M = 30\text{kNm}$ , pede-se para obter os deslocamentos da viga.

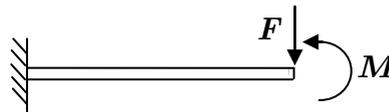


Figura 3

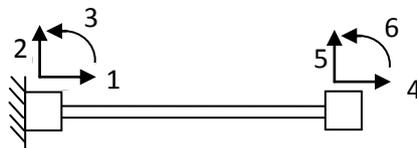


Figura 4

### Resolução

Precisamos antes de tudo apagar as variáveis usadas anteriormente e depois apagar a tela, para isso digite na tela de comando:

```
>> clear;
```

```
>>clc;
```

Após isso vamos digitar os parâmetros do problema.

```
>> EI=3.24e4;
>>EA=3e5;
>>l=4;
```

Tendo digitado os parâmetros podemos definir a matriz de rigidez do elemento biengastado usando a formula abaixo:

```
>>k_=[(EA)/l 0 0 -(EA)/l 0 0;...
      0 (12*EI)/(l^3) (6*EI)/(l^2) 0 -(12*EI)/(l^3) (6*EI)/(l^2);...
      0 (6*EI)/(l^2) (4*EI)/(l) 0 -(6*EI)/(l^2) (2*EI)/(l);...
      -(EA)/l 0 0 (EA)/l 0 0;...
      0 -(12*EI)/(l^3) -(6*EI)/(l^2) 0 (12*EI)/(l^3) -(6*EI)/(l^2);...
      0 (6*EI)/(l^2) (2*EI)/(l) 0 -(6*EI)/(l^2) (4*EI)/(l)];
```

Obs.: O “...” empregado no código acima serve para escrever comando extensos, permitindo otimizar o uso da tela e tornar mais visível os comandos.

Digitando na tela k\_, pode-se ver o resultado da matriz de rigidez local. Vale ressaltar que essa não é a maneira mais inteligente de trabalhar com matrizes de rigidez local do elemento, mas por enquanto será o método usado.

$$U_l = K_l^{-1}(R_l - R_{l0} - K_{lb}U_b) \quad (1.1)$$

$$U_b = 0$$

$$R_{l0} = 0$$

$$R = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -F \\ M \end{bmatrix}$$

Para acharmos os deslocamentos livres precisamos definir **R**.

```
>>R=[0 0 0 0 -10 30]';
>>R =
      0
      0
      0
      0
     -10
      30
```

Agora precisamos definir os graus de liberdade da estrutura, que no caso são três:

$$GLL = [4 \ 5 \ 6]^T$$

No Matlab fica:

```
>>GLL=[4 5 6]';
```

Finalmente resolvendo a equação (1.1), obtemos:

```
>> inv(k_(GLL,GLL))*R(GLL)
```

```
ans =  
  
0  
0.0008  
0.0012
```

### Funções no Matlab

Duas características básicas do Matlab são ser um ambiente interativo para cálculos computacionais e ao mesmo tempo ser uma linguagem de programação interpretada. A programação no Matlab é feita usando o arquivo de extensão .m (M-files). Um M-file pode ser uma função ou um script. Nessa seção vamos apenas explorar as funções, posteriormente discutindo os scripts.

Para criar um M-file clique em File>New>M-file (figura 5).

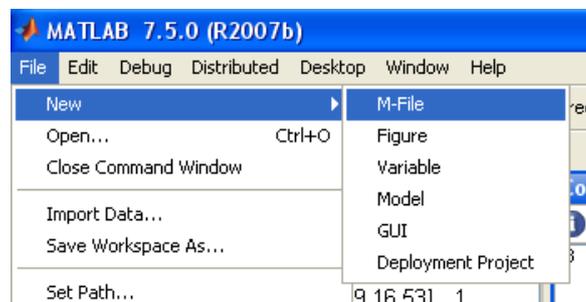


Figura 5

Ao criar um M-File aparecerá uma nova janela, é nesse ambiente que se programa no Matlab. Uma função no Matlab é definida digitando no início do M-file um comando do tipo:

```
function c = soma(a,b)
```

Onde `function` é como se define no M-file uma função, `c` é a variável de saída da função, “soma” é o nome da função, como ela é chamada pela janela de comando e por outros M-files, e “a” e “b” são as variáveis de entrada, siga o exemplo:

- Crie um novo M-file
- Digite nele:

```
function c=soma(a,b)
```

c=a+b;

- Salve com o nome soma
- Na tela de comando digite:

soma (2, 3)

- resultado:

ans=5

Não é a maneira mais inteligente de somar no Matlab, mas serve de exemplo simples. Um exemplo mais útil é definir uma função que dando as variáveis  $EA$ ,  $EI$  e  $l$ , me dê a matriz de rigidez local de uma viga biengastada. Exemplo:

```
function k=biengastada(EI,EA,l)
```

```
k=[(EA)/l 0 0 -(EA)/l 0 0;...  
0 (12*EI)/(l^3) (6*EI)/(l^2) 0 -(12*EI)/(l^3) (6*EI)/(l^2);...  
0 (6*EI)/(l^2) (4*EI)/l 0 -(6*EI)/(l^2) (2*EI)/l;...  
-(EA)/l 0 0 (EA)/l 0 0;...  
0 -(12*EI)/(l^3) -(6*EI)/(l^2) 0 (12*EI)/(l^3) -(6*EI)/(l^2);...  
0 (6*EI)/(l^2) (2*EI)/l 0 -(6*EI)/(l^2) (4*EI)/l];
```

Essa função será usada nos próximos exemplos muitas vezes e aconselha-se criar funções desse tipo para guardar as matrizes de rigidez locais. Outra função importante é a de rotação do sistema de eixos. Exemplo:

```
function T=rot(alpha)
```

```
T=[cos(alpha) sin(alpha) 0 0 0 0;...  
-sin(alpha) cos(alpha) 0 0 0 0;...  
0 0 1 0 0 0;...  
0 0 0 cos(alpha) sin(alpha) 0;...  
0 0 0 -sin(alpha) cos(alpha) 0;...  
0 0 0 0 0 1];
```

Onde alpha é o ângulo de rotação seguindo a convenção abaixo.

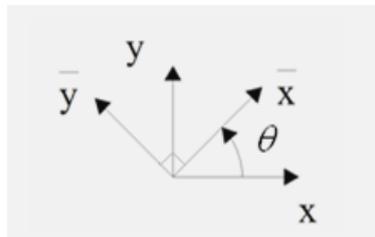


Figura 6