Instituto Politécnico de Leiria

Escola Superior de Tecnologia e Gestão

Matemática Discreta - Componente PL

EI (D+PL)

Ano letivo 2018/2019 - 2.º Sem. Ficha prática 2

Matrizes e vetores

No Scilab existem várias formas de criar matrizes e operar com as mesmas. Nesta ficha prática são explorados métodos para definir e manipular matrizes, aceder aos seus elementos ou realizar operações elemento a elemento.

- De seguida apresentam-se os símbolos necessários à criação de uma matriz de uma forma simples:

parêntesis retos	[ ]	Indicam o início e o fim da matriz
vírgula	,	Separa elementos pertencentes à mesma linha
ponto e vírgula	;	Indica a quebra/passagem para a linha seguinte

Exemplo:

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

Uma forma alternativa de definir uma matriz `m×n` sem fazer uso da vírgula ou do ponto e vírgula consiste em separar os elementos da mesma linha usando o *space* e definir a mudança de linha usando o *return*:

```
A=[a11 a12 ...  a1n
    a21 a22 ...  a2n
    ...
    am1 am2 ...  amn]
```

Exemplo:

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

- No seguinte quadro são apresentadas funções que permitem a criação de certos tipos de matrizes:

Matrizes específicas	
`eye(m,n)`	Matriz $m \times n$ com 1's na diagonal e 0's nas restantes posições
`ones(m,n)`	Matriz $m \times n$ com todos os elementos iguais a 1
`zeros(m,n)`	Matriz $m \times n$ com todos os elementos iguais a 0

Exemplo:

```
-->A = ones(2,3)
A =
1.  1.  1.
1.  1.  1.
```

- No seguinte quadro são apresentadas funções que permitem obter informação sobre uma matriz ou criar uma nova matriz a partir de outra:

`A`	Apresenta a matriz `A`
`size(A)`	Permite saber a dimensão (n.^{o} de linhas e n.^{o} de colunas) da matriz `A`
`matrix(A,m,n)`	Cria uma nova matriz a partir de `A`, redistribuindo os k elementos desta matriz por m linhas e n colunas (de notar que m×n tem que ser igual a k)
`resize_matrix(A,m,n)`	Cria uma nova matriz com dimensão m×n, truncando as linhas e/ou colunas da matriz original `A`, ou completando as linhas/colunas em falta com 0's

Exemplos:

```
-->B=eye(3,3)
 B=
 1.  0.  0.
 0.  1.  0.
 0.  0.  1.

-->size(B)
 ans =
 3.  3.

-->C=resize_matrix(B,3,2)
 C=
 1.  0.
 0.  1.
 0.  0.
```

- No seguinte quadro são apresentadas algumas funções que permitem trabalhar com determinado elemento, determinada linha/coluna ou determinada parte de uma matriz:

`A(i,j)`	Apresenta o elemento da matriz `A` que se encontra na linha i
	e coluna j
`A(i,j)=k`	Substitui o elemento da matriz `A` que se encontra na linha i
	e coluna j por k
`A(:,:)`	Apresenta a matriz `A`
`A(i,:)`	Apresenta a linha i da matriz `A`
`A(:,j)`	Apresenta a coluna j da matriz `A`
`A(i:j,k)`	Apresenta os elementos da matriz `A` que se encontram na coluna k, da linha i até à linha j
`A(i,:)=[]`	Faz com que a linha i da matriz `A` seja truncada

Exemplos:

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

```
-->A(2 ,1)
ans =
4.

-->A(12 ,1)
!-- error 21
Invalid index .

-->A(2,:)
ans=
4.  5.  6.

-->A(:,3)
ans=
3.
6.

-->A =[ 1 2 3
4 5 6
7 0 0]

A=
 1.  2.  3.
 4.  5.  6.
 7.  0.  0.

A(2,3)=9
A=
 1.  2.  3.
 4.  5.  9.
 7.  0.  0.

-->A(:  ,3) = []
A =
1.  2.
4.  5.
7.  0.

-->B = matrix (A ,1 ,6)
B =
1.  4.  7.  2.  5.  0.
```

- No Scilab podem ser aplicados a matrizes vários operadores: a adição, a subtracção, a multiplicação e a potência (os operadores usuais).

Operadores com matrizes	
+	Adição
−	Subtração
*	Multiplicação
^	Potência

Exemplo:

```
A=[3,-24,30];
B=[
 9 -36 30
 -36 192 -180
 30 -180 180
];

-->A*B
ans =
1791.  - 10116.  9810.
```

- Além destes, existem as operações de multiplicação e potenciação elemento a elemento. Neste caso, deve ser adicionado um ponto antes do símbolo do operador. Por exemplo, sendo X e Y matrizes da mesma dimensão, a multiplicação elemento a elemento significa que o resultado de $Z = X.*Y$ será dado por

$$Z(i,j) = X(i,j) * Y(i,j)$$

  e não pela regra habitual da multiplicação de matrizes. Estas operações têm vantagens, por exemplo, quando é necessário realizar operações sobre matrizes de grande dimensão.

Operadores aritméticos sobre matrizes com a mesma dimensão (elemento a elemento)	
`.*`	Multiplicação elemento a elemento
`.^`	Potência elemento a elemento

Funções Matriciais e Álgebra Linear Numérica	
`A'`	Transposta da matriz `A`
`rank(A)`	Característica da matriz `A`
`det(A)`	Determinante da matriz quadrada `A`
`trace(A)`	Soma dos elementos da diagonal principal da matriz quadrada `A`
`inv(A)`	Inversa da matriz quadrada `A` de determinante não nulo

Exemplos:

```
A=[3,-24,30];
B=[ 9 -36 30
 -36 192 -180
 30 -180 180];

-->det(B)
ans =
 2160.

-->rank(B)
ans =
 3.
```

Programação

O Scilab contém diversos comandos para controlar a sequência dos programas, como por exemplo o teste de condições `if` ou os ciclos de controlo. Os ciclos permitem repetir instruções um número pré-determinado de vezes ou até que uma condição se verifique.

- Instrução Condicional: **If** ... **else** ... **end**

O teste `if` avalia uma proposição e executa um grupo de instruções se essa proposição for verdadeira. Se a proposição tiver valor lógico falso, as instruções a executar podem opcionalmente ser definidas através do comando `else`. O comando `end` termina o teste. A sintaxe do teste `if` é:

```
if expressão lógica 1,
    instruções executáveis se a expressão lógica 1 for verdadeira
else
    instruções executáveis se a expressão lógica 1 for falsa
end
```

Exemplo:

```
a=cos(%pi/5)+ log(1/2)-exp(2);
if a>0
    s=a;
else
    s=0;
end
```

- Ciclo de controlo: **for** ... **end**

Este ciclo consiste em atribuir um valor inicial a uma variável, executar instruções especificadas e incrementar a variável, ou em 1 (por defeito), ou em determinado passo. O ciclo repete-se até que seja atingido ou ultrapassado um valor definido inicialmente. O ciclo termina com a instrução `end`. A sintaxe do ciclo `for` é:

```
for índice = início:incremento:fim
    instruções executáveis
end
```

Exemplo:

```
for i=1:3
    for j=1:3
      H(i,j)=1/(i+j-1);
    end
end
```

- Ciclo de controlo: **while** ... **end**

Neste ciclo de controlo as instruções especificadas são executadas enquanto a expressão lógica for verdadeira. A sua sintaxe é:

```
while expressão lógica,
    instruções executáveis
end
```

Exemplo:

```
s=1; i=0;
while s < 19/10
    s = s + 2^(-i)
    i=i+1;
end
```

Exemplos de outros comandos relacionados com ciclos: `pause`, `resume`, `return e abort` (ver HELP).

Funções

- Quando for necessário definir uma função, teremos em consideração a seguinte estrutura:

```
out_var=myfunction(in_var)
```

onde:
* `out_var` é o nome da variável que contém os dados de saída (por exemplo `y`)
* `myfunction` é o nome da função (por exemplo `f`)
* `in_var` é o nome da variável que contém os dados de entrada (por exemplo `x`)
Assim:

```
y=f(x)
```

Caso a função tenha `n` dados de entrada e `m` dados de saída, devemos ter em consideração a seguinte forma:

```
[out_1,...,out_n]=myfunction(in_1,...,in_m)
```

- A função é implementada através dos comandos `function` e `endfunction`.

```
function [lista de parâmetros de saída] = nome_função(lista de parâmetros de entrada)
instruções
endfunction
```

Exemplo:

```
function y = f(x)
y = 2*x
endfunction
```

Exemplo: Função soma de dois números: nome da função **fun**, parâmetros de entrada **a** e **b** e parâmetro de saída **y**.

```
function y = fun(a,b)
//fun(a,b) - Soma de a + b
y=a+b;
endfunction
```

Exercício: Guarde a função que implementou anteriormente num ficheiro SciNotes designado por *Exemplo_função.sce.*

Estas formas de definir uma função podem ser implementadas, quer na janela de comandos, quer num ficheiro ".sce". Neste último caso, é relevante salientar que, antes da função ser usada, é necessário executar o ficheiro onde a função foi definida.

```
-->exec ("C:\...\Exemplo_função.sce")
--> f(6)
 ans =
 12.
```

Visualização gráfica

- O Scilab tem muitas opções relativas à criação e edição de figuras e gráficos, sejam estes em $2D$ ou em $3D$. Os gráficos em $3D$ podem ser gerados usando o comando `surf` (ver help), enquanto que, para criar gráficos em $2D$, usamos o comando `plot`. Este permite diferentes possibilidades de utilização.

`plot(Y)`	Representa a lista de pontos (X, Y) cujas ordenadas são os elementos do vetor Y e as abcissas são os índices de 1 a n, onde n é o número de elementos do vetor Y
`plot(X,Y)`	Representa a lista de pontos (X, Y) cujas abcissas são os elementos do vetor X e as ordenadas são os elementos do vetor Y
`plot(X,Y,'S')`	Igual ao `plot(X,Y)` onde `S` é uma sequência de caracteres constituída por elementos das colunas da tabela seguinte, escritos entre plicas

Nota: O vector `X` pode ser definido usando qualquer um dos comandos de definição de vectores estudados anteriormente.

Caso tenha sido definida uma função `f`, o vector `Y` pode ser definido pelo comando `Y=f(X).`

A seguinte tabela apresenta os caracteres correspondentes às várias características do gráfico que são possíves de definir através da string `'S'`, usando o comando `plot(X,Y,'S'):`

Símbolo	**Cor**	**Símbolo**	**Marcador**	**Símbolo**	**Tipo de linha**
`b`	azul	`.`	ponto	`-`	linha contínua
`g`	verde	`o`	círculo	`---`	linha tracejada
`r`	vermelho	`x`	X	`-.`	traço e ponto
`c`	ciano	`+`	+	`:`	linha pontilhada
`m`	magenta	`*`	asterisco		
`y`	amarelo	`s`	quadrado		
`k`	preto	`d`	losango		
`w`	branco	$\vee$	triângulo para baixo		
		$\wedge$	triângulo para cima		
		$<$	triângulo para a esquerda		
		$>$	triângulo para a direita		
		`p`	pentagrama		

- O Scilab permite personalizar os gráficos, com diversas opções relacionadas com o controle dos eixos e com a inserção de texto em gráficos. Para saber como funciona cada uma das opções apresentadas a seguir consulte o menu HELP.

Inserção de texto em gráficos	
`legend`	Legenda das curvas do gráfico
`title`	Título do gráfico
`xlabel`	Legenda do eixo dos xx
`ylabel`	Legenda do eixo dos yy
`xtitle("T1","T2","T3")`	Adiciona o título do gráfico (`T1`), o título do eixo dos xx (`T2`) e o título do eixo dos yy (`T3`)

O controle das definições dos eixos e outras definições do gráfico pode ser feito diretamente através do editor da janela gráfica (Graphic Window).

No seguinte quadro são apresentados comandos usados na definição da janela gráfica:

`figure(n)`	Esboça o gráfico da função na janela `n`
`clf`	Apaga as figuras anteriores

Exemplo:

```
function y = f(x)
   y = x.^2
endfunction
xdata = linspace (1,10,50);
ydata = f(xdata);
plot (xdata,ydata,"+ g")
xtitle ("Função quadrática ", "eixo dos xx", "eixo dos yy");
legend ("y=x^2");
```

Exercícios

1. Considere as seguintes matrizes:

$$A = \begin{pmatrix} 2 & 1 & -1 \\ -1 & 2 & 1 \\ 1 & 3 & -3 \end{pmatrix}, \quad B = \begin{pmatrix} 3 \\ -1 \\ 0 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & -2 & 2 \\ 0 & 2 & 3 & 0 \\ 1 & 0 & -1 & 0 \end{pmatrix}, \quad D = \begin{pmatrix} 1 & -1 & 0 & 2 \end{pmatrix}.$$

   (a) Determine, se possível:

(i)	AB;	(ii)	CD;	(iii)	CD^T;	(iv)	AB^T;
(v)	$A^{-1}C$;	(vi)	$3AB + 2CD^T$;	(vii)	$A.\hat{}2$;	(viii)	A^2;
(ix)	$\det(AA^T)$;	(x)	BDC^T;	(xi)	A^{-2};	(xii)	A^2B^T.

   (b) Obtenha todos os elementos de todas as linhas que se encontram entre as colunas 2 e 3 da matriz C.

2. Faça a representação gráfica da função polinomial definida por

$$f(x) = x^5 - 3x^4 - 3x^3 + 7x^2 + 6x$$

   no intervalo $[-1,5; 2,5]$ com incremento de $0,125$.

3. Seja f a função definida por $f(x) = \cos(x) + e^x$.

   (a) Defina/implemente a função f.

   (b) Determine o valor de $f(0)$ e de $f(\pi)$.

   (c) Construa um vetor linha X cujo primeiro elemento seja `1`, o último seja `2` e tenha um total de 98 elementos.

   (d) Calcule $f(X)$. O que acontece?

   (e) Faça a representação gráfica da função f

4. O polónio tem uma meia-vida de 140 dias, o que significa que, devido ao decaimento radioactivo, a quantidade de polónio restante depois de 140 dias é metade da original. A quantidade restante, após um certo período de tempo t, é dada por

$$r(t) = C_0(0.5)^{t/v}, \quad \text{(t em dias)}$$

   sendo C_0 a quantidade inicial e v o tempo de meia-vida. Se hoje tivermos 10 gramas de polónio, qual a quantidade restante ao fim de cada uma das próximas 10 semanas? Elabore um gráfico que mostre o comportamento observado ao longo desse período.

5. Considere a sucessão Z definida por $Z(n) = \sum_{k=1}^{n} \left(\frac{1}{2}\right)^k$. Usando comandos do Scilab:

   (a) faça uma representação gráfica de Z onde se possam visualizar os seus 20 primeiros termos.

   (b) determine o primeiro valor de n tal que seja verificada a condição $|Z(n) - Z(n-1)| < 10^{-10}$.

6. Elabore um algoritmo que permita determinar a nota final (por avaliação periódica) de um estudante na UC de Matemática Discreta. Os elementos de avaliação, os respectivos pesos e a nota mínima estão descritos na tabela seguinte:

	Nº de provas	Peso	Nota mínima
Componente Teórica	2 provas ($PT1, PT2$)	60%	8.0 na média das 2 provas teóricas
Componente prática	1 prova (PP)	40%	8.0 na prova prática única

   Apenas devem ser pedidas ao utilizador as notas parciais.

Algoritmo - Pseudo-código

- Dados de entrada: $nota\_PT1$, $nota\_PT2$ e $nota\_CP$
- Dados de saída: classificação final.
- Definir $nota\_CT = (nota\_PT1 + nota\_PT2) * 0.5$
- Se $nota\_CT < 8.0$ ou $nota\_CP < 8.0$ então
  - "Não atingiu o mínimo numa das componentes."
- Caso contrário
  - $resultado = 0.60 * nota\_CT + 0.40 * nota\_CP$.
  - "A classificação final na UC é *resultado*."