

Funções

Tópico

- Funções (Sub-rotinas)

FUNÇÕES

Uma função é um arquivo que pode ser usado de dois modos:

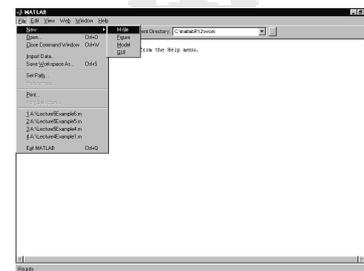
- ❖ Para realizar uma tarefa frequente. Exemplos: calcular o valor de uma função matemática para diferentes valores da variável independente; realizar uma série de comandos com diferentes valores de variáveis.
- ❖ Como uma sub-rotina em uma rotina principal. Neste modo, uma rotina grande pode ser dividida em pequenos blocos que podem ser testados individualmente ou reutilizados em outras rotinas (redução de código).

FUNÇÕES

- As funções são usadas do mesmo modo que as funções nativas. i.e. uma vez criadas, as funções podem ser usadas na command window, em rotinas ou dentro de outras funções.
- Geralmente, os dados são transferidos para a função pelas variáveis de entrada e os resultados são devolvidos pelas variáveis de saída.
- Todos os cálculos e variáveis usadas dentro da função são locais, i.e. eles não são reconhecidos, transferidos ou disponibilizados para outras partes do MATLAB.
- Funções são como as sub-rotinas em FORTRAN e BASIC, procedimentos em PASCAL e funções em C.

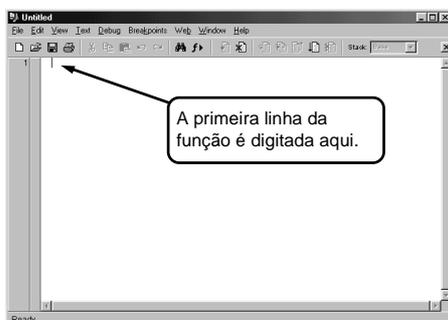
CRIAÇÃO DE UMA FUNÇÃO

Uma função é criada no editor/depurador de M-file (a mesma janela usada para criar rotina). Na command window clique no menu **File**, selecione **New** e, em **M-file**.



Após **M-file** ser selecionado, é aberta a janela do M-file Editor/Debugger.

A janela do M-file Editor/Debugger



LINHA DE DEFINIÇÃO DA FUNÇÃO

A primeira linha de uma função **PRECISA** ser a **linha de definição da função** (sem esta linha o arquivo será apenas uma rotina).

A linha de definição da função:

1. Define que o M-file é uma função.
2. Define o nome da função.
3. Tem a lista das variáveis de entrada e saída.

```
function [variáveis de saída] = nome_da_funcao(variáveis de entrada)
```

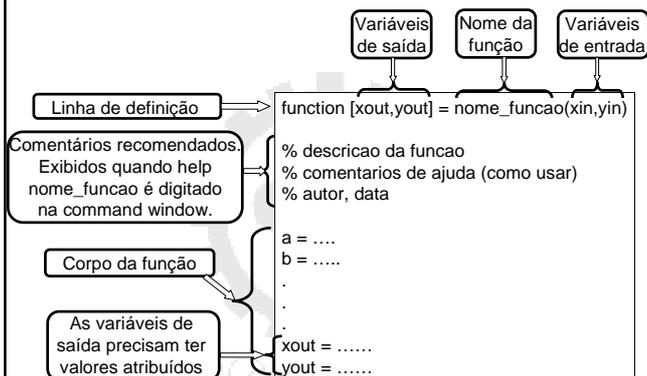
Quando a função for salva, o nome do arquivo **PRECISA** ser idêntico ao nome_da_funcao (com a extensão .m).

EXEMPLOS DE LINHAS DE DEFINIÇÃO DE FUNÇÃO E NOMES DE ARQUIVOS

<u>Linha de definição da função</u>	<u>Nome do arquivo</u>
function [A] = AreaRet(a,b)	AreaRet.m
function[V, S] = VolAreaEsfera(r)	VolAreaEsfera.m
function[d,h] = projetil(v,teta)	projetil.m
function = PlotaCirculo(r)	PlotaCirculo.m

7

FORMATO DE UMA FUNÇÃO



8

COMENTÁRIOS À RESPEITO DE FUNÇÕES

- A palavra function é a primeira palavra da linha de definição da função e deve ser digitada com letras minúsculas.
- Os colchetes não são necessários na linha de definição caso a função tenha somente uma variável de saída.

```
function [A] = AreaRet(a,b)
function A = AreaRet(a,b)
```

Ambas as formas estão ok.

- Se não existirem variáveis de saída, os colchetes e o sinal de igual podem ser omitidos.

```
function = PlotaCirculo(r)
function PlotaCirculo(r)
```

Ambas as formas estão ok.

9

COMENTÁRIOS À RESPEITO DE FUNÇÕES

- As variáveis de entrada e saída, fornecidas na linha de definição e no corpo da função, são locais. Isto significa que outros nomes de variáveis podem ser usados na chamada da função. Números ou expressões matemáticas também podem ser usados como variáveis de entrada. (Exemplo no próximo slide)
- As variáveis são atribuídas de acordo com sua posição na lista de variáveis de saída e de entrada da linha de definição da função.
- Em geral, não se usa os comandos input e disp numa função, pois a entrada de dados é feita através do argumento da função (entre parenteses) e a saída é obtida através das variáveis de saída.

10

Linha de definição da função

Exemplo de variáveis quando a função é usada

```
function [A] = AreaRet(a,b)
S=AreaRet(g,r)
T=AreaRet(8,25)
```

No primeiro exemplo, *g* e *r* precisam conter valores antes delas serem usadas como variáveis de entrada na função. Quando a função é executada, *a* terá o valor de *g*, *b* terá o valor de *r*. O valor de *A* será atribuído a *S*.

No segundo exemplo, a variável *a* é atribuído o valor 8 e a *b* o valor 25.

11

COMENTÁRIOS À RESPEITO DE FUNÇÕES

- Tal como acontece na command window e nas rotinas, nas funções o ponto-e-vírgula também suprime a saída de um comando.
- Se o ponto-e-vírgula não for digitado, a saída é exibida na command window. Isto pode ser útil para a depuração da função.

12

SALVANDO UM ARQUIVO DE FUNÇÃO

- ❖ Após a função ser concluída, ela precisa ser salva. Para isso, use **Save As...**
- ❖ Não nomeie a função com nome já usado pelo MATLAB para uma função nativa. Para verificar se o nome já é usado, digite "help nome_escolhido" na command window.

FEL-USP

13

EXEMPLO DE FUNÇÃO

```
function valor = valorpoupanca(depo,t,taxa) % Linha de definição
% A funcao calcula o valor de uma poupanca
% em que o rendimento aumenta anualmente.
% A saida da funcao eh o valor da conta.
% A entrada da funcao eh:
% depo: o deposito inicial.
% t: numero de anos.
% taxa: a taxa de rendimento em percentual.
format bank
valor=depo*(1 + taxa/100)^t; % Valor atribuído à variável de saída
```

Variável de saída: valor
Nome da função: valorpoupanca
Variáveis de entrada: depo, t, taxa
Comentários: % A funcao calcula o valor de uma poupanca...
Valor atribuído à variável de saída: valor=depo*(1 + taxa/100)^t;

14

EXECUÇÃO DA FUNÇÃO valorpoupanca

Abaixo são mostrados três exemplos de execução da função valorpoupanca na command window:

```
>> x = valorpoupanca(20000,15,6.5)
x =
    51436.82
```

```
>> quantia = 20000;
>> anos = 15;
>> initx = 6.5;
>> dinheiro = valorpoupanca(quantia,anos,initx)
dinheiro =
    51436.82
```

```
>> quantia = 20000;
>> valorpoupanca(quantia,15,6.5)
ans =
    51436.82
```

15

EXEMPLO DE FUNÇÃO

```
function [dmax,hmax] = trajetoria(v0,teta)
% A funcao calcula a trajetoria de um projetil.
% A entrada da funcao eh:
% v0: a velocidade inicial (unidade: m/s) do projetil.
% teta: o angulo (unidade: grau) em que o projetil eh disparado.
% A saida da funcao eh:
% dmax: a distancia maxima (unidade: m) que o projetil percorre.
% hmax: a altura maxima (unidade: m) que o projetil alcanca.
% alem disso, a funcao cria o grafico da trajetoria.

v0x = v0*cos(teta*pi/180); % Componente horizontal da velocidade inicial.
v0y = v0*sin(teta*pi/180); % Componente vertical da velocidade inicial.
(Continua no próximo slide)
```

16

EXEMPLO DE FUNÇÃO

```
hmax = v0y^2/(2*9.81); % A altura maxima.
t = v0*sin(teta*pi/180)/9.81; % O tempo ate atingir o ponto mais alto.
ttotal = 2*t; % O tempo total de voo do projetil.
dmax = v0x*ttotal; % A distancia maxima percorrida.
tplot = linspace(0,ttotal,200); % Criacao de um vetor de tempo.
x = v0x*tplot; % Coordenada x em funcao do tempo.
y = v0y*tplot+0.5*(-9.81)*tplot.^2; % Coordenada y em funcao do tempo.
plot(x,y)
xlabel('DISTANCIA')
ylabel('ALTURA')
title('TRAJETORIA DO PROJETIL')
```

17

EXECUÇÃO DA FUNÇÃO trajetoria

Execução da função na command window para:

$V_0 = 250$ m/s e $teta = 32$ graus.

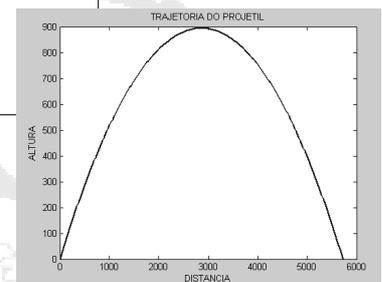
```
>> [dist,alt]=trajetoria(250,32)
```

```
dist =
```

```
    5.7263e+003
```

```
alt =
```

```
    894.5414
```



18