

Matlab num Instante
Compacto
Versão 1.4

José Manuel Neto Vieira
Departamento de Electrónica e Telecomunicações e Informática
da
Universidade de Aveiro

3 de março de 2018

Conteúdo

Introdução	3
O que é o Matlab?	3
Ajuda	3
O comando HELP	3
O comando LOOKFOR	3
O “Help Desk”	4
O sistema Matlab	4
Matrizes	4
Criação de matrizes	4
Índices	5
O operador “:”	5
Expressões	6
Variáveis	6
Números	6
Operadores	6
Funções	7
Expressões	7
Polinómios	8
Manipulação de matrizes	8
Criação de matrizes	8
Concatenação de matrizes	9
Remoção de colunas e linhas	10
Gráficos	10
Criação de gráficos	10
Gráficos de variáveis complexas	12
Visualização de funções de duas variáveis	13

Agradecimentos

O autor agradece aos seus colegas Prof.^a Ana Maria Tomé, Prof. António Joaquim Teixeira, Prof. Paulo Jorge Ferreira e Prof. José Luís Azevedo, pelas sugestões e correcções realizadas neste documento.

© 2018, José Manuel Neto Vieira (jnvieira@ua.pt)
<http://sweet.ua.pt/jnvieira/MatlabNumInstante.pdf>

O “Matlab num Instante” pode ser distribuído sujeito às seguintes restrições:

- O seu conteúdo não pode ser alterado.
- Não pode ser utilizado (no todo ou em parte) na venda ou promoção de qualquer produto ou documento.

Matlab num Instante

Compacto

Introdução

O que é o Matlab?

O Matlab é um sistema para cálculo científico que proporciona um ambiente de fácil utilização com uma notação intuitiva mas poderosa. Permite a realização de algoritmos numéricos sobre matrizes com o mínimo de programação. Além disso, no ambiente Matlab é possível a criação e manipulação de matrizes sem a necessidade de dimensionamento prévio e a manipulação das variáveis pode ser realizada de forma interactiva. O termo “Matlab” tem origem na conjugação dos termos “MATrix” e “LABoratory”.

Este documento pretende ser apenas uma introdução muito breve ao Matlab e permitir a um utilizador não “iniciado” começar a dominar os aspectos mais básicos em pouco tempo. A leitura deste documento deve ser realizada ao lado de um PC com o Matlab, para o utilizador poder testar os exemplos e fazer os diferentes exercícios.

Ajuda

O comando HELP

Para esclarecer a maior parte das dúvidas acerca da utilização de uma dada função do Matlab o comando `help` é de grande utilidade. Se se pretender, por exemplo, informação sobre a função `sin`, basta fazer

```
>>help sin
```

obtendo-se a seguinte descrição

```
SIN Sine.
```

```
SIN(X) is the sine of the elements of X.
```

O Matlab possui todas as funções organizadas em grupos e a própria estrutura de directórios onde o Matlab é armazenado em disco reflecte esse facto. Por exemplo, todas as funções de álgebra linear estão armazenadas no directório `matfun`. Para obter uma lista completa deste tipo de funções basta fazer

```
>>help matfun
```

Como não é fácil decorar os nomes de todas as categorias de funções, existe uma janela de ajuda mais organizada, bastando para tal escrever o comando

```
>>helpwin
```

O comando LOOKFOR

Quando se pretende encontrar uma função para resolver um problema, mas desconhece-se se existirá alguma adequada no Matlab, o comando `lookfor` permite pesquisar as primeiras linhas do “help” de todas funções da instalação do Matlab. Esta pesquisa é adequada para resolver a maior parte das situações uma vez que a primeira linha do “help” de uma função contém sempre uma descrição sumária da sua funcionalidade. O seguinte exemplo procura pela palavra “inverse”.

```
>>lookfor inverse
```

Se se pretender que a função `lookfor` pesquise todas as linhas do “help”, pode-se utilizar a opção `-all`, tal como o seguinte exemplo ilustra.

```
>>lookfor -all inverse
```

O “Help Desk”

A partir da versão 5.0, o Matlab vem acompanhado com uma ajuda em formato “html” e todos os manuais em formato “pdf”. Essa informação pode ser encontrada no endereço <http://www.mathworks.com>.

O sistema Matlab

O sistema Matlab é constituído pelas seguintes partes:

- A linguagem
Permite a manipulação e criação de matrizes de forma rápida e intuitiva. Diferentes soluções para um problema podem ser testadas numa fracção do tempo que levaria com outras linguagens (C ou Fortran por ex.). Possui um conjunto muito vasto de funções que permitem resolver problemas complexos de forma eficiente.
- O ambiente de trabalho
O Matlab proporciona um ambiente de trabalho que permite a gestão e visualização das variáveis, ler e gravar variáveis em disco e gerar programas em linguagem Matlab, possibilitando assim a automatização de cálculos complexos.
- Gráficos
As funções de criação, visualização e manipulação de gráficos são muito fáceis de usar e permitem a criação de gráficos 2D e 3D. O ajuste de escala é automático e o utilizador pode começar a utilizar as funções de geração de gráficos pouco tempo depois do primeiro contacto com o ambiente do Matlab.
- “Toolboxes”
O Matlab disponibiliza um conjunto de pacotes de funções para as mais variadas áreas de cálculo científico, sendo estes denominados “toolboxes”. Existem “toolboxes” para estatística, processamento de sinal, processamento de imagem, controlo, cálculo simbólico, etc.

Matrizes

Criação de matrizes

Uma matriz é uma estrutura de dados bidimensional que permite guardar números de uma forma ordenada e indexável. Para criar, por exemplo, a seguinte matriz com duas linhas e três colunas

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix},$$

basta introduzir o comando:

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

ou em alternativa

```
>>A= [1,2,3;4,5,6]
```

Os vectores coluna e linha são casos particulares de matrizes e são criados utilizando a mesma notação. Para criar uma variável com apenas um elemento basta fazer

```
>>a= 10
```

Este é um caso particular de uma matriz de dimensão 1×1 , e pode-se igualmente utilizar a notação mais geral

```
>>a= [10]
```

Índices

O elemento da linha i e da coluna j de uma matriz A é designado por $A(i, j)$. Por exemplo o elemento da linha 1 e coluna 3 da matriz A é designado por $A(1, 3)$. Em notação Matlab, para obter o elemento $A(1, 3)$ definida anteriormente, pode-se escrever

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

e obtém-se

```
ans= 3
```

Para alterar o valor do elemento $A(1, 3)$ para 7 basta fazer

```
>>A(1,3)= 7
```

Os índices das matrizes são listas de números inteiros positivos que podem ser armazenadas em vectores declarados previamente. Se pretendemos por exemplo, extrair a segunda linha da matriz A podemos fazer

```
>>v= A(2, [1 2 3])
```

```
v=
```

```
4      5      6
```

ou declarando primeiro um vector para os índices das colunas

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

```
>>v= A(2,k)
```

```
v=
```

```
4      5      6
```

O operador “:”

A criação de vectores elemento a elemento é bastante morosa e para matrizes de grandes dimensões quase irrealizável. O Matlab permite gerar sequências de números de forma rápida se fizermos uso do operador “:”. Se quisermos gerar o vector $a = [1, 2, 3, \dots, 10]$ podemos fazer

```
>>a= 1:10
```

A notação geral para o operador “:” é a seguinte

Número_inicial : incremento : Número_final

e permite a geração de sequências de números inteiros como no exemplo anterior ou mesmo de números reais. Eis alguns exemplos:

```
>>e= 0:pi/20:2*pi
```

```
>>f= 10:-1:-10
```

O operador “:” pode ser utilizado na geração de vectores de índices obtendo-se uma notação muito compacta. Se quisermos obter as colunas ímpares da matriz

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 4 \\ 6 & 1 & 2 \end{bmatrix},$$

podemos fazer

```
>>B=A(1:3,1:2:3)
```

No caso anterior são indexadas todas as linhas da matriz. Para simplificar a notação, quando não se conhece exactamente o número de linhas de uma matriz, pode-se utilizar a notação

```
>>B=A(:,1:2:3)
```

Se quiséssemos obter a primeira linha da matriz A podíamos fazer

```
>>A(1,:)
```

Expressões

O Matlab permite a construção de expressões matemáticas sem qualquer declaração do formato numérico ou dimensão das matrizes. Existem quatro constituintes básicos nas expressões do Matlab:

- Variáveis
- Números
- Operadores
- Funções

Variáveis

Todas as variáveis do Matlab são do tipo matriz e a sua criação é automática. Por exemplo, o comando

```
>>Custo_total= 1000
```

resulta na criação em memória de uma matriz de 1×1 com o valor 1000. O Matlab distingue as letras maiúsculas das minúsculas nos nomes das variáveis e só toma em consideração os primeiros 31 caracteres. Para visualizar o valor de uma variável basta escrever o seu nome.

Existe uma variável especial que é utilizada pelo Matlab quando não se atribui o resultado de uma expressão a qualquer variável. Esta variável designa-se por “**ans**” (do termo “answer”, resposta) e pode ser utilizada numa sessão interactiva para continuação dos cálculos, tal como o exemplo seguinte demonstra

```
>>2*sin(2)
ans=
    1.8186
>>ans/102
ans =
    0.017829
```

Números

O Matlab utiliza uma notação standard para a representação dos números, admitindo notação científica e números complexos. A unidade imaginária $j = \sqrt{-1}$ é representada no Matlab pelas letras i ou j . Como estas letras podem ser utilizadas no nome de outras variáveis convém ter o cuidado de confirmar o seu valor antes de as usar. A seguir ilustram-se alguns exemplos:

10	-11	0.312
1+i	10j	3e2
10e6j	10e-2	-2.3e-4

Operadores

As operações nas expressões do Matlab seguem as regras habituais de precedência e podem ser aplicadas quer a matrizes quer a números.

+	Adição
-	Subtracção
*	Multiplicação
/	Divisão
^	Potenciação
'	Transposta
()	Parêntesis

Funções

O Matlab possui um conjunto muito grande de funções matemáticas que permitem resolver grande parte dos problemas de cálculo encontrados, tais como: **sin**, **cos**, **tan**, **sqrt**, **log**, etc. Estas aceitam números complexos como argumento e podem também devolver resultados do tipo complexo. Se, por exemplo, se calcular a raiz quadrada de um número negativo não ocorre um erro mas o correspondente número complexo é automaticamente devolvido. Quando as funções são aplicadas sobre matrizes a função é aplicada a cada um dos elementos. Eis uma lista de algumas das funções matemáticas mais comuns e disponíveis no Matlab:

cos	coseno (radianos)	log	logaritmo neperiano (base e)
sin	seno	log10	logaritmo base 10
tan	tangente	rem	resto da divisão inteira
acos	arco-cosseno	abs	valor absoluto
asin	arcoseno	sign	sinal
atan	arco-tangente	round	arredonda para o mais próximo
sqrt	raiz quadrada	floor	arredonda para baixo
exp	exponencial	ceil	arredonda para cima
max	máximo	min	mínimo
real	parte real de um complexo	abs	módulo
imag	parte imaginária de um complexo	angle	argumento de um complexo

Para obter ajuda sobre a utilização de qualquer função basta fazer

```
>>help func
```

em que **func** é o nome da função.

As funções **max** e **min** permitem obter, para além do valor máximo e mínimo de um vector, o índice onde ele ocorre. Vejamos um exemplo:

```
>>a= [1 4 9 2];
>>[m,idx]= max(a)
>> m = 9
>> idx = 3
```

Existem algumas funções especiais que devolvem o valor das constantes matemáticas mais utilizadas:

pi	3.14159265
i	$\sqrt{-1}$
j	$\sqrt{-1}$
eps	Precisão relativa do formato double 2^{-52}
realmin	O menor número real 2^{-1022}
realmax	O maior número real $(2 - \text{eps})2^{1023}$
Inf	Infinito
NaN	“Not-a-Number”

Expressões

Na construção de expressões podemos utilizar operadores, funções e variáveis. Eis alguns exemplos:

```
>>r= (pi+1)/(pi-1)
r =
    1.9339
>>a= abs(4+3i)
a=
     5
>>t= angle(4+3i)
```



```

t=
    0.6435
>>a= 0/0
Warning: Divide by zero.
a =
    NaN
>>x= sqrt(-2)
x =
    0 + 1.4142i
>>x= log(-1)
x=
    0 + 3.1416i

```

Polinómios

Existe no ambiente Matlab um conjunto de funções para manipular polinómios. A definição de polinómios é feita criando vectores cujos elementos são os coeficientes do polinómio ordenados por potência decrescente tal como o exemplo seguinte ilustra. Para representar o polinómio $p(x) = x^2 - 3x + 2$, cria-se o vector

```
p=[1 -3 2];
```

e para calcular as suas raízes existe a função

```

r= roots(p)
r=
    2
    1

```

Para obter os coeficientes de um polinómio a partir das raízes pode-se utilizar a função

```

poly(r)
ans=
    1    -3     2

```

Também é possível calcular o valor de um polinómio num conjunto de pontos utilizando a função `polyval` tal como o exemplo ilustra

```

polyval(p,r)
ans=
    0
    0

```

e que, como se pode ver, calculou o valor do polinómio nos seus zeros.

O produto entre dois polinómios é obtido através da função `conv` (convolução). O exemplo seguinte ilustra o produto entre os polinómios $x^2 + 1$ e $x^3 + x - 1$ dado por $x^5 + 2x^3 - x^2 + x - 1$

```

p1= [1 0 1]
p2= [1 0 1 -1]
conv(p1,p2)
ans=
    1  0  2 -1  1 -1

```

Manipulação de matrizes

Esta secção introduz alguns conceitos elementares sobre a criação e manipulação de matrizes.

Criação de matrizes

As cinco funções seguintes permitem a criação de algumas matrizes elementares:

• zeros	Cria uma matriz preenchida com zeros
• ones	Cria uma matriz preenchida com uns
• eye	Cria a matriz identidade
• rand	Cria uma matriz de números aleatórios com distribuição uniforme $[0 \dots 1]$
• randn	Cria uma matriz de números aleatórios com distribuição normal de média nula e desvio padrão 1.

Vejam os alguns exemplos de utilização destas funções:

```
>>Z= zeros(2,5)
Z =
    0    0    0    0    0
    0    0    0    0    0
>>a= ones(2,3)*3
a =
    3    3    3
    3    3    3
>>n= round(3*rand(1,10))
n =
    3    1    2    1    3    2    1    0    2    1
>>A= randn(1,3)+j*randn(1,3)
A =
   -0.43256   +   0.28768i   -1.6656   -   1.1465i   0.12533   +   1.1909i
```

Concatenação de matrizes

Concatenar matrizes consiste em formar matrizes a partir de outras mais pequenas. A notação é idêntica à utilizada para formar matrizes com números. Os seguintes exemplos ilustram a concatenação de matrizes.

```
>>a= [1 2; 3 4]
a =
    1    2
    3    4
>>A= [a a; a a]
A =
    1    2    1    2
    3    4    3    4
    1    2    1    2
    3    4    3    4
>>b= (1:4)'
b =
    1
    2
    3
    4
>>B= [b b [a;a]]
B =
    1    1    1    2
    2    2    3    4
    3    3    1    2
    4    4    3    4
```

Remoção de colunas e linhas

É possível remover de uma dada matriz qualquer conjunto de linhas e colunas. Para tal, basta atribuir o valor de uma matriz vazia definida por “[]” às linhas e colunas que se pretende remover. No exemplo que se segue, elimina-se a 2ª coluna da matriz *A*.

```
>>a= [1 2; 3 4];
>>A= [a a; a a]
A =
     1     2     1     2
     3     4     3     4
     1     2     1     2
     3     4     3     4
>>A(:,2)= []
A =
     1     1     2
     3     3     4
     1     1     2
     3     3     4
```

A remoção de um elemento isolado de uma matriz não é possível uma vez que esta deixaria de respeitar as propriedades de uma matriz

```
>>A(1,2)=[]
??? Indexed empty matrix assignment is not allowed.
```

Gráficos

A geração de gráficos no Matlab representa um dos seus aspectos mais úteis. Nesta secção serão apresentados alguns dos comandos mais importantes para a criação de gráficos a partir dos valores armazenados em matrizes.

Criação de gráficos

A função `plot` é a mais utilizada no Matlab para gerar gráficos variando o seu comportamento consoante os parâmetros de entrada. A sua forma mais simples consiste em passar como entrada apenas um vector:

```
>>plot(y)
```

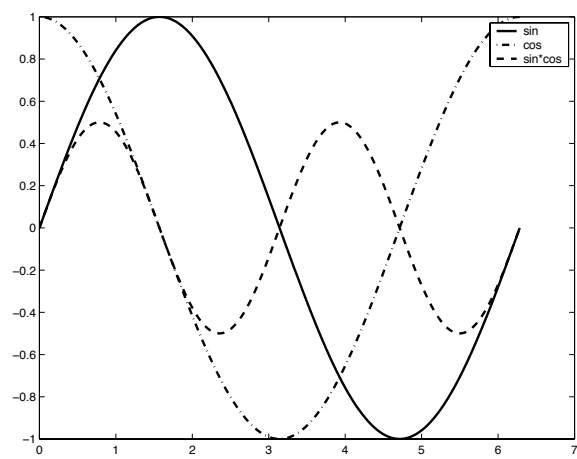
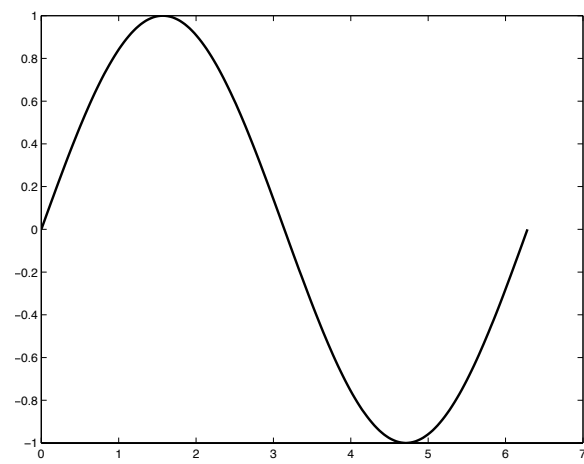
O gráfico gerado apresenta em abcissas os índices *i* dos elementos do vector e em ordenadas o valor de cada um dos elementos do vector. Também é possível utilizar um segundo vector para o eixo das abcissas tal como no exemplo seguinte:

```
>>w=0:pi/100:2*pi;
>>x= sin(w);
>>plot(w,x)
```

A função `plot` admite a representação simultânea de várias curvas, acrescentando mais argumentos de entrada, devendo os vectores possuir o mesmo número de amostras. Vejamos um exemplo:

```
>>w=0:pi/100:2*pi;
>>x1= sin(w);
>>x2= sin(w+pi/2);
>>x3= x1.*x2;
>>plot(w,x1,w,x2,w,x3)
>>legend('sin','cos','asin*cos')
```

A função `plot` permite escolher o tipo de linha, a cor, etc; e existem ainda funções para por exemplo, acrescentar etiquetas aos eixos, criar uma grelha, etc, (ver o “help” da função `plot` para referência). Destas funções destacam-se as seguintes:



<code>xlabel</code>	Etiqueta do eixo das abcissas
<code>ylabel</code>	Etiqueta do eixo das ordenadas
<code>title</code>	Nome da figura
<code>legend</code>	legenda com o significado de cada linha
<code>figure</code>	Cria uma janela nova
<code>figure(gcf)</code>	Coloca a janela de gráfica corrente à frente
<code>zoom</code>	Para aumentar zonas de um gráfico
<code>grid</code>	Grelha
<code>axis</code>	Define os limites dos eixos do gráfico

O Matlab disponibiliza uma pequena função que permite gerar um certo número de valores num dado intervalo. Se se pretender por exemplo, gerar 100 pontos no intervalo de $-\pi$ a π :

```
>>x= linspace(-pi,pi,100);
```

Para além da função `plot`, o Matlab tem ainda várias funções de desenho de gráficos, que se utilizam de forma semelhante mas cujo resultado é diferente. Na tabela em baixo podemos ver alguns exemplos:

<code>semilogx</code>	Gráfico xy com a escala do eixo horizontal logarítmica
<code>semilogy</code>	Gráfico xy com a escala do eixo vertical logarítmica
<code>loglog</code>	Gráfico xy com a escala dos dois eixos logarítmicas
<code>bar</code>	Gráfico de barras verticais
<code>barh</code>	Gráfico de barras horizontais

Para gerar os vectores para os gráficos com escalas logarítmicas convém gerar vectores cujo espaçamento entre pontos seja igualmente logarítmico. A função `logspace` facilita esta tarefa e tem a seguinte sintaxe: os dois primeiros argumentos são a potência de base 10 dos pontos inicial e final respectivamente, o terceiro argumento é o número de pontos. Vejamos um exemplo. Se se pretender gerar 15 pontos entre 10 e 100 com espaçamento logarítmico, pode-se utilizar o comando

```
>>x= logspace(1,2,15);
```

Note-se que $10^1 = 10$ e que $10^2 = 100$.

Gráficos de variáveis complexas

Quando o argumento ds função `plot` é complexo, em abcissa é colocada a parte real do vector e em ordenadas a parte imaginária. Pode-se pensar neste processo como uma forma compacta de representação. O comando que se segue

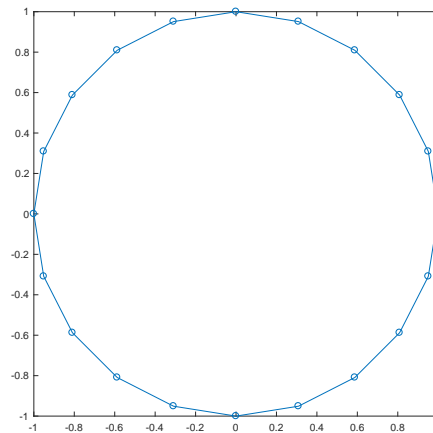
```
>>plot(z)
```

é equivalente a fazer

```
>>plot(real(z),imag(z))
```

Vejamos um exemplo em que se utiliza a função `axis square`, para que a figura obtida seja uma circunferência.

```
>>w= 0:pi/10:2*pi;
>>plot(exp(i*w),'-o')
>>axis square
```



Visualização de funções de duas variáveis

Para visualizar uma função de duas variáveis da forma $z = f(x, y)$ o Matlab possui a função `mesh`. Para utilizar esta função é necessário gerar duas matrizes X e Y em que a primeira possui colunas idênticas e a segunda linhas idênticas, com os valores de x e y em que se pretende calcular a função f . A função `meshgrid` gera as matrizes X e Y a partir dos valores iniciais, finais e respectivo incremento. O seguinte exemplo ilustra a forma de gerar um gráfico 3D da função $\sin(r)/r$, com $r = \sqrt{x^2 + y^2}$.

```
>> [X,Y] = meshgrid(-8:.5:8);
>> R = sqrt(X.^2 + Y.^2) + eps;
>> Z = sin(R)./R;
>> mesh(X,Y,Z)
```

