

UNIVERSIDADE DO VALE DO ITAJAÍ – UNIVALI – CES VII
CURSO: ENGENHARIA DE COMPUTAÇÃO
DISCIPLINA: INTRODUÇÃO À ENGENHARIA DE COMPUTAÇÃO

SciLab

INSTRUÇÕES BÁSICAS

Prof. Raimundo C. Ghizoni Teive

Versão do Tutorial: 1.0.0
Outubro – 2003

Tabela de Conteúdos

Conceitos Básicos.....	3
Símbolos e Constantes.....	4
Vetores & Matrizes.....	5
String Array.....	9
Gráficos.....	10
Polinômios.....	11

SciLab é um pacote de software livre que provê um poderoso ambiente de computação numérica para aplicações científicas e de engenharia. Sendo software livre, o usuário tem liberdade de copiar, distribuir e instalar em qualquer computador de sua propriedade ou de terceiros, além de ter total acesso ao código fonte, podendo modificá-lo e redistribuí-lo alterado.

O SciLab está disponível para download, podendo ser executado em diversas plataformas de hardware e sistema operacional, no endereço <http://scilabsoft.inria.fr/>. Até a data de edição deste documento, o SciLab estava na versão 2.7.2.

Este tutorial foi produzido com OpenOffice (<http://www.openoffice.org/>) sendo executado no Sistema Operacional GNU/Linux (<http://www.gnu.org>, <http://www.linux.org/>), ambos software livre, buscando atingir a independência de sistemas proprietários para a realização das tarefas acadêmicas, científicas e de engenharia.

Será bem-vinda qualquer sugestão ou crítica que venha melhorar a qualidade deste tutorial. Portanto, se você tem alguma, mande e-mail para teive@sj.univali.br ou probst@sj.univali.br.

Atenciosamente,

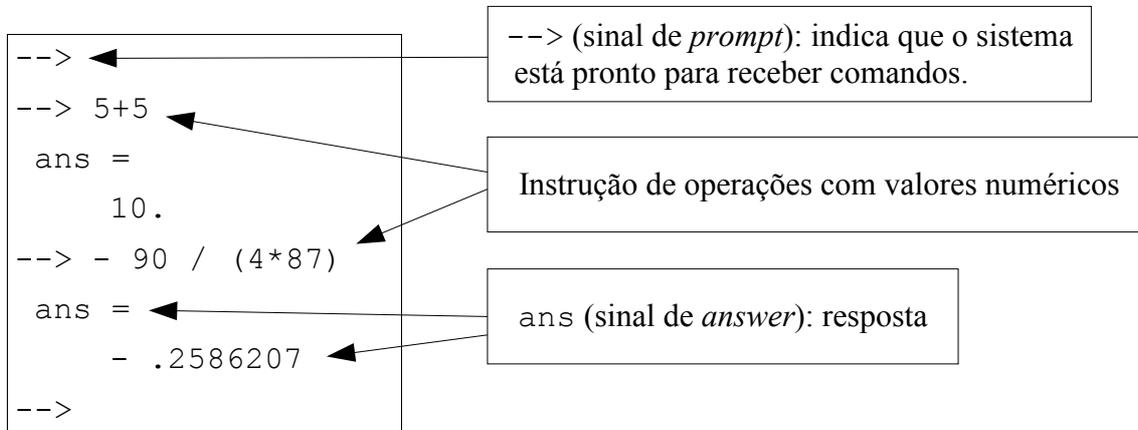
Prof. Raimundo C. Ghizoni Teive

Edição: AFP - probst@sj.univali.br

SciLab – Conceitos Básicos

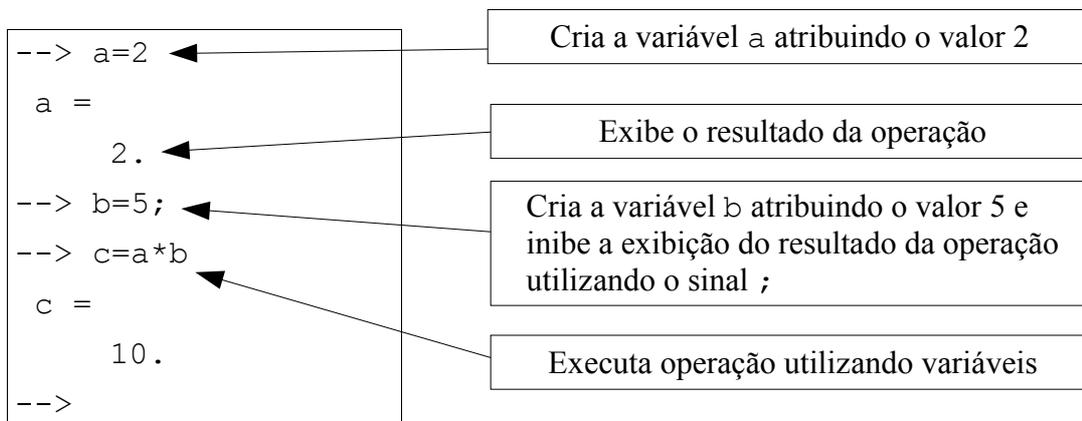
- Operações na linha de comando

A melhor maneira de começar a entender o SciLab é digitando instruções nas linhas de comando e observar os resultados.



OBS.: Para o SciLab letras minúsculas e maiúsculas são diferentes.

- Variáveis: atribuição de valores



- Variáveis: comandos `who` e `whos ()`

- `who`: exibe os nomes de todas as variáveis do *workspace*.

- `whos ()`: exibe os nomes, os tamanhos e os tipos de dados de todas as variáveis do *workspace*.

- Comandos Utilitários

- `pwd` ou `getcwd()`: exibe o diretório corrente
(ainda pode-se usar o menu <File>, opção <Get current Directory ...>)
--> `pwd`
ans =
/home/usuario
- `chdir <diretório>` : muda o diretório corrente
(ainda pode-se usar o menu <File>, opção <Change Directory ...>)
--> `chdir /tmp`
- `..ls` ou `..dir` : exibe os nomes dos arquivos do diretório corrente
- `unix "<comando>"` ou `..<comando>` : executa um comando do sistema operacional
--> `unix "rm arquivo.m"`
ou
--> `..rm arquivo.m`

SciLab – Símbolos e Constantes

- `ans`: resposta mais recente.
- `%eps`: precisão numérica corrente. Ex. 2.220E-16
- `%pi`: 3.1415927
- `%i`: parte imaginária de números complexos.
- `%inf`: infinito. Ex: 1/0.
- `%nan`: "not a number"

SciLab – Vetores & Matrizes

- Sintaxe básica: escalares e vetores

```
--> a=10
a =
    10.
--> b=[1 2 3]
b =
!   1.  2.  3.  !
--> c= [ 5; 6; 7]
c =
!   5.  !
!   6.  !
!   7.  !
-->
```

Escalar é uma matrix 1 x 1

Vetor linha é uma matrix 1 x n

Vetor colula é uma matrix n x 1
- espaço ou , separam elementos na mesma linha
- ; separa elementos da mesma coluna

OBS.: Matrizes podem ser numéricas ou alfa-numéricas.

- Sintaxe básica: matrizes

```
--> x=[1 5 9; 3 7 2; 6 4 8]
x =
!   1.   5.   9.  !
!   3.   7.   2.  !
!   6.   4.   8.  !
--> x(1,3)
ans =
    9.
--> x(8)
ans =
    2.
--> x(2:4)
ans =
!   3.  !
!   6.  !
!   5.  !
```

Matriz

Indexação de matrizes: segue a sintaxe matemática tradicional.

x(1,3) indica elemento na primeira linha e terceira coluna.

x(8) indica o oitavo elemento da matriz. A contagem é feita por coluna.

x(2:4) indica intervalo do segundo ao quarto elemento da matriz.

- Matrizes Numéricas

Qualquer tipo de número ou operação numérica pode ser elemento de uma matriz numérica.

```
--> a=[-2.8,sqrt(7),(5+4)/8,1; sin(0.4),exp(0.75),fix(4.5),100]
a =
! -2.8          2.6457513    1.125    1.    !
!  .3894183    2.117          4.        100.  !
```

Acrescenta o valor na matriz e preenche os outros índices com 0.

```
--> a(1,7)=555
a =
! -2.8          2.6457513    1.125    1.    0.    0.    555.  !
!  .3894183    2.117          4.        100.  0.    0.    0.    !
```

- Vetores “monotônicos”

(:) cria vetores com elementos linearmente espaçados.

```
--> x=1:10
x =
! 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  !
```

Vetor 1 a 10

```
--> x=0:0.5:2
x =
! 0.  .5  1.  1.5  2.  !
```

Cria vetor com:
- primeiro elemento 0
- último elemento 2
- incremento 0.5

```
--> y=1:-0.2:0
y =
! 1.  .8  .6  .4  .2  0.  !
```

Incremento pode ser negativo

- Operações Matriciais

Matriz:
 + (soma)
 - (subtração)
 * (multiplicação)
 / (divisão)
 \ (divisão a esquerda)
 ^ (potência)
 ' (transposta)

Elemento a elemento:
 .* (multiplicação escalar)
 ./ (divisão escalar)
 .^ (potência escalar)

```
--> a=[1 2 3; 4 5 6; 7 8 9]
a =
! 1.  2.  3.  !
! 4.  5.  6.  !
! 7.  8.  9.  !
--> b1=a*a
b1 =
! 30.  36.  42.  !
! 66.  81.  96.  !
! 102. 126. 150.  !
--> b2=a.*a
b2 =
! 1.  4.  9.  !
! 16. 25. 36.  !
! 49. 64. 81.  !
```

- Exemplo: Operações Matriciais

Operação de divisão a esquerda: **Sistema Linear.**

$$\begin{aligned} -x_1 + x_2 + 2x_3 &= 2 \\ 3x_1 - x_2 + x_3 &= 6 \\ -x_1 + 3x_2 + 4x_3 &= 4 \end{aligned}$$

$$\begin{aligned} x_1 &= 1 \\ x_2 &= -1 \\ x_3 &= 2 \end{aligned}$$

```
--> a=[-1 1 2; 3 -1 1; -1 3 4];
--> b=[2;6;4];
--> x=a\b
x =
! 1.  !
! -1.  !
! 2.  !
```

- Matriz Randômica: função rand()

rand(): gera números randômicos uniformemente distribuídos entre [0, 1].

```
-->rand(3,3)
ans =
! .2113249 .3303271 .8497452 !
! .7560439 .6653811 .6857310 !
! .0002211 .6283918 .8782165 !
```

rand(n,n): gera matriz randômica n x n

```
-->rand(2,3)
ans =
! .5608486 .7263507 .5442573 !
! .6623569 .1985144 .2320748 !
```

rand(n,m,p,...): gera matriz n x m x p ...

OBS.: rand("seed", num) altera a base para geração de números.

- Concatenação de Matrizes

Utilizamos os símbolos [] , ; para concatenar matrizes.

```
-->a=[1 2 3 4 5];
-->b=[6 7 8 9 0];
-->c=[a, b] ← Concatenação por linha

c =
! 1. 2. 3. 4. 5. 6. 7. 8. 9. 0. !

-->d=[a;b] ← Concatenação por coluna

d =
! 1. 2. 3. 4. 5. !
! 6. 7. 8. 9. 0. !
```

- Funções Matriciais Básicas I

- `det(m)` : calcula determinante da matriz quadrada m.
- `inv(m)` : gera matriz inversa da matriz quadrada m.
- `zeros(n, m, p...)` : gera matriz de zeros n x m x p...
--> `zeros(3,4)` ;
- `ones(n, m, p...)` : gera matriz de uns n x m x p...
--> `ones(3,4)` ;
- `eye(n,m)`, `eye(m,m)` : gera matriz identidade n x m ou m x m.
--> `eye(3,4)` ;
--> `eye(3,3)` ;
- `diag(m)` : se m é uma matriz, retorna os elementos da diagonal principal
--> `diag(m)` ;
- `diag(v)` : gera matriz diagonal utilizando os elementos do vetor v.
--> `v=[1 2 3 4 5]` ;
--> `diag(v)`
ans =
! 1. 0. 0. 0. 0. !
! 0. 2. 0. 0. 0. !
! 0. 0. 3. 0. 0. !
! 0. 0. 0. 4. 0. !
! 0. 0. 0. 0. 5. !

- Funções Matriciais Básicas II

- `matrix(x, n, m)`: re-formata a matriz x com dimensões n x m.

```
-->x=[1 2 3 4 5 6 7 8 9 0];  
-->matrix(x, 2, 5)  
ans =  
!   1.   3.   5.   7.   9. !  
!   2.   4.   6.   8.   0. !
```

- `tril(m)`: extrai matriz triangular inferior.

```
-->m=rand(3, 3)  
m =  
!   .5738457   .4734382   .3586156 !  
!   .6160412   .0312314   .7726796 !  
!   .6048677   .0701504   .7317216 !  
  
-->tril(m)  
ans =  
!   .5738457   0.   0.   !  
!   .6160412   .0312314  0.   !  
!   .6048677   .0701504  .7317216
```

- `triu(m)`: extrai matriz triangular superior.

```
-->triu(m)  
ans =  
!   .5738457   .4734382   .3586156 !  
!   0.   .0312314   .7726796 !  
!   0.   0.   .7317216 !
```

SciLab – String Array

- Utilizamos o símbolo ' para criar string arrays.

```
-->str1='Oi !'  
str1 =  
Oi !  
  
-->str2='Tudo bem ?'  
str2 =  
Tudo bem ?  
  
-->str3=str1 + ' ' + str2  
str3 =  
Oi ! Tudo bem ?
```

String arrays

Concatenação

SciLab – Gráficos

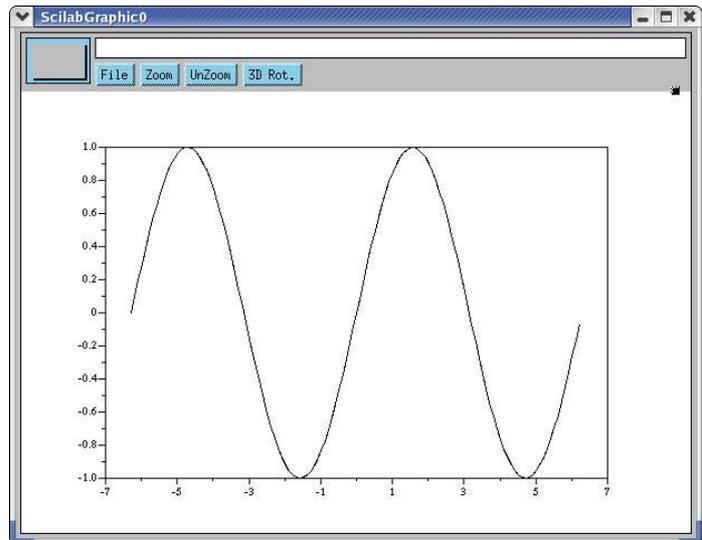
– Gráficos 2D

`plot2d(xdata, ydata):`

- `xdata`: vetor contendo dados em X;

- `ydata`: vetor contendo dados em Y;

```
-->x=[-2*%pi:0.1:2*%pi]';  
-->y=sin(x);  
-->plot2d(x,y)
```



– Exemplos

Teste a função `plot2d` utilizando outras funções e composições de funções.

- `cos(x)`: cosseno.
- `exp(x)`: exponencial (e^x).
- `log(x)`: logaritmo natural (\ln).
- `sqrt(x)`: raiz quadrada.
- `sin(x) .* exp(x)`
- `abs(sqrt(x))`
- `3*sin(x) - 0.5*cos(5*x)`

```
-->x=[-2*%pi:0.1:2*%pi]';  
-->y=3*sin(x) - 0.5*cos(5*x);  
-->k=sin(x) .* exp(x);  
-->plot2d(x, [y, k])
```

Sintaxe para exibição de várias curvas

OBSERVAÇÕES:

1. `xdata` deve ser um vetor coluna, por isso aplica-se a transposta (');
2. A função `plot2d` possui várias sintaxes diferentes. Consulte o *Help* do SciLab.

SciLab – Polinômios

- Função `roots`: obtém raízes de polinômios.

```
-->s=poly(0,"s");
-->p=2*s + 3
p =
  3 + 2s
-->roots(p)
ans =
  - 1.5
-->p=6*s^5 - 15*s^4 + 3*x^3 + 2*s^2 + 3*x + 5
p =
      5      4      3      2      1      0
      5 + 3s + 2s + 3s - 15s + 6s
-->roots(p)
ans =
! - .0845548 + .7107609i !
! - .0845548 - .7107609i !
! - .6380605             !
!  1.2233204             !
!  2.0838496             !
```

Seta s como a variável do polinômio

$p(x) = 2x + 3$

$p(-1.5) = 0$

$p(x) = 6x^5 - 15x^4 + 3x^3 + 2x^2 + 3x + 5$

Este polinômio possui três raízes reais e duas raízes imaginárias.