INTRODUÇÃO AO MATLAB

Aurélio Lima Araújo
E-mail:aaraujo@ipb.pt

PROGRAMA

- Introdução.
- Vectores e matrizes.
- Polinómios.
- Gráficos 2D e 3D.
- Programação.
- Análise numérica.
- Ajuda.

INTRODUÇÃO

- O Que é o Matlab?, MATrix LABoratory

O MATLAB é um programa para realizar cálculos numéricos com ***vectores e matrize***s. Como caso particular pode também trabalhar com números escalares, tanto reais como complexos. Uma das capacidades mais atractivas é a de realizar uma ampla variedade de ***gráficos*** em duas e três dimensões.

Introdução

Os elementos básicos do Matlab, como qualquer outra linguagem de programação, são: constantes, variáveis, operações, expressões e funções.

Constantes numéricas:

- Números inteiros: 2 35 -48
- Números reais: 2. -35.2 48.45
    - Máximo de 16 dígitos significativos
    - Utilizando a letra E na continuação de um nº com ponto decimal [2.2250e-308 1.7e+308].
- Números complexos: 2+3i 4*j   i,j=$(-1)^{1/2}$

Operações aritméticas elementares:

Soma: +   Multiplicação: *   Exponenciação: ^
Subtracção: -   Divisão: /

Primeiro exponenciações, depois divisões e multiplicações e por último somas e subtracções.

Variáveis: é a etiqueta que identifica uma porção de memória;

O Matlab diferencia entre maiúsculas e minúsculas

Para ver as variáveis definidas num determinado instante escreve-se:

>> who

ou

>> whos

Para eliminar alguma variável executa-se:

>> clear variable1 variable2

Expressões numéricas: são um conjunto de números, funções e variáveis previamente definidas, relacionados todos eles por operadores aritméticos. Se uma expressão é demasiado grande indica-se a sua continuação mediante três pontos *(...)*.

Introdução

Formatos: por defeito o Matlab tem formato *short* mas pode-se escolher entre os seguintes formatos:

- >> format long (14 dígitos significativos)
- >> format short (5 dígitos significativos)
- >> format short e (notação exponencial)
- >> format long e (notação exponencial)
- >> format rat (aproximação racional)

Variáveis predefinidas em Matlab:

$i = (-1)^{½}$ pi = π Inf= ∞ NaN= cálculos indefinidos

eps = < nº que + outro nº=nº vírgula flutuante 2.22e-16

date = valor da data actual

rand = gera números aleatórios [0 1]

realmin = <nº+   realmax= >nº+

Funções do Matlab:

nome(argumento)

- sqrt (x) raiz quadrada
- abs(x) módulo de x
- conj(z) conjugado de um complexo
- real(z), imag(z) parte real e imaginária de z, respectivamente
- exp(x) calcula e^x, sendo x real ou complexo
- sin(x) asin(x) [$-\pi/2$ $\pi/2$] cos(x) acos(x) [0 π] tan(x)
- atan(x) [$-\pi/2$ $\pi/2$] angle(z) log(x) (em base e) log10(x)
- rats(x) rem(x,y) resto de x/y round(x) sign(x)

Comandos de ajuda:

- help
- lookfor
- what ficheiros .m e .mat do directório actual
- dir ficheiros do directório actual

Introdução

- type   nome_ficheiro Mostra o conteúdo do ficheiro
- delete nome_ficheiro Apaga o ficheiro
- cd muda de directoria
- pwd indica a directoria actual
- which nome_ficheiro indica a directoria onde está
- ! Abre uma janela de MSDOS que se fecha quando regressamos ao Matlab

startup.m ficheiro de arranque ao executar o matlab.
Para guardar num ficheiro os comandos que se executam numa sessão:

```
>> diary nome_ficheiro
...
>> diary off
```

Introdução

diary tema1.dia
clear

Exercício 1.1 Calcular o valor da expressão:

$$J=\frac{42.1768+234}{2^{10}-10247}$$

Exercício 1.2 Calcular o valor da expressão:

$$H=\frac{9.8*10^{14}+5.876*10^{-5}}{9.987*10^{5}-10^{6}}$$

E escrever o resultado em pelo menos 2 formatos

Exercício 1.3 Calcular $I=\sqrt[7]{\frac{3\,sen(32^{o}\,15')}{42.1^{3}}}$

Exercício 1.4 Segundo Hill e Lounasmaa, a equação da curva de inversão do hélio é $P=-21.0+5.44T-0.132T^{2}$
Onde a pressão P vem dada em atmosferas e a temperatura T em Kelvin. Calcular o valor da pressão a uma temperatura de 293 K. Calcular o valor da temperatura para uma pressão de 1N/m^2.
Nota: 1N/m^2=9.265*10^{-6}atm

Introdução

```
diary off
dir
type tema1.dia
```

VECTORES E MATRIZES

- As matrizes são o tipo fundamental de estrutura em Matlab.

```
» A=[1 3 5; 6 9 2; 4 8 7]

A =

     1     3     5
     6     9     2
     4     8     7

» det(A)

ans =

     5
```

```
» A^2+3*A

ans =

    42    79    61
    86   142    68
    92   164   106
```

<u>MAT</u>rix <u>LAB</u>oratory
-- dados são matrizes
-- regras da álgebra linear

Vectores e matrizes

Os vectores podem ser vectores linha ou vectores coluna

Vectores linha: os elementos de uma mesma linha estão separados por ***espaços*** ou ***vírgulas***,

>> v =[2 3 4]

Vectores coluna: os elementos de uma mesma coluna estão separados por ***ponto e vírgula*** (;).

>> w =[2;3;4;7;9;8]

A dimensão de um vector obtém-se pelo comando length (nome do vector) sub_w=w(i:k:j)

Criação de vectores:

• Especificando o incremento das suas componentes v=a:h:b;

• Especificando a sua dimensão linspace(a,b,n) se se omite n toma 100 por defeito; o incremento é k=(b-a)/(n-1)

• Com componentes logaritmicamente espaçados logspace (a,b,n) gera um vector linha de n pontos logaritmicamente espaçados entre 10^a e 10^b. Se se omite o valor de n, toma por defeito 50

Vectores e matrizes

Operações com escalares:

v+k adição
v-k subtracção
v*k multiplicação
v/k divide por k cada elemento de v
k./v divide k por cada elemento de v
v.^k potenciação: cada componente de v elevado a k
k.^v potenciação: k elevado a cada componente de v

Operações entre vectores:

v+w adição
v-w subtracção
v.*w multiplicação: cada elemento de v pelo correspondente de w
v./w divide cada elemento de v pelo correspondente de w
v.^w potenciação: cada componente de v elevado ao correspondente de w

Produto escalar de vectores v*w calcula o produto escalar de v por w

Funções Matlab específicas para vectores:

sum(v) soma
prod(v) produto
v' transposição de vectores (linhas $\leftrightarrow$ colunas)
dot(v,w) produto escalar de vectores
cross(v,w) produto vectorial de vectores
[y,k]=max(v) valor máximo das componentes de um vector (k indica a posição) o mesmo para min(v) (valor mínimo)

Exemplo: Cálculo de erros relativos

Suponhamos que para resolver uma equação diferencial ordinária usamos:

-Um método analítico mediante o qual sabemos que a solução num intervalo [0,1] é $y(x)=x^2+\cos(x)$.

-Um método numérico para aproximar a solução no intervalo [0,1] com parâmetro de discretização 0.2

Ângulo (radianos)	Solução aproximada
0	1.0030
0.2	1.0234
0.4	1.0825
0.6	1.1869
0.8	1.3342
1	1.5415

Vectores e matrizes

Para definir uma matriz *não é necessário estabelecer de antemão o seu tamanho* (de facto, pode-se definir um tamanho e mudá-lo posteriormente). O MATLAB determina o número de linhas e de colunas em função do número de elementos que se utilizam. ***As matrizes definem-se por linhas***; os elementos de uma mesma linha estão separados por ***espaços*** ou ***vírgulas***, ao passo que as linhas estão separadas por ***ponto e vírgula*** (;). Por exemplo, o seguinte comando define uma matriz **A** de dimensão (3x3):

» A=[1 2 3; 4 5 6; 7 8 9]

A resposta do programa é a seguinte:

A =
1 2 3
4 5 6
7 8 9

matriz transposta: Em MATLAB o apóstrofo (') é o símbolo de transposição matricial.

matriz inversa: A inversa de **A** calcula-se com a função ***inv( ):***

B=inv(A).

Vectores e matrizes

Em MATLAB acede-se aos elementos de um vector pondo o índice entre parêntesis (por exemplo *x(3) ou x(i)*). Os elementos das matrizes acedem-se pondo os dois índices entre parêntesis, separados por uma vírgula (por exemplo *A(1,2) ou A(i,j)*). As matrizes ***são armazenadas por colunas*** (apesar de se ***introduzirem por linhas***) e tendo em conta isto pode-se aceder a qualquer elemento de uma matriz com um só sub-índice. Por exemplo, se **A é** uma matriz (3x3) obtém-se o mesmo valor escrevendo *A(1,2) ou A(4)*.

Os operadores matriciais em MATLAB são os seguintes:

+ adição
– subtracção
* multiplicação
' adjunta (transposta ou transposta conjugada)
^ potenciação
\ divisão-esquerda
/ divisão-direita
.* produto elemento a elemento
./ e .\ divisão elemento a elemento
.^ elevar a uma potência elemento a elemento

Vectores e matrizes

diag(A) Obtenção da diagonal de uma matriz. sum(diag(A)) calcula o traço da matriz A
diag(A,k) extrai a k-ésima diagonal.

norm(A) norma de uma matriz, calcula o máximo dos valores absolutos dos elementos de A

Geração de matrizes:
- Matriz de zeros, zeros(n,m)
- Matriz de uns, ones(n,m)
- Matriz identidade eye(n,m)
- Matriz de elementos aleatórios rand(n,m)
- Matrizes com diagonal dada diag(v), diag(v,k)
- [X Y] colunas, [X; Y] linhas

Vectores e matrizes

Desde a versão 5 que o Matlab admite variáveis subindiciadas multidimensionalmente

a=ones(2,2,3)

a(:,:,1)= 1 1   a(:,:,2)= 1 1   a(:,:,3)= 1 1

             1 1   1 1   1 1

Matrizes esparsas são aquelas que têm grande qantidade de elementos nulos

>> a= sparse(i,j,c,m,n)

m indica linhas, n colunas, c vector que contém os elementos não nulos e i,j são dois vectores que indicam a posição de cada elemento de c

para visualizar a matriz inteira >>full(a)

[i2,j2,c2]=find(a)

Exemplo1: Tomemos a seguinte matriz

$$\begin{pmatrix} 12 & -4 & 0 & 0 & 0 \\ 7 & 3 & 0 & 0 & -8 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -13 & 11 & 0 \\ 0 & 0 & 2 & 7 & -4 \end{pmatrix}$$

E trabalhemos com ela como uma matriz esparsa. Para tal definimos o vector de elementos não nulos, o vector definido pelas linhas e o vector definido pelas colunas.

Vectores e matrizes

```
m=[12,-4,7,3,-8,-13,11,2,7,-4];
f=[1,1,2,2,2,4,4,5,5,5];
c=[1,2,1,2,5,3,4,3,4,5];
a=sparse(f,c,m,5,5)
full(a)

m1=[12,7,-4,3,-13,2,11,7,-8,-4];
f1=[1,2,1,2,4,5,4,5,2,5];
c1=[1,1,2,2,3,3,4,4,5,5];
b=sparse(f1,c1,m1,5,5];
full(b)

a=sparse(f,c,m)
s=a+b
p=a*b
[f2,c2,m2]=find(p)
e=full(sparse(f2,c2,m2))
```

Exercício 2.1 Dados os vectores definidos por x=(1,4.5,7.8) e y=(sen(90),cos(45),0). Realizar os cálculos seguintes:
x+y; produto escalar de x por y; calcular o ângulo formado por ambos os vectores.

Exercício 2.2 Para um laboratório compram-se os materiais especificados na tabela seguinte:

Ref. artigo	Preço	Quantidade
1520	1146	200
1621	3450	250
1428	6225	150
1429	7100	150
1628	8500	100

Utilizar vectores e o produto de vectores para calcular o custo de cada produto e o total a pagar (a tabela de preços não inclui IVA, devendo portanto aplicar-se a taxa de 19%.

Vectores e matrizes

Exercício 2.3 Dadas as matrizes *a* e *b*, calcular: *a*+*b*, *a*+*0*, inversa de *b*. Comprovar que o produto de matrizes não é comutativo. Seleccionar a submatriz de *a* formada pela primera coluna e terceira linha e a submatriz de *b* formada pela segunda e terceira colunas e calcular o produto

$$a=\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 4 & 2 & 5 \end{pmatrix} \qquad b=\begin{pmatrix} 4 & 2 & 1 \\ 2 & 0 & 4 \\ 1 & -2 & -5 \end{pmatrix}$$

Exercício 2.4 Utilizar matrizes para construír uma tabela que contenha: Na 1ª coluna a variável graus celsius no intervalo [0 100] com um passo de 2. Na 2ª coluna o seu valor em graus Fahrenheit e na 3ª em Kelvin. Como guardaria num ficheiro a tabela anterior?

$$\frac{9*celsius}{5}+32$$

Exercício 2.5 Considere-se a matriz tridiagonal a_{ij} definida por:

$$a_{ij}= \begin{cases} i, \text{ se } i=j \\ j, \text{ se } i=j+1 \\ j, \text{ se } i=j-1 \\ 0, \text{ nos restantes casos} \end{cases}$$

Introduzir a matriz no espaço de trabalho como matriz esparsa para n=10

POLINÓMIOS

Os polinómios são representados em Matlab por um vector linha de dimensão *n+1* sendo *n* o grau do polinómio.

Dado um polinómio

$$x^3+2x$$

este representa-se como

```
>> pol1=[1 0 2 0]
```

para o cálculo das raízes de um polinómio existe o comando roots:

```
>>raizes=roots(pol1)
```

produz um vector coluna, enquanto pol1 é um vector linha

Polinómios

Um polinómio pode ser reconstruído a partir das suas raízes com o comando poly

>> p=poly(raizes) (dá um vector linha)

No caso em que o argumento de poly seja uma matriz obteríamos como resultado o polinómio característico da matriz. Assim se queremos calcular os valores próprios da matriz basta calcular as raízes do polinómio característico.

Exemplo1:

```
pol2=[2 4 0 1]; % definição do polinómio 2x³+4x²+1
raizes=roots(pol2) % cálculo das suas raízes
pol2_n=poly(raizes) % reconstrução do polinómio
real(pol2_n)  % o que acontece?
```

Exemplo2:

```
A=[1 2 3 ; 2 3 4; 4 2 5]; p=poly(A) % pol. característico
roots(p) % valores próprios de A
```

Polinómios

Para calcular o valor de um polinómio *p* num ponto dado *x* basta utilizar o comando polyval

>>y=polyval(p,x)

Exemplo3:

p=[1 –1 –1 1] % definição do polinómio x^3-x^2-x+1

polyval(p,2)

Exemplo4:

x=-2:0.1:2;

Y=polyval(p,x);

Para multiplicar e dividir polinómios temos os comandos especiais conv(p1,p2) e deconv(p1,p2)

Exemplo5:

p1=[1, -2, 1]; p2=[1,1]; p3=conv(p1,p2)

p4=deconv(p3,p2)

[p4,r]=deconv(p3,p2) % resto da divisão

Polinómios

Para conhecer o resto da divisão de polinómios basta escrever

```
>>[p4,r] = deconv(p3,p2)
```

O comando *residue*, permite o cálculo do desenvolvimento em série de fracções simples do quociente p1/p2.
p2 deve ter raízes reais
O formato do comando é: `>>[r,p,k] =residue(p1,p2)`
onde:
r= vector coluna com os numeradores
p= vector coluna com as raízes do denominador
k= vector linha com os coeficientes do polinómio independente.

$$\frac{p1(x)}{p2(x)} = \frac{r(1)}{x-p(1)} + \cdots + \frac{r(n)}{x-p(n)} + k(x)$$

>>[p1,p2]=residue(r,p,k) faz a operação inversa

Exemplo6:

Decompor em fracções simples o quociente

$$\frac{x^3+x^2+1}{x^3-3x^2+4}$$

```
p1=[1 1 0 1]
p2=[1 –3 0 4]
[r,p,k]=residue(p1,p2)
rats(r)
```

$$\frac{x^3+x^2+1}{x^3-3x^2+4}=\frac{35}{9(x-2)}+\frac{13}{3(x-2)^2}+\frac{1}{9(x+1)}+1$$

```
[pol1,pol2]=residue(r,p,k)
```

Para calcular a derivada de um polinómio temos o comando,

```
>>polyder(p)
```

Exemplo7:

Dado o polinómio x^3+6x^2+1 a sua derivada é

```
p=[1, 6,0,1];
d=polyder(p)  %  3x^2+12x
```

Polinómios

Exercício3.1 Consideremos o polinómio $p(x)=x-1$. Calcular $p(x)^3$ e identificar o polinómio obtido e calcular as suas raízes

Exercício3.2 Segundo Hill e Lounasmaa, a equação da curva de inversão do hélio é $P=-21+5.44T-0.132T^2$ onde P vem dada em atmosferas e T em Kelvin.

a) Calcular o valor da pressão a uma temperatura de –268.25 ºC.

b) Calcular o valor da temperatura para uma pressão de $3N/m^2$. Nota $1N/m^2=9.265e-6$ atm.

Exercício 3.3 Calcular a solução geral da EDO

$$y^{(5)}-y^{(4)}+2y'''-2y''+y'-y=0$$

Exercício3.4 Calcular o integral $I=\int \frac{x^4+2x+1}{x-1}\,dx$

Exercício3.1

```
p=[1 –1];
q=conv(p,p)
q=conv(q,p)
raizes=roots(q)
format short
real(raizes)
```

Exercício3.2

a) `pressao=[-0.132 4.55 –21.0];`
   `pressao0=polyval(pressao,273.15-268.25)`

b) `pressao1=3*9.265e-006;`
   `pressao(3)=pressao(3)-pressao1`
   `temp=roots(pressao)`

Polinómios

Exercício3.3

```
format long
p=[1 –1 2 –2 1 –1];
raizes=roots(p)

der_p=polyder(p)
polyval(der_p,round(raizes(2)))
```

A solução exacta é:

$y(x)=c_1e^x+c_2sen(x)+c_3cos(x)+c_4xsen(x)+c_5xcox(x)$

```
dsolve(‘D5y-D4y+2*D3y-2*D2y+Dy-y=0’,’x’)
```

Exercício3.4

```
numerador=[1 0 0 2 1];
denominador=[1 –1];
[cociente,resto]=deconv(numerador,denominador)
```

$$I=\int\left(x^3+x^2+x+3+\frac{4}{x-1}\right)dx=\frac{x^4}{4}+\frac{x^3}{3}+\frac{x^2}{2}+3x+4\ln(x-1)$$

```
int('(x^4+2*x+1)/(x-1)')
```

EQUAÇÕES LINEARES

Dado um sistema algébrico de equações lineares da forma Ax=b, resolvê-lo-emos por métodos clássicos e com funções próprias do Matlab
Alguns comandos cujos argumentos são matrizes e que são úteis para a resolução de sistemas:

```
>>det(A) %determinante de uma matriz quadrada
>>inv(A) %inversa de uma matriz quadrada
>>rank(A) %rank de uma matriz (ordem do maior menor com determinante não nulo)
```

Exemplo1:

```
x=ones(4,4);v=[2 2 2 2];y=x+diag(v)
rank(y)
```

Para ver se um sistema é compatível

Ax=b compatível se

```
>>rank(A)-rank([A,b])
```

Número de condição de uma matriz

```
>>cond(A)
```

este número indica a sensibilidade da solução de um problema em ordem a variações relativas nos dados de entrada. (é a relação entre o maior e o menor valor singular da matriz, deve ser próximo de 1)

Equações lineares

Exemplos de resolução de um sistema Ax=b:

Regra de Cramer

Exemplo2: Dado um sistema Ax=b

```
A=[1,2,3;2,3,4;4,2,5]
b=[4;5;1];
% comprovamos que a matriz não é singular
det(A)
D1=A;D1(:,1)=b
D2=A;D2(:,2)=b
D3=A;D3(:,3)=b
x=[det(D1);det(D2);det(D3)]/det(A)
```

Equações lineares

A solução por Cramer é tediosa e pouco eficaz.

Uma primeira possibilidade mais cómoda.
Se a matriz é quadrada e o seu determinante é diferente de zero:
Ainv=inv(A)
Y o sistema se resolveria como
x=Ainv*b

Outra solução seria utilizar a divisão matricial
X=A\b

A\b, produz um resultado mesmo que a matriz não seja quadrada e não invertível.
Por exemplo se o sistema é sobredeterminado (mais equações que incógnitas) calcula-se a solução que minimiza a norma 2 de Ax-b (solução de mínimos quadrados)

Equações lineares

Com matrizes esparsas

Exemplo3: Consideremos uma matriz tridiagonal de dimensão 32 com a diagonal principal identicamente igual a 4 e as diagonais superior e inferior iguais a 2, e estudamos a variação no número de operações a realizar consoante tratemos a matriz como cheia ou esparsa. Ao resolver o sistema “Ax=b”, com b=d=[1:32]’;

```
% define-se a diagonal inferior
diag=sparse(2:32,1:31,2*ones(1,31),32,32)
% define-se a diagonal principal
a=sparse(1:32,1:32,4*ones(1:32));
% constrói-se a matriz tridiagonal esparsa
a=a+diag+diag';
% constrói-se a matriz cheia
b=full(a)
```

Equações lineares

```
% define-se o segundo membro
d=[1:32]';
```

Valores e vectores próprios de uma matriz.

```
>>eig(A)  (vector coluna)

>>[V,D]=eig(A)
```

Equações lineares

Exercício4.1
Considere a matriz A. Calcular o determinante da matriz A. Resolver o sistema, sendo b um vector coluna igual a
(1 2 3)

$$A=\begin{pmatrix} 6 & 18 & 2 \\ 7 & -2 & -4 \\ 4 & 10 & -6 \end{pmatrix}$$

Substituir na matriz A o elemento da linha 1 coluna 3 por –10 e fazer o mesmo.

Exercício 4.2 Dada a matriz M, comprovar o teorema de Cayley-Hamilton: Toda a matriz é raiz do seu polinómio característico

$$M=\begin{pmatrix} 1 & 1 & -2 \\ -1 & -2 & 0 \\ 3 & 0 & 1 \end{pmatrix}$$

Exercício4.3

Considere a matriz A. Calcular os seus valores e vectores próprios

$$A=\begin{pmatrix} 1 & 2 & 0 \\ 2 & 5 & -1 \\ 4 & 10 & -1 \end{pmatrix}$$

Exercício4.1

```
A=[6 18 2;7 –2 –4;4 10 –6];
d=det(A)
b=[1;2;3];
sol=A\b
A(1,3)=-10
d=det(A)
sol=A\b
A*sol-b
```

Exercício 4.2

```
M=[1 1 –2; -1 –2 0; 3 0 1];
poly(M)
M^3+4*M+13*eye(3)
```

Exercício 4.3

```
A=[1 2 0; 2 5 -1;4 10 -1];
[V,D]=eig(A)
diag(D)
```