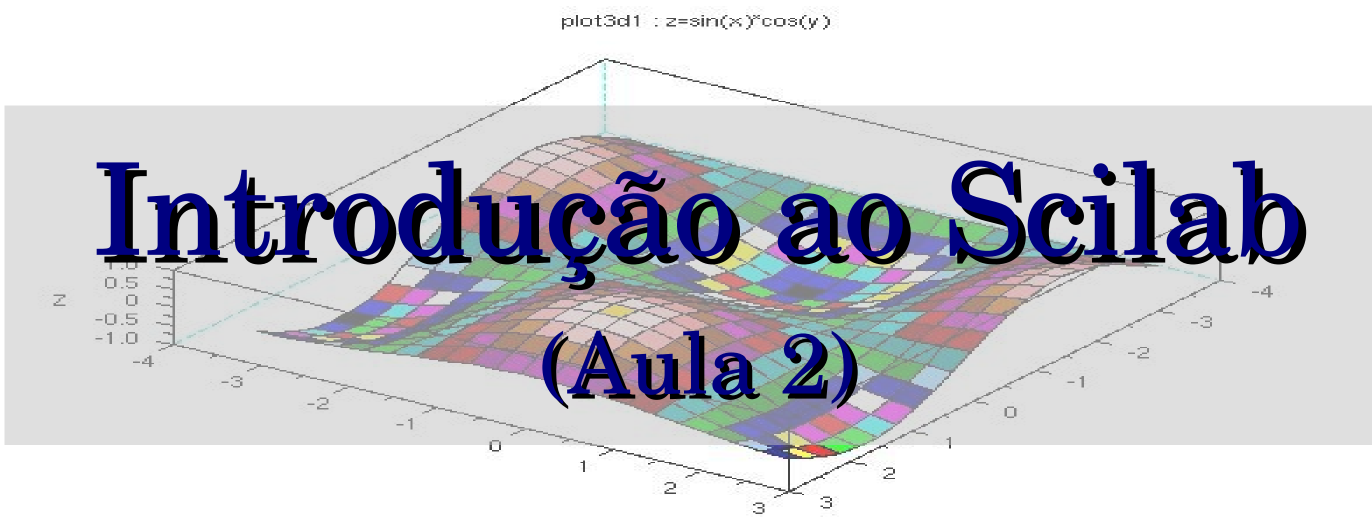
Universidade do Estado do Rio de Janeiro
Faculdade de Engenharia
Laboratório de Engenharia Elétrica



Introdução ao Scilab

(Aula 2)

Elaine de Mattos Silva

Orientador: Prof. José Paulo Vilela Soares da Cunha

Abril de 2007

Apoio: Programa de Estágio Interno Complementar do CETREINA/SR-1/UERJ

Contatos

- E-mail:
  - elaine@lee.eng.uerj.br

- Página do curso:
  - http://www.lee.eng.uerj.br/~elaine/scilab.html

- Apostila *Introdução ao Scilab versão 3.0*:

  Prof. Paulo Sérgio da Motta Pires (UFRN)

  http://www.dca.ufrn.br/~pmotta

Conteúdo Geral

- Aula 1

    O que é o Scilab

    Principais Características do ambiente Scilab

    Operações Básicas

- Aula 2

    Polinômios, Vetores e Matrizes

- Aula 3

    Listas

    Programação com Scilab

- Aula 4

    Gráficos em Scilab

    Introdução ao Scicos

Aula 2

- Aula 2

1 – Polinômios

1.1 - Definição

1.2 - Formas de declaração

1.3 - Operações com polinômios

2 – Vetores

2.1 - Definição

2.2 - Declarando vetores

2.3 - Operações com vetores

Aula 2 (cont.)

- Aula 2

3 – Matrizes

3.1 - Definição

3.2 - Formas de declaração

3.3 - Operações com matrizes

3.4 - Acesso a elementos

3.5 - Matrizes com polinômios

3.6 – Matrizes racionais

3.7 - Matrizes simbólicas

3.8 - Operadores Especiais

1 - Polinômios

1.1 - Definição

$$P(x) = a_n + a_{n-1}x + \ldots + a_2x^{n-2} + a_1x^{n-1} + a_0x^n$$

1 – Polinômios (cont.)

1.2 - Formas de declaração

Ex.: $p = s^2 - 3s + 2$

forma fatorada: $p = (s - 1)(s - 2)$

- Usando a função *poly()*
  - Pelas raízes:

    p = poly([1 2], 's')
  - Pelos coeficientes:

    p = poly([2 -3 1], 's' , 'coeff ')

Obs.: os coeficientes devem ser digitados do menor grau para o maior.

1 – Polinômios (cont.)

1.2 - Formas de declaração

- Usando a variável %s

```
p = %s^2-3*%s+2
```

- Declarando um polinômio x

```
x = poly(0, 'x')
p = x^2-3*x+2
```

1 – Polinômios (cont.)

1.2 - Formas de declaração

```
-->p=poly([1 2],'s')                          //pelas raizes
 p  =
                 2
    2 - 3s + s

-->p=poly([2 -3 1],'s','coeff')               //pelos coeficientes
 p  =
                 2
    2 - 3s + s


-->x = poly(0, 'x');p = x^2-3*x+2
 p  =
                 2
    2 - 3x + x
```

1 – Polinômios (cont.)

1.3 - Operações com polinômios

- A função *roots()* calcula as raízes da função polinômio

  ex.: *roots(p)*

- A função *horner()* calcula o valor da função polinômio

  ex.: *horner(p,2)*

1 – Polinômios (cont.)

1.3 - Operações com polinômios

Ex.: $p = s^2 - 3s + 2$:

```
-->p=poly([2 -3 1],'s','coeff')
 p  =
                2
    2 - 3s + s

-->roots(p)          //calcula raizes
 ans  =
    1.
    2.

-->horner(p,2)       //substitui s por 2 (uma raiz)
 ans  =
    0.
```

1 – Polinômios (cont.)

1.3 - Operações com polinômios (mesma variável)

Ex.: p=poly([2 -3 1], 's' , 'coeff ') ; // $p = s^2 - 3s + 2$

q=poly([1 0 2],'s' , 'coeff '); // $q= 2s^2 + 1$

- Operações básicas:
  - p + q
  - p - q
  - p * q
  - p / q // não efetua a divisão, apenas gera fração racional
  - pdiv(p,q) // efetua a divisão e calcula quociente e resto

1 – Polinômios (cont.)

1.3 - Operações com polinômios (mesma variável)

```
-->p=poly([2 -3 1],'s','coeff');
-->q=poly([1 0 2],'s','coeff');
-->p/q                        //fração racional
ans  =
              2
    2 - 3s + s
    ----------
            2
      1 + 2s

-->[r,q]=pdiv(p,q)           // r=resto q=quociente
 q  =
    0.5
 r  =
    1.5 - 3s
```

2 – Vetores (cont.)

2.1 - Definição

- Diz-se que **x** é um vetor de dimensão n em R se:

$$\boldsymbol{x} = (x_{1,} x_{2,} \ldots, x_n)$$

Mas, ao invés de pensar em coordenadas, pode-se pensar em matrizes-coluna:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{bmatrix}$$

2 – Vetores (cont.)

2.2 - Declarando vetores (seqüências)

A = valor_inicial:incremento:valor_final

```
-->A=0:2:10              //incrementa de 2 em 2
 A  =
    0. 2. 4. 6. 8. 10.

-->b=0:10                //incrementa de 1 em 1 (padrao)
 b  =
    0. 1. 2. 3. 4. 5. 6. 7. 8. 9. 10.

-->C=10:-2:0             //decresce de 2 em 2
 C  =
    10. 8. 6. 4. 2. 0.
```

2 – Vetores (cont.)

2.2 - Declarando vetores (vetor coluna)

- Consideremos o vetor $A = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix}$. Seus elementos devem ser separados por ' ; ' (ponto e vírgula) no Scilab.

Ex.:

```
-->A=[4;5;6]
 A  =


    4.
    5.
    6.
```

2 – Vetores (cont.)

2.2 - Declarando vetores (vetor linha)

- $A = [a_{11}, a_{12}, a_{13}]$ ou $A=[a_{11}\ \ a_{12}\ \ a_{13}]$
  - Obs.: os elementos são separados por ' , ' (vírgula) ou espaço

Ex.:

```
-->A=[4,5,6];B=[4 5 6]; //elementos separados por (,)

-->A,B
 A  =
    4.    5.    6.
 B  =
    4.    5.    6.
```

2 – Vetores (cont.)

2.3 - Operações com vetores

- Transposição
  - A'

Ex.:
```
-->A=[1 2 3]    //vetor linha
 A  =
    1.  2.  3.

-->A'           //vetor coluna
 ans  =
    1.
    2.
    3.
```

2 – Vetores (cont.)

2.3 - Operações com vetores

- Dimensão
  - *length(x)*

    A função length(x) retorna a dimensão de um vetor.

Ex.:
```
-->A=[1,2,3];

-->length(A)
 ans  =

     3.
```

2 – Vetores (cont.)

2.3 - Operações com vetores

- Elementos iguais a 1
  - A= *ones* (4,1)

    Gera vetor com todos os elementos iguais a 1.

- Vetor nulo
  - B= *zeros* (4,1)

    Gera vetor com todos os elementos iguais a 0.

2 – Vetores (cont.)

2.3 - Operações com vetores

- Operações Básicas
  - adição e subtração (se dois vetores possuem **mesma dimensão**):
    - A+B
    - A-B
  - multiplicação e divisão por escalar:
    - 3*A
    - A/2

2 – Vetores (cont.)

2.3 - Operações com vetores

Ex.:

Sejam $x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ e $y = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$ calcule x+y :

```
-->x=[1;2;3];y=[4;5;6];   //declarando vetores

-->x+y
 ans  =
     5.
     7.
     9.
```

2 – Vetores (cont.)

2.3 - Operações com vetores

- Multiplicação de Vetores

  - Produto interno (produto escalar)

Se dois vetores possuem **mesma dimensão**, define-se produto escalar entre x e y:

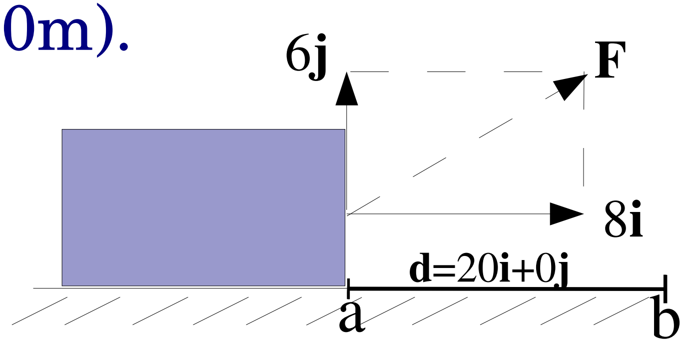
$$z = x^T y = (x_1 * y_1 + x_2 * y_2 + \ldots + x_n * y_n)$$

Ex.:
```
-->x=[1;2;3];y=[4;5;6];

-->z=x'*y
 z  =

    32.
```

2 – Vetores (cont.)

Exercício

• Calcular o trabalho realizado pela força **F**= 8**i** + 6**j** para deslocar o corpo de a até b (20m).



Podemos decompor a distância em d= 20**i** + 0**j**

Lembre-se que (W=F •d)

3 – Matrizes

3.1 - Definição

- Uma matriz geral consiste em *mn* números dispostos em *m* linhas e *n* colunas:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

3 – Matrizes (cont.)

3.2 - Formas de Declaração

Ex.: $$\boldsymbol{M} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

M=[1 2 3; 4 5 6; 7 8 9]

M=[1, 2, 3; 4, 5, 6; 7, 8, 9]

M=[1 2 3 *<enter>*
4 5 6 *<enter>*
7 8 9]

3 – Matrizes (cont.)

3.2 - Formas de Declaração

Funções para geração de matrizes:

- *ones (m,n)* - matriz com todos os elementos iguais a 1
- *zeros(m,n)* - matriz nula

Ex.:
```
-->A=ones(1,2)
 A  =
      1.    1.
-->B=zeros(3,2)
 B  =
     0.    0.
     0.    0.
     0.    0.
```

3 – Matrizes (cont.)

3.2 - Formas de Declaração

Funções para geração de matrizes:

- *eye(m,n)* – matriz identidade

Ex.:
```
-->C=eye(4,4)
 C  =
1.     0.     0.     0.
0.     1.     0.     0.
0.     0.     1.     0.
0.     0.     0.     1.
```

3 – Matrizes (cont.)

3.2 - Formas de Declaração

- Podemos gerar matrizes a partir de elementos de outras matrizes:

Ex.:

```
-->D=[1 2;3 4]
 D  =
    1.    2.
    3.    4.
-->E=[5 6;7 8]
 E  =
    5.    6.
    7.    8.
-->F=[D E]
 F  =
    1.    2.    5.    6.
    3.    4.    7.    8.
```

3 – Matrizes (cont.)

3.2 - Formas de Declaração

Ex.:

```
-->D=[1 2;3 4]
 D  =
      1.     2.
      3.     4.
-->E=[5 6;7 8]
 E  =
      5.     6.
      7.     8.
-->G=[D;E]
  G  =
      1.     2.
      3.     4.
      5.     6.
      7.     8.
```

3 – Matrizes (cont.)

3.2 - Formas de Declaração

- Pode-se declarar uma matriz modificando o formato de outra com a função *matrix()*

Ex.:
```
-->a=[1 2 3;4 5 6]  //definindo matriz 2x3
 a  =
    1.    2.    3.
    4.    5.    6.
-->b=matrix(a,1,6)  //modificando matriz a para 1x6
 b  =
    1. 4. 2. 5. 3. 6.
-->c=matrix(a,3,2)  //modificando matriz a para 3x2
 c  =
    1.    5.
    4.    3.
    2.    6.
```

3 – Matrizes (cont.)

3.3 – Operações com matrizes

- Multiplicação e divisão por escalar
- Soma e subtração (somente para matrizes de mesma dimensão)

3 – Matrizes (cont.)

3.3 – Operações com matrizes

Ex.: Dado $\boldsymbol{M} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ calcule 3M e M/2

```
-->M=[1 2 3;4 5 6;7 8 9]
-->3*M
 ans  =
    3.      6.      9.
    12.     15.     18.
    21.     24.     27.
-->M/2
 ans  =
    0.5     1.      1.5
    2.      2.5     3.
    3.5     4.      4.5
```

3 – Matrizes (cont.)

3.3 – Operações com matrizes

Ex.:Dado $\boldsymbol{M}=\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ e $\boldsymbol{N}=\begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$ calcule M+N

```
-->M=[1 2 3;4 5 6;7 8 9],N=[9 8 7;6 5 4;3 2 1];
-->M+N
 ans  =

    10.    10.    10.
    10.    10.    10.
    10.    10.    10.
```

3 – Matrizes (cont.)

3.3 – Operações com matrizes

- Transposição: M'

Ex.: Dado $\boldsymbol{M}=\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}$ calcule M'

```
-->M=[1 2 3;4 5 6;7 8 9];
-->M'
 ans  =
    1.     4.     7.
    2.     5.     8.
    3.     6.     9.
```

3 – Matrizes (cont.)

3.3 – Operações com matrizes

- Dimensão – A função *size()* retorna o número de linhas e colunas da matriz

```
-->M=[1 2 3;4 5 6;7 8 9];
-->size(M)
 ans  =
     3.    3.
-->//a matriz M possui 3 linhas e 3 colunas
```

3 – Matrizes (cont.)

3.3 – Operações com matrizes

- Multiplicação – Se $A \in \Re^{mxp}$ e $B \in \Re^{pxn}$ define-se como produto das matrizes A e B:

$$C = AB \in \Re^{mxn}$$

3 – Matrizes (cont.)

3.3 – Operações com matrizes

Ex.: Dados $\boldsymbol{A}=\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}$ e $\boldsymbol{B}=\begin{bmatrix}1 & 4\\ 2 & 5\\ 3 & 6\end{bmatrix}$, calcule AB

```
-->A=[1 2 3;4 5 6;7 8 9];B=[1 4;2 5;3 6];
-->size(A),size(B)  //verificando dimensoes
 ans  =
    3.    3.
 ans  =
    3.    2.
-->A*B
 ans  =
    14.    32.
    32.    77.
    50.    122.
```

3 – Matrizes (cont.)

3.4 – Acesso a Elementos

- Pode-se acessar elementos de matrizes através diversos métodos, entre eles:
  - Uso explícito do índice do elemento

  ex.:

```
-->y=[1 2 3;4 5 6]  //gerando matriz 2x3
 y  =
    1.    2.    3.
    4.    5.    6.
-->y(1,2)   //acesso ao elemento da linha 1 coluna 2
 ans  =
    2.
```

3 – Matrizes (cont.)

3.4 – Acesso a Elementos

- Uso do símbolo ' : ' (dois pontos) – intervalo. Se usado sozinho, o símbolo indica todos os elementos.

  ex.:

```
-->y=[1 2 3;4 5 6]  //gerando matriz 2x3
 y  =
    1.     2.     3.
    4.     5.     6.
-->y(2:4)           //acesso do segundo ao quarto elemento
 ans  =
     4.
     2.
     5.
```

3 – Matrizes (cont.)

3.4 – Acesso a Elementos

- Uso do símbolo ' $ ' - último elemento

  ex.:

```
-->y=[1 2 3;4 5 6]  //gerando matriz 2x3
 y  =
    1.    2.    3.
    4.    5.    6.
-->y($)             //acesso ao último elemento
 ans  =
     6.
```

3 – Matrizes (cont.)

3.4 – Acesso a Elementos

- Uso de operações booleanas

  ex.:

```
-->y=[1 2 3;4 5 6];  //gerando matriz 2x3

-->y([%t %t %f;%f %t %f]); //mostra elementos associados
                           // a variavel true
 ans  =
    1.
    2.
    5.
```

3 – Matrizes (cont.)

3.4 – Acesso a Elementos

Exemplos diversos:

- criando matriz M com o primeiro e segundo elemento da segunda coluna da matriz y

```
-->y=[1 2 3;4 5 6]
 y  =
    1.    2.    3.
    4.    5.    6.
-->M=y([1 2],2)
 M  =
    2.
    5.
```

3 – Matrizes (cont.)

3.4 – Acesso a Elementos

Exemplos diversos:

- criando matriz M com todos os elementos da terceira coluna da matriz y

```
-->y=[1 2 3;4 5 6]
 y  =
    1.     2.     3.
    4.     5.     6.
-->M=y(:,3)
 M  =
    3.
    6.
```

3 – Matrizes (cont.)

3.4 – Acesso a Elementos

Exemplos diversos:

- criando matriz M com todos os elementos da primeira linha da matriz y

```
-->y=[1 2 3;4 5 6]
 y  =
    1.     2.     3.
    4.     5.     6.
-->M=y(1,:)
 M  =
    1.     2.     3.
```

3 – Matrizes (cont.)

3.4 – Acesso a Elementos

Exemplos diversos:

- criando matriz M com todos os elementos da terceira, segunda e primeira colunas da matriz y

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

-->M=y(:,[3 2 1])
 M  =
     3.     2.     1.
     6.     5.     4.
```

3 – Matrizes (cont.)

3.4 – Acesso a Elementos

Exemplos diversos:

- alterando o valor do elemento da segunda linha, segunda coluna da matriz y

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

-->y(2,2)=14
 y  =
     1.     2.      3.
     4.     14.     6.
```

3 – Matrizes (cont.)

3.4 – Acesso a Elementos

Exemplos diversos:

- alterando o valor dos elementos y(1,2) e y(2,2) da matriz y

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

-->y([1 2],2)=[-1;-2]

 y  =
     1.   - 1.     3.
     4.   - 2.     6.
```

linha 2, coluna 2

linha 1, coluna 2

3 – Matrizes (cont.)

3.5 – Matrizes com polinômios

- Os elementos de uma matriz também podem ser polinômios:

Ex.:

```
-->s=poly(0, 's'); p=1-2*s+s^2;  //definindo polinomio

-->M=[p, p-1; p+1, 3]             //definindo matriz
```

p   p - 1

```
 M  =

                 2             2
    1 - 2s + s      - 2s + s

                 2
    2 - 2s + S         3
```

p + 1   3

3 – Matrizes (cont.)

3.5 – Matrizes com polinômios

- Podem ser usadas as funções para polinômios:

```
 -->M
  M  =
                     2             2
     1 - 2s + s        - 2s + s
                     2
     2 - 2s + s          3
  -->horner(M,2)           //avalia M em s=2
  ans  =
     1.     0.
     2.     3.
```

3 – Matrizes (cont.)

3.6 – Matrizes racionais

- A partir de uma matriz M podemos criar uma matriz apenas com os numeradores :

```
-->M=[1/s, (s+2)/(s-1);2, 3]
 M  =
    1      2 + s
    -      -----
    s    - 1 + s

    2      3
    -      -
    1      1
-->N=M('num')    //seleciona apenas os numeradores
 N  =
    1      2 + s
    2      3
```

3 – Matrizes (cont.)

3.6 – Matrizes racionais

- A partir de uma matriz M podemos criar uma matriz apenas com os denominadores :

```
-->M=[1/s, (s+2)/(s-1);2, 3]
 M  =
    1      2 + s
    -      -----
    s    - 1 + s

    2      3
    -      -
    1      1
-->N=M('den')    //seleciona apenas os denominadores
 N  =
   s    - 1 + s
   1      1
```

3 – Matrizes (cont.)

3.7 – Matrizes simbólicas

- Uma matriz simbólica pode ser constituída de elementos to tipo *string*

```
 -->M=['a' 'b';'c' 'd']
  M  =
 !a  b !
 !     !
 !c  d !
```

- Se atribuirmos valores às variáveis podemos visualizar a forma numérica da matriz com a função *evstr()*

```
-->a=2;b=4;c=1;d=6;
-->evstr(M)
 ans  =
    2.    4.
    1.    6.
```

3 – Matrizes (cont.)

3.8 – Operadores especiais

- operador \ (divisão à esquerda)

Seja **Ax=b** um sistema de equações lineares escrito na forma matricial sendo **A** a matriz dos coeficientes, **x** o vetor das incógnitas e **b** o vetor dos termos independentes:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}_{nxn} \qquad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ . \\ . \\ . \\ b_n \end{bmatrix}$$

3 – Matrizes (cont.)

3.8 – Operadores especiais

A resolução deste sistema é $x=A^{-1}b$, ou seja, basta obter a matriz inversa de **A** e multiplicá-la pelo vetor **b**. No Scilab isto pode ser feito desta forma:

```
-->A=[2 0;0 4];b=[1;8];
-->inv(A)               //checando se A admite inversa
 ans  =
    0.5    0.
    0.     0.25
-->x=inv(A)*b           //solucao do sistema linear
 x  =
    0.5
    2.
```

3 – Matrizes (cont.)

3.8 – Operadores especiais

Esta solução pode ser obtida com o operador “divisão à esquerda” cujo símbolo é \

```
-->A=[2 0;0 4];b=[1;8];
-->inv(A)              //checando se A admite inversa
 ans  =
    0.5    0.
    0.     0.25
-->x=A\b               //solucao do sistema linear
 x  =
    0.5
    2.
```

3 – Matrizes (cont.)

3.8 – Operadores especiais

• operador . (ponto)

Este operador é usado com outros operadores para realizar operações elemento a elemento.

```
-->u=[1;2;3];v=[2;4;6];
-->u.*v
 ans  =
    2.
    8.
    18.
-->u./v
 ans  =
    0.5
    0.5
    0.5
```

3 – Matrizes (cont.)

3.8 – Operadores especiais

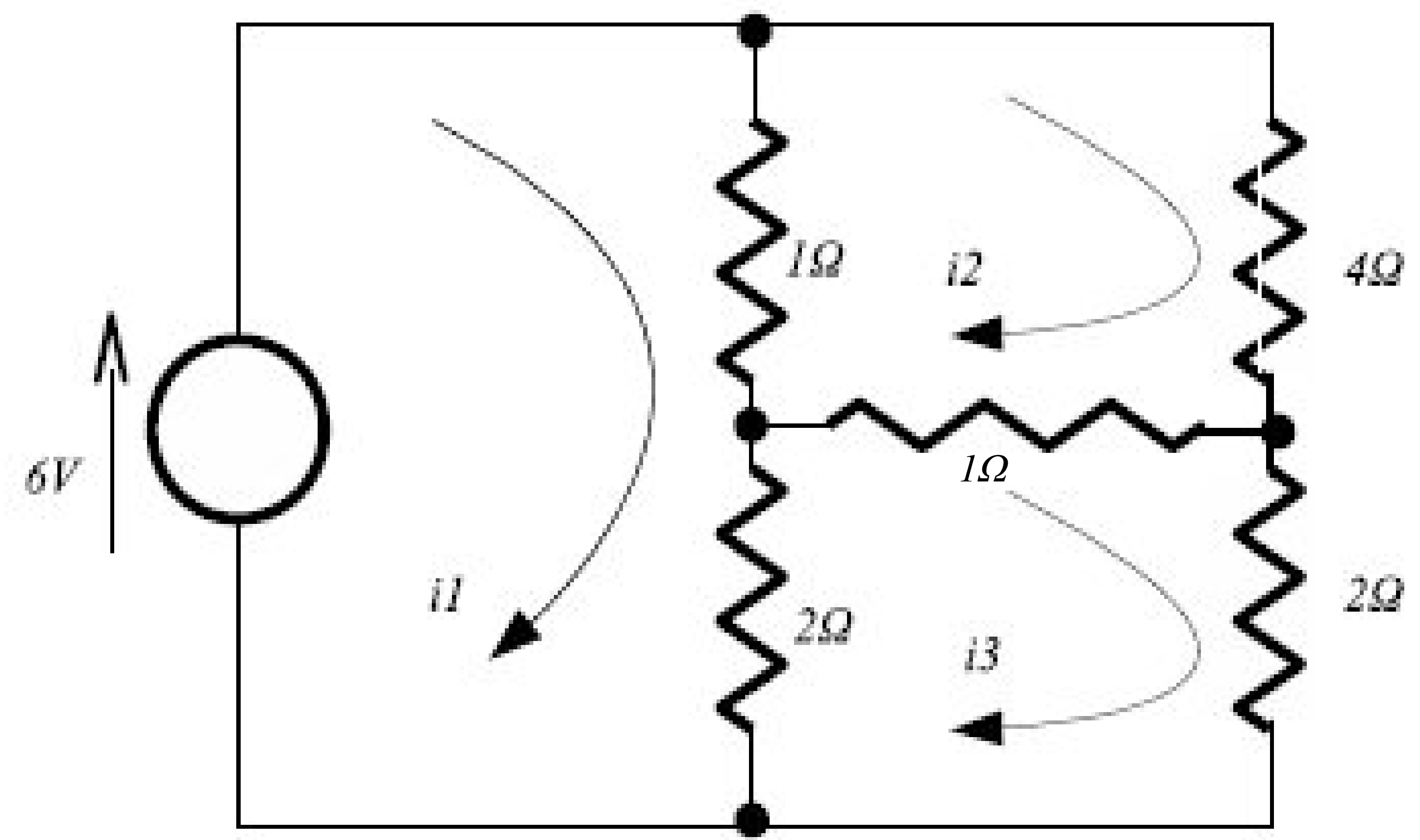
- operador . (ponto)

```
-->u=[1;2;3];v=[2;4;6];
-->u.^v
 ans  =
    1.
    16.
    729.

-->v.^u
 ans  =
    2.
    16.
    216.
```

4- Exercícios

4.1 – Exercícios

Dado o circuito abaixo, calcule as correntes de laço.



4- Exercícios (cont.)

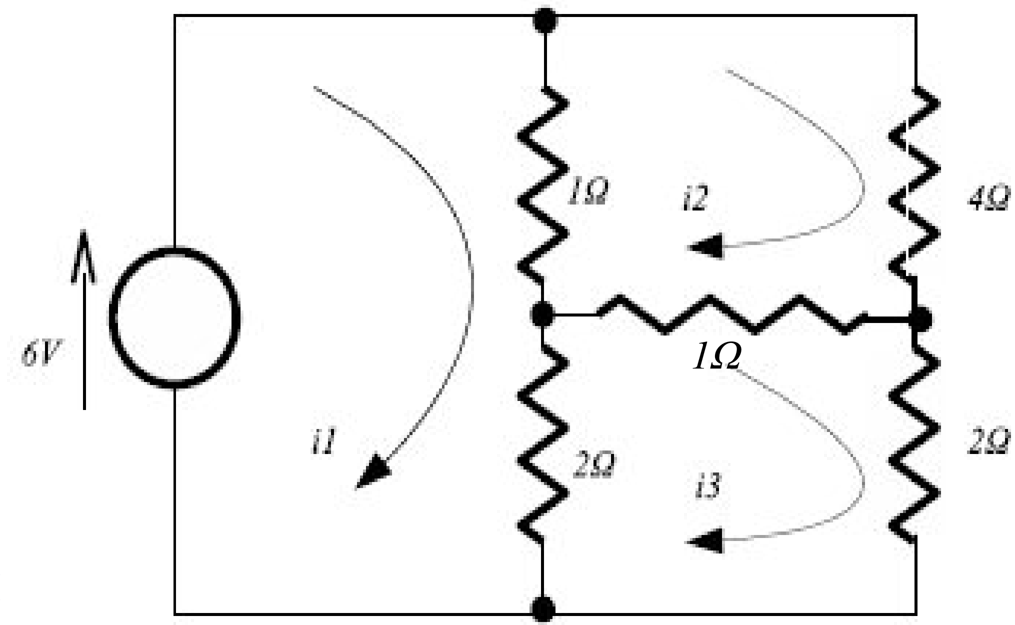
4.1 – Exercícios

As equações de laços são:

$3i_1 - 1i_2 - 2i_3 = 6$

$-1i_1 + 6i_2 - 1i_3 = 0$

$-2i_1 - 1i_2 + 5i_3 = 0$



Na forma matricial temos:

$$\begin{bmatrix} 3 & -1 & -2 \\ -1 & 6 & -1 \\ -2 & -1 & 5 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix}$$

4- Exercícios (cont.)

4.1 – Exercícios

Podemos resolver o sistema no Scilab usando o operador \

```
-->//definindo matriz A
-->A=[-3 1 2;1 -6 1;2 1 -5]
A =
! - 3. 1. 2. !
! 1. - 6. 1. !
! 2. 1. - 5. !

-->//definindo matriz b
-->b=[6;0;0]
b =
! 6. !
! 0. !
! 0. !
```

4.1 – Exercícios

```
-->x=A\b
 x  =

    3.2222222
    0.7777778
    1.4444444
```

As correntes são aproximadamente:

$i_1 = 3{,}22$ A

$i_2 = 0{,}78$ A

$i_3 = 1{,}44$ A

Referências

- Pires, P.S.M. (2004). *Introdução ao Scilab*, Rio Grande do Norte.
- Noble, B. e Daniel, J.W. (1986). *Álgebra Linear Aplicada*, Prentice Hall do Brasil, Rio de Janeiro.