

Métodos Numéricos em Equações Diferenciais

Prática 1 - Introdução ao Matlab

Profa. Vanessa Rolnik

curso: Matemática Aplicada a Negócios

MATLAB

- ▶ **MA**trix **LAB**oratory - matrizes ou arranjos são os seus elementos de dados básico
- ▶ é uma poderosa linguagem usada na computação técnica
- ▶ específico para cálculos matemáticos, modelagem e simulações, análise de dados e processamento, visualização e traçado de gráficos; no desenvolvimento de algoritmos
- ▶ usado amplamente em Universidades e indústria, para pesquisa e desenvolvimento de projetos
- ▶ programa padrão com funções + *toolboxes* (ferramentas opcionais) formadas por conjuntos de programas especializados para resolver problemas específicos. Ex. EDP, aquisição de imagens, processamento de imagens

Toolboxes

- ▶ Parallel Computing Toolbox
- ▶ Partial Differential Equation Toolbox
- ▶ Statistics and Machine Learning Toolbox
- ▶ Curve Fitting Toolbox
- ▶ Optimization Toolbox
- ▶ Global Optimization Toolbox
- ▶ Neural Network Toolbox
- ▶ Fuzzy Logic Toolbox
- ▶ Signal Processing Toolbox
- ▶ Image Processing Toolbox
- ▶ Image Acquisition Toolbox
- ▶ Data Acquisition Toolbox
- ▶ Bioinformatics Toolbox
- ▶ etc

Limitações

- ▶ execução de algoritmos mais lenta que em linguagens de programação (C, Fortran, etc)
- ▶ alguns procedimentos gráficos e de interação com o usuário são restritos aos comandos do MATLAB
- ▶ não é possível gerar arquivos executáveis com o MATLAB, ou seja, um arquivo MATLAB só poderá ser executado no ambiente MATLAB
- ▶ é limitado para processamento de expressões matemáticas analíticas

Janelas

- ▶ Janela de Comandos (Command Window): janela principal, permite a entrada de variáveis e roda programas
- ▶ Janela de Figuras (Figure Window): contém a saída de comandos gráficos
- ▶ Janela de Edição (Editor Window): usada para escrever e editar programas
- ▶ Janela de Ajuda (Help Window): fornece informação de ajuda
- ▶ Janela de Lançamento (Launch Pad Window): permite o acesso a ferramentas, demonstrações e documentação
- ▶ Janela Histórico de Comandos (Command History Window): mostra os comandos utilizados na janela de comandos
- ▶ janela Área de Trabalho (Workspace Window): fornece informações sobre as variáveis utilizadas
- ▶ janela de Diretório Corrente (Current Directory Window): mostra os arquivos no diretório corrente

MATLAB - Floating-Point Numbers

“MATLAB® represents floating-point numbers in either double-precision or single-precision format. The default is double precision...”

```
» x = 25.783;  
  
» whos x  
  
» isfloat(x)  
  
» y = single(x);  
  
» quemx = whos('x');  
  
» quemx.bytes
```

MATLAB - Floating-Point Numbers

```
» str = 'The range for double is: sprintf(str, -realmax,  
-realmin, realmin, realmax)
```

```
» realmax + .0001e+308
```

```
» -realmax - .0001e+308
```

```
» str = 'The range for single is: sprintf(str,  
-realmax('single'), -realmin('single'), ...  
realmin('single'), realmax('single'))
```

```
» realmax('single') + .0001e+038
```

```
» -realmax('single') - .0001e+038
```

MATLAB - Precisão

```
» format long  
» eps(5)  
» eps(50)  
» x = single(5); eps(x)  
» double(single(3.14) - 3.14)
```

What You Get Is Not What You Expect

```
» e = 1 - 3* (4/3 - 1)

» a = 0.0;
» for i = 1:10
    a = a + 0.1;
end
» a == 1

» b = 1e-16 + 1 - 1e-16;
» c = 1e-16 - 1e-16 + 1;
» b == c

» (2^53 + 1) - 2^53

» sin(pi)
```

What You Get Is Not What You Expect

```
» sqrt(1e-16 + 1) - 1
```

```
» A = diag([2 eps]);
```

```
» b = [2; eps];
```

```
» y = A\b;
```

Exercício

Procurar no HELP: Floating-Point Functions

- | | |
|--------------------|---------------------|
| 1) ceil | 2) class |
| 3) fix | 4) floor |
| 5) round | 6) nan |
| 7) format long e | 8) bitmin/bitmax |
| 9) int8 | 10) isnumeric |
| 11) i e j | 12) format shortEng |
| 13) format compact | 14) format hex |
| 15) format shortG | |

MATLAB - Matrices

```
» A=[ 5 35 43; 4 76 81; 21 32 40]
```

```
» cd = 6 ; e = 3 ; h = 4;
```

```
» Mat=[ e , cd*h , cos(pi/3) ; h^2, sqrt(h* h/cd) , 14 ]
```

```
» x1=[1:2:13]
```

```
» x2=1:2:13
```

```
» S='Exemplo de string:%123'
```

```
» x3=linspace(1,13,7)
```

```
» length(A)
```

```
» x1(1)
```

```
» size(A)
```

```
» A(2,3)
```

```
» zeros(m,n)
```

```
» u=x1(3:6)
```

```
» ones(m,n)
```

```
» mat(2:3;2:3)
```

```
» eye(n)
```

MATLAB - Operações com matrizes

- » $A = \begin{bmatrix} 5 & 35 & 43 \\ 4 & 76 & 81 \\ 21 & 32 & 40 \end{bmatrix};$
- » $A_soma = A + eye(3)$
- » $A_mult = A * A_soma$
- » $AV = \begin{bmatrix} 2 & 5 & 1 \end{bmatrix}; BV = \begin{bmatrix} 3 \\ 1 \\ 4 \end{bmatrix};$
- » $AV * BV$
- » $BV * AV$
- » $A_transp = A'$ (ou $transpose(A)$)
- » $A_det = det(A)$
- » $A_inv = inv(A)$ (ou A^{-1})

MATLAB - Solução de sistemas lineares

» $A = [4 \ -2 \ -3 \ 6; \ -6 \ 7 \ 6.5 \ -6; \ 1 \ 7.5 \ 6.25 \ 5.5; \ -12 \ 22 \ 15.5 \ -1];$

» $b = [12; \ -6.5; \ 16; \ 17];$

» $x = A \backslash b$ (divisão à esquerda)

» $x = b/A'$ (divisão à esquerda)

» $x = \text{inv}(A)*b$ (ou » $x = A^{-1}*b$) (inversão da matriz A)

» $x = \text{linsolve}(A,b)$

MATLAB - Operações elemento a elemento

- » $A=[2 \ 6 \ 3; 5 \ 8 \ 4]; B=[1 \ 4 \ 10; 3 \ 2 \ 7];$
- » $A-2$ (subtrai 2 de cada elemento da matriz A)
- » $A.*B$ (multiplica os elementos que ocupam a mesma posição)
- » $A./B$ (divide cada elemento de A pelo correspondente de B)
- » $A.\backslash B$ (divide cada elemento de B pelo correspondente de A)
- » $A.^B$ (cada elemento de A é elevado ao elemento correspondente de B)

MATLAB - Arquivos de programa

- ▶ extensão .m
- ▶ abrir Editor Window (criado no ícone New Script)
- ▶ digitar código
- ▶ salvar no diretório corrente (se necessário, alterar o diretório)
- ▶ executar digitando seu nome na janela de comandos (ou na janela de edição clicando no ícone Run)
- ▶ tanto as variáveis declaradas na janela de comandos quanto as declaradas no arquivo são **globais**
- ▶ os dados são exibidos digitando a variável + enter
ou `disp(nome da variável)`
ou `disp('texto')`
ou `fprintf('texto %5.2f texto', nome da variável)`

MATLAB - Arquivos de funções

- ▶ 1a. linha:

```
function[arg_saída]=nomefunção(arg_entrada)
```

- ▶ pode conter: comentários, código, atribuição de valores, outras funções criadas pelo usuário, funções residentes, laços de repetições, testes condicionais, etc
- ▶ todas as variáveis presentes no arquivo de função são **locais**

MATLAB - Programação

► Operadores lógicos e relacionais

$x < y$ $x > y$ $x \leq y$ $x \geq y$ $x == y$ $x \sim = y$

$x \& y$ AND $x | y$ OR $\sim x$ NOT

sentença verdadeira recebe valor 1 e falsa recebe valor 0

► Estrutura condicional

if-end if-else-end if-elseif-else-end

► Laços de repetição

for k=i:p:f (ou k=i:f com p=1 default) - end

while-end

Exemplo programação

```
V=[5, 17, -3, 8, 0, -7, 12, 15, 20, -6, 6, 4, -2, 16];  
n=length(V);  
for k=1:n  
    if V(k)>0 && (rem(V(k),3)==0 || rem(V(k),5)==0)  
        V(k)=2*V(k);  
    elseif V(k)<0 && V(k)>-5  
        V(k)=V(k)^3;  
    end  
end  
V
```

Exemplo programação

» V=[5, 17, -3, 8, 0, -7, 12, 15, 20, -6, 6, 4, -2, 16];

No script:

```
function [Z]=teste(V);
```

```
n=length(V);
```

```
for k=1:n
```

```
    if V(k)>0 && (rem(V(k),3)==0 || rem(V(k),5)==0)
```

```
        V(k)=2*V(k);
```

```
    elseif V(k)<0 && V(k)>-5
```

```
        V(k)=V(k)^3;
```

```
    end
```

```
end
```

```
Z=V;
```

MATLAB - gráficos

- ▶ gráficos de funções `fplot('função',[xi,xf])`
- ▶ gráfico de vetores `plot(x,y)` gera a curva com os valores de x na abscissa e y na ordenada
- ▶ obs: x e y de mesma dimensão!!!!
- ▶ escala linear, intervalo entre os dados padrão, dados conectados por segmentos de retas
- ▶ Ex.
 - » `x=linspace(-5,5,11)`
 - » `y=x.*2`
 - » `plot(x,y)`

MATLAB - gráficos

- ▶ `plot(x,y,'especificadores de linha')`
- ▶ Ex.
 - » `plot(x,y,'-r','linewidth',2,'markersize',12)`
 - » `xlabel('Temperatura (C)');`
 - » `ylabel('Pressão (atm)');`
 - » `title('Dados experimento para determinar zero absoluto');`
- ▶ Obs: A formatação de gráficos também pode ser feita no Editor de Gráficos

Ex 1 - determinação da temperatura de zero absoluto

$T(^{\circ}\text{C})$	0	10	20	30	40	50	60	70	80	90	100
$p(\text{atm.})$	0,94	0,96	1,0	1,05	1,07	1,09	1,14	1,17	1,21	1,24	1,28

- 1) trace um gráfico (diagrama de dispersão) dos dados;
- 2) faça um ajuste linear;
- 3) trace o gráfico da função de ajuste e estenda (extrapole) até que cruze o eixo T horizontal. Esse ponto é uma estimativa da temperatura do zero absoluto.

Funções residentes do MATLAB

`p=polyfit(x,y,n)` ou `[p,S,mu]=polyfit(x,y,n)`

p: vetor contendo os coeficientes do polinômio que melhor se ajusta aos dados, (x,y): pontos, n: grau do polinômio de ajuste, S: erro, mu: vetor com média e desvio-padrão

Ex.

» `x=0:10;`

» `y=[ 0, 0.7, 2.4, 3.1, 4.2, 4.8, 5.7, 5.9, 6.2, 6.4, 6.4];`

» `coef=poyfit(x,y,1)`

» `reta=coef(1).*x+coef(2);`

» `plot(x,reta,x,y,'o')`

repetir para `n=2`, `n=3`

Referências bibliográficas

- [1] A. Gilat; V. Subramaniam, Métodos Numéricos para Engenheiros e Cientistas: uma introdução com aplicações usando o MATLAB, Bookman, 2008.
- [2] R. L. Burden; J. D. Faires, Análise Numérica, Cengage Learning, 2008.