

Não entregar.

Iniciar o Matlab e entrar os comandos conforme apresentados e descobrir o que acontece em cada caso.

1. Matlab como calculadora

```
2+2;
2+2
ans
disp(5)
2*ans
x=2
exp(x)
x=x+1
sin(pi/2)
y=[1 2 3; 4 5 6; 7 8 9]
who
whos
why
```

2. Comandos de área de trabalho

```
help exp
lookfor float
help diary
pwd
mkdir cci22
cd cci22
diary diario.txt
diary on
help pinv
type pinv
disp('Hello world')
% this is a comment
diary off
edit diario.txt
save dados
clear
whos
load dados
whos
pause
pause(3)
quit
```

3. Ponto flutuante

```
help format
format long e
0.12-0.1
eps
realmax
realmin
format rat
pi
1:30
format compact
1:30
```

4. Manipulação de matrizes

```
vetor=[1 2 -2 4 1]
vetcoluna=[1; 2; -2; 4; 1]
hcat=[vetcoluna, vetcoluna]
vcat=[vetcoluna; vetcoluna]
vetor'
vetor*vetcoluna
vetcoluna*vetor
2*vetor
norm(vetor)
vetor.*vetor
matriz=[1 1 3; 3 4.0 2; 1 5 1]
matriz'
matriz*matriz
matriz^2
matriz.*matriz
matriz.^2
inv(matriz)
det(matriz)
trace(matriz)
eig(matriz)
rank(matriz)
size(matriz)
eye(3)
zeros(3)
ones(3,4)
```

5. Intervalos e índices

```
matriz(2,3)
matriz(2,3)=7
matriz(1,:)
matriz(1,:)=[3 4 5]
matriz(1:2, 2:3)
matriz(:,2)
matriz(2:end, 1:end-1)
matriz(:)
1:10
1:0.5:10
10:-0.5:1
linspace(1, 10, 13)
logspace(1, 2, 10)
max(1:10)
prod(1:10)
sum(1:10)
mean(1:10)
max(matriz)
max(max(matriz))
sum(sum(matriz))
sin(0:pi/16:pi)
```

6. Gráficos

```
t=0:pi/16:2*pi
plot(cos(t),sin(t))
axis square
grid
title('grafico da circunferencia')
whitebg('b')
xlabel('eixo x')
ylabel('eixo y')
hold on
plot(cos(3*t),sin(t),'m:')
plot(cos(t),sin(5*t),'g--')
legend('circ','curva 3','curva 5')
figure
subplot(2,1,1);
plot(t,cos(t),'ro');
title('cosseno');
subplot(2,1,2);
plot(t,sin(t),'ro');
title('seno');
[x,y]=ginput(1)
```

7. Gráficos em 3 dimensões

```
figure;
[x,y]=meshgrid(-1:0.4:2, 0:0.4:4)
d= sqrt(x.^2+y.^2);
z=sin(d)./d;
surf(x,y,z);
figure;
[x,y]=meshgrid(-10:0.4:20, 0:0.4:40);
d= sqrt(x.^2+y.^2);
z=sin(d)./d;
surf(x,y,z);
pcolor(x,y,z);
mesh(x,y,z);
surfc(x,y,z,abs(cos(d)));
colormap(bone);
plot3(x,y,z)
contour3(x,y,z)
```

Rotacione os gráficos utilizando os controles da janela gráfica. Exporte os gráficos como imagens BMP, por exemplo.

8. Scripts

```
edit grafico.m
```

Copie uma seqüência de comandos dos itens 6 ou 7 para o editor do arquivo grafico.m e salve. Agora, basta chamar:

```
grafico
```

Observe o manual para os comandos estruturados:

```
help if
help while
help relop
help for
```

Observe que, no editor do Matlab, o script pode ser executado passo a passo e as variáveis podem ser consultadas.

9. Funções

```
type factorial
```

Veja como é definida a função fatorial. O valor de retorno é atribuído a uma variável de retorno definida na primeira linha do arquivo. Mais de um valor ou objeto pode ser retornado pela função:

```
help function
```

Não é feito controle dos parâmetros passados, que devem ser testados pelo próprio programa (ver nargin e narginout). Tente deixar a função mais geral, podendo aceitar vetores e matrizes como parâmetros e não apenas valores.

Tente reescrever a função fatorial para que possa receber matrizes como parâmetros.

```
help global
```

As variáveis definidas dentro de funções têm escopo definido dentro da função. Variáveis globais devem ser declaradas como global dentro e fora da função.

Veja agora uma função mais complexa que realiza a eliminação gaussiana com pivoteamento parcial de uma matriz.

```
type rrefmovie
```

10. Observações sobre o Matlab

Experimente criar um script com a seguinte seqüência de comandos:

```
clear;
tic;
for i=1:1000
    for j=1:1000
        f(i,j)=5;
    end;
end;
toc
```

Veja o tempo que levou para construir uma matriz. Agora teste novamente com

```
clear;
tic;
f=zeros(1000,1000);
for i=1:1000
    for j=1:1000
        f(i,j)=5;
    end;
end;
toc
```

E novamente com

```
clear;
tic;
f=5*ones(1000,1000);
toc
```

Assim, devem ser evitadas operações de manipulação da memória, como a extensão do tamanho de vetores, e malhas de repetição muito internas como o segundo for do primeiro script acima. Devem ser utilizadas preferencialmente operações de matrizes, buscar com “find”, somar com “sum”, comparar com “any” e “all”, entre as outras ao invés de construir malhas de repetição.