Curso de Scilab

Luís Cláudio OLIVEIRA-LOPES

Parte II

06 de Outubro de 2007

Uberlândia, Minas Gerais, Brasil

SUMÁRIO

Mascote (*puffin*)

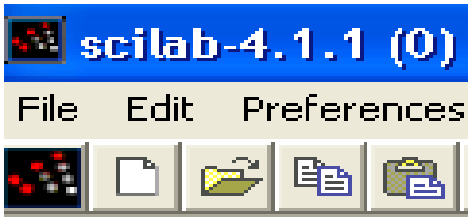
- Objetos Scilab
  - Real/Complexo/*String*:Matriz
  - Polinômio
  - Listas
  - Células
- I/O no Scilab
  - De teclado/menu para tela
  - De/para arquivo
  - Gráficos
- Funções no Scilab
  - Ambiente script e scipad()
- Exercícios

Objetos no Scilab...

Objetos...

Scilab: Sentenças e Variáveis

- Atribuição:
  - Nome var qq tamanho, considera até 24.
  - Ex.: var = 2 (ecoando na tela)
  - Ex.: var=2; (sem ecoar na tela)
- Comparação
  - Ex.: 2 > 1; b <= 4; a ~= 3; a == b
- Estruturas primitivas:
  - Variável numérica real:
    - Ex.: 2 , 3.45; 4.2e-04
  - Variável complexa ; %i= sqrt(-1):
    - Ex.: 2+3*%i
  - Variável lógica
    - Ex.: %F, %f, %T, %t
  - Variável cadeia de caracter (*string*):
    - Ex.: a='teste'




```
-->var=2
 var  =

    2.

-->var=2;

-->2>1
 ans  =

  T

-->a=2+3*%i
 a  =

    2. + 3.i

-->a='teste'
 a  =

 teste

-->
```

Scilab: Sentenças e Variáveis

- Operador de Intervalo
  - [início:incremento:final]
    - Ex.: a=[1:2:10]
- Estruturas não primitivas:
- Vetores e Matrizes:

A=[1.0, 3.45; 4.2e-04, 2+3*%i]

  - Uso de Matrizes/vetores
    - Acesso: A(i,j)
    - Exclusão: A(:,j)=[]
    - **Indicadores especiais:**
      - i:j representa intervalo
      - A(:,i) todas as linhas e coluna j
      - $ representa última posição
    - Operações elemento a elemento
      - Ex.: A*B (regular) e A.*B (elemento)


```
-->a=[1:2:10]
 a   =

    1.    3.    5.    7.    9.

-->A=[1.0, 3.45;  4.2e-04, 2+3*%i]
 A   =

    1.          3.45
    0.00042     2. + 3.i
```

Sensível ao caso: a <> A

Operacão elemento a elemento: uso de . (ponto)

Estrutura não primitiva: Matrizes/vetores


```
-->B=rand(4,4)
 B   =

    0.2113249    0.6653811    0.8782165    0.7263507
    0.7560439    0.6283918    0.0683740    0.1985144
    0.0002211    0.8497452    0.5608486    0.5442573
    0.3303271    0.6857310    0.6623569    0.2320748

-->C=B(2:3,2:4)  (faixa)
 C   =

    0.6283918    0.0683740    0.1985144
    0.8497452    0.5608486    0.5442573

-->D=B($,:)      (indicadores especiais=  :  $)
 D   =

    0.3303271    0.6857310    0.6623569    0.2320748

-->B(:,2)=[]     (remoção de linha/coluna)
 B   =

    0.2113249    0.8782165    0.7263507
    0.7560439    0.0683740    0.1985144
    0.0002211    0.5608486    0.5442573
    0.3303271    0.6623569    0.2320748
```

: indica faixa
todas as linhas/colunas

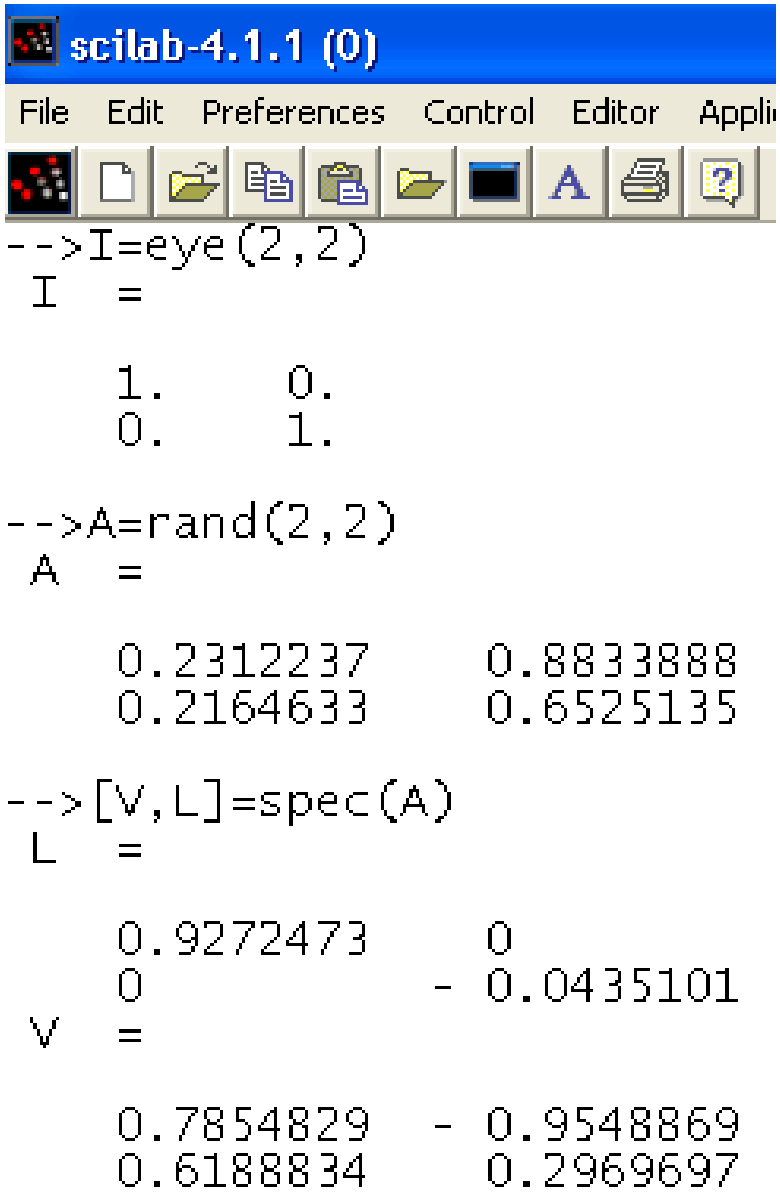
$ indica
última posição

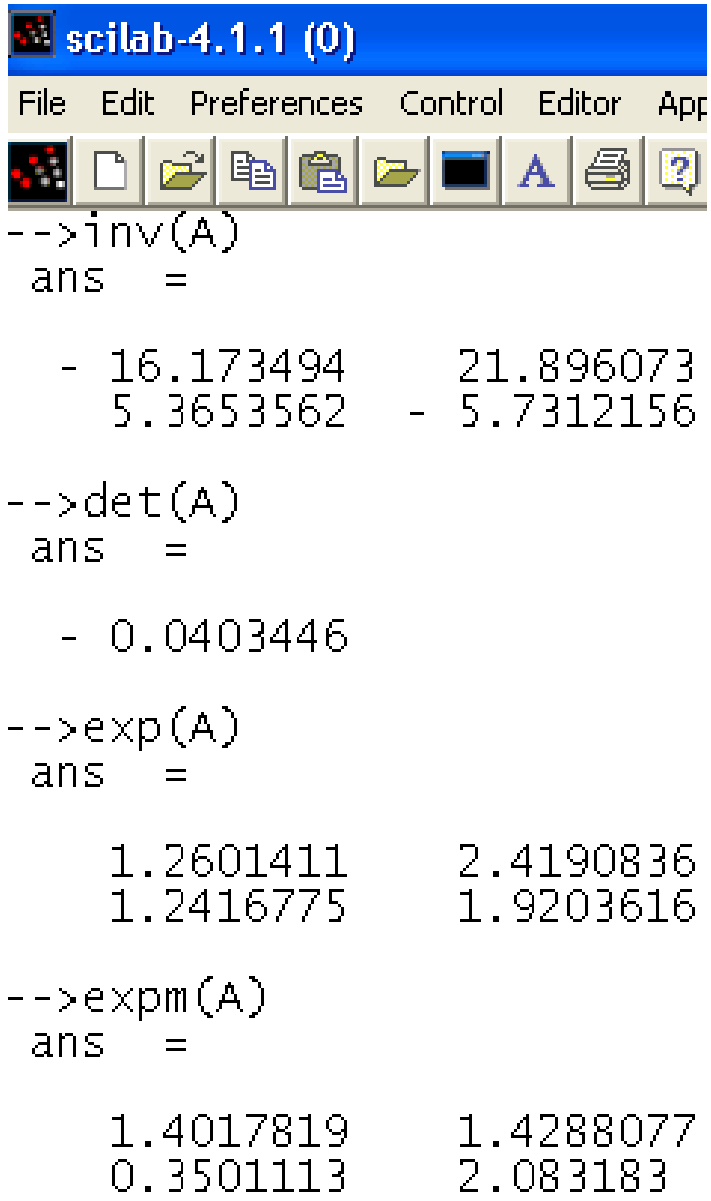
Operadores para Matrizes/vetores

Operador	Função	
[]	definição de matriz e concatenação	
;	separador de linhas	
()	extração de elemento	$m = a(k)$
()	inserção de elemento	$a(k) = m$
′	transposta	
+	adição	
−	subtração	
*	multiplicação	
\	divisão a esquerda	
/	divisão a direita	
^	potenciação	
.*	multiplicação elemento a elemento	
.\	divisão à esquerda elemento a elemento	
./	divisão à direita elemento a elemento	
.^	potência elemento a elemento	
. * .	produto de kronecker	
./.	divisão de kronecker a direita	
.\.	divisão de kronecker a esquerda	

Algumas funções para matrizes:

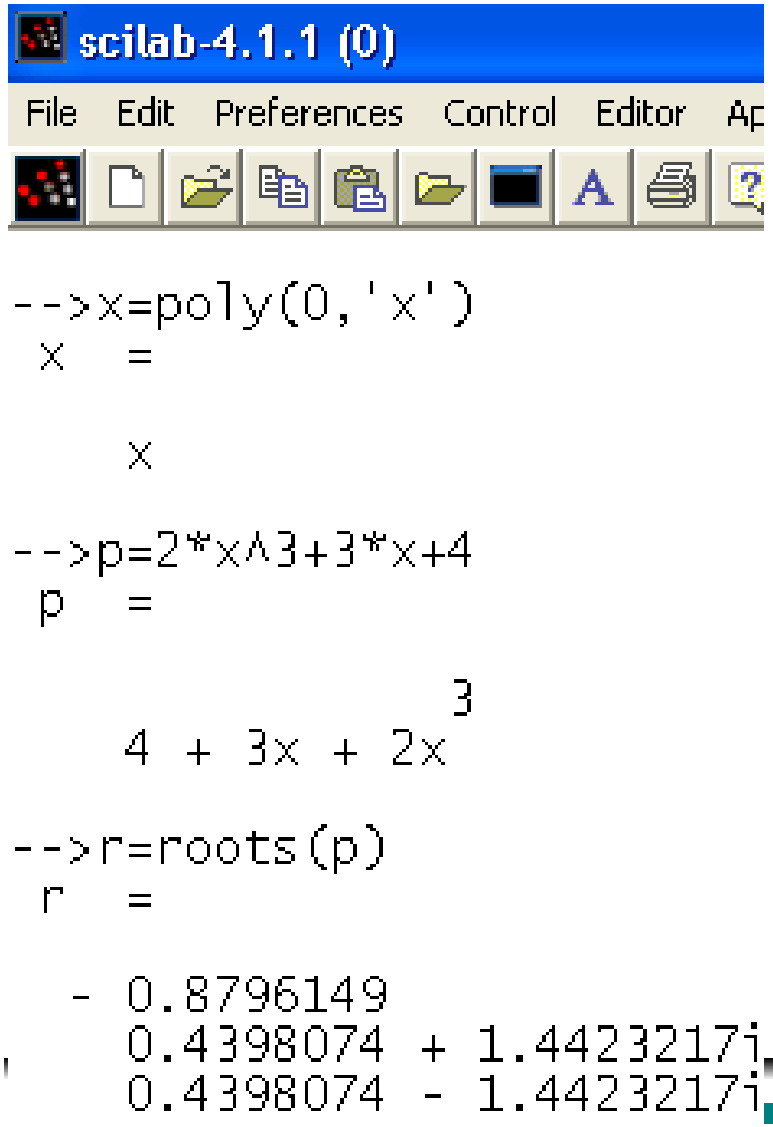
- eye()
- rand()
- [V,L]=spec()
- spec()
- inv()
- det()
- rank()
- exp()
- expm()
- zeros()
- ones()

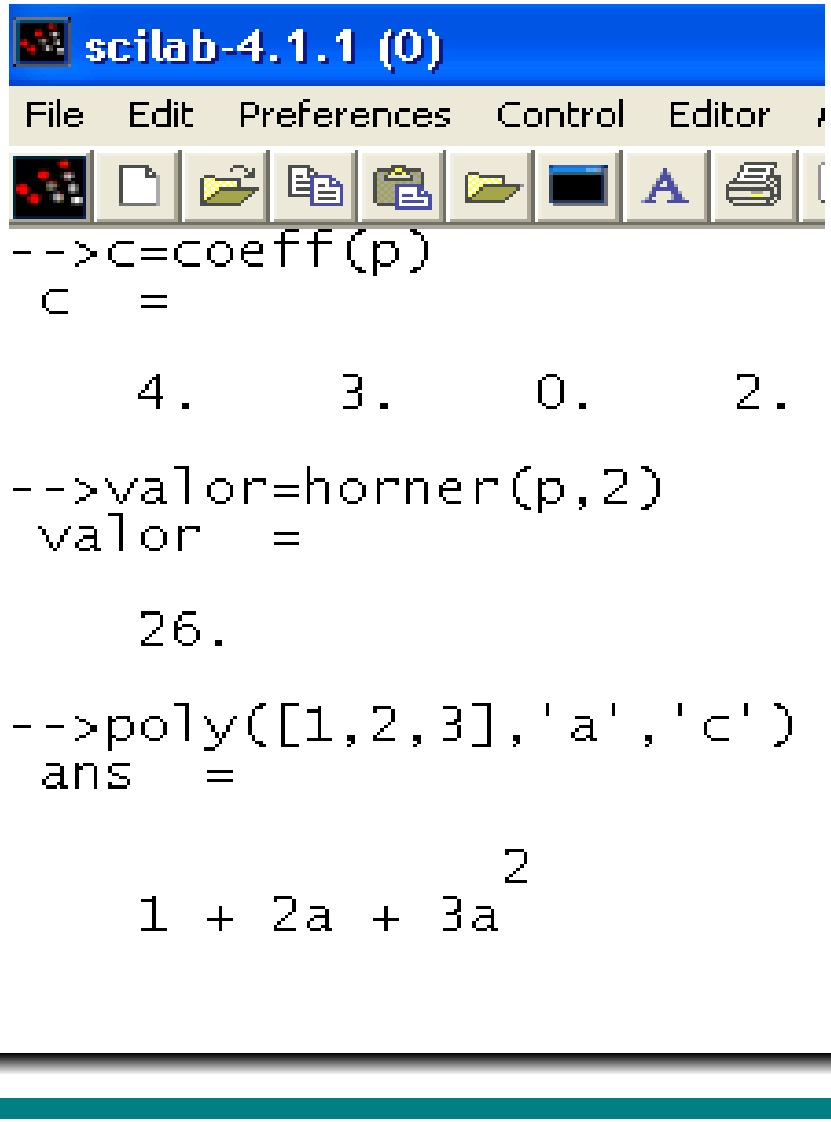




Estrutura não primitiva: polinômio

- [p]=poly(a,"x", ["flag"])
  - “flag" : “roots" ou “coeff", valor default é “roots”.
- Polinômio p(x) em que a é o vetor de “flag”





Estrutura não primitiva: list e tlist

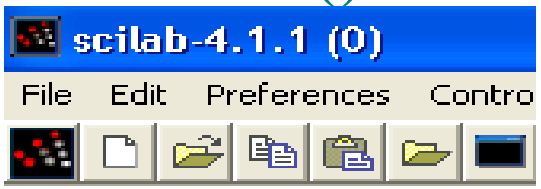
- [L]=list(a1,a2,...an)
- [T]=tlist(tipo,a1,....an) (lista com tipo)
- [M]=mlist(tipo,a1,...an) (lista com tipo orientada para matriz)

```
tipo=['nome do tipo'; 'nome_a1';'nome_a2',...'nome_an']
```

- extração : [x,y,z...]=L(v)
- Inserção no índice i : L(i)=a
- Adiciona elemento no final : L($+1)=b
- Adiciona elemento no início : L(0)=b
- Apagar elemento : L(i)=null()
- Concatenação listas : L3 = lstcat(L1,L2)
- Número de elementos: size(L) ou length(L)

Estrutura não primitiva: cell

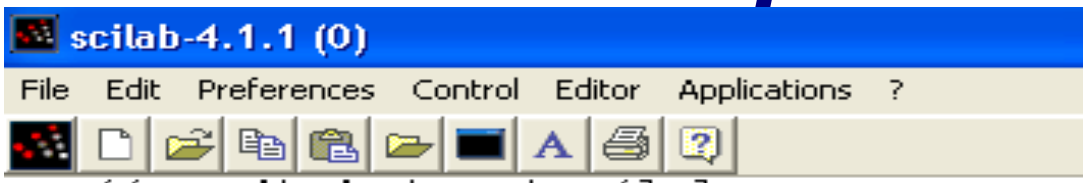
- c=cell(m1,m2,...)
- c=cell(x)
- c=cell()




```
-->b=cell(1,4)
 b   =

!{}   {}   {}   {}   !

-->c=cell([2,3,2])
 c   =

(:,:,1)

!{}   {}   {}   !
!                !
!{}   {}   {}   !
(:,:,2)

!{}   {}   {}   !
!                !
!{}   {}   {}   !
```


```
-->// Atribuindo entrada

-->b(1).entries=1:4
 b   =


![1,2,3,4]   {}   {}   !

-->b(2).entries='Scilab'
 b   =


![1,2,3,4]   "Scilab"   {}   !

-->b(3).entries=poly(1:2,'x','r')
 b   =


![1,2,3,4]   "Scilab"   2-3*x+x^2   !
```

Estrutura não primitiva: cell




```
-->// Atribuindo sub-células

-->X=cell(3,2);

-->X(:,1)=b
 X  =


![1,2,3,4]   {}   !
!             !
!"Scilab"    {}   !
!             !
!2-3*x+x^2   {}   !

-->// Referenciando sub-células

-->b(1) // resultado é célula
 ans  =


 [1,2,3,4]

-->b(1:2)
 ans  =


![1,2,3,4]   !
!            !
!"Scilab"    !

-->b(1).entries // resultado é vetor
 ans  =

    1.    2.    3.    4.
```

Célula

Vetor

Algumas Funções de Uso Geral

- clc - limpa tela de comandos
- clear [vars] - limpa variáveis [vars] não protegidas
- who - mostra lista de variáveis correntes
- whos - who com formato detalhado
- browsevar() - mostra janela com objetos correntes
- help - tela de ajuda do Scilab
- pwd - mostra o nome do diretórios corrente
- chdir - muda diretório
- diary('arquivo').....diary(0) - liga e desliga diário
- what - lista primitivos do Scilab
- type - mostra o tipo da variável
- save....load – salvar e ler (formato binários)
  - comando(arquivo [,x1,x2,...,xn]) ou comando(arquivo)
  - comando(unid [,x1,x2,...,xn]) ou comando(unid)

Diferenças Scilab x Matlab!!!!

scilab-4.1.1 (0)

File Edit Preferences Control Editor Applica

```
-->a=rand(2,2)
 a   =

    0.5465335    0.7395657
    0.9885408    0.0037173

-->a+[]
 ans   =

    0.5465335    0.7395657
    0.9885408    0.0037173

-->a-[]
 ans   =

    0.5465335    0.7395657
    0.9885408    0.0037173

-->[]+a
 ans   =

    0.5465335    0.7395657
    0.9885408    0.0037173

-->[]-a
 ans   =

   - 0.5465335   - 0.7395657
   - 0.9885408   - 0.0037173
```

MATLAB

File Edit View Web Window Help

```
>> a=rand(2,2)

a =

    0.9501    0.6068
    0.2311    0.4860

>> a+[]
??? Error using ==> +
Matrix dimensions must agree.

>> b=2;
>> b+[]

ans =

     []

>> b-[]

ans =

     []

>> b-[]

ans =

     []
```

CUIDADO!
Existem várias diferenças

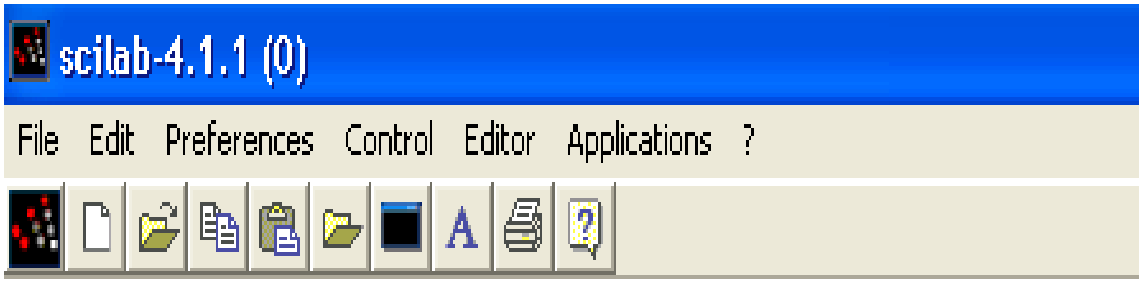
Entrada-Saída (I/O) no Scilab...

Entrada-Saída,....

Funções para entrada de dados: teclado/mouse

Função	Sintaxe
input	[x]=input(mensagem,["string"])
scanf(formato);	ler dados formatados via teclado
read(%io(1),...)	leitura de dados formatados via teclado
x_dialog	r=x_dialog(labels,valor_inicial)
getvalue	[ok,x1,..,x14]=getvalue(Título,labels,tipos,inicial)
x_mdialog	r=x_mdialog(titulo,labels,valor_default_inicial)
x_message	[n]=x_message(strings [,botões])
x_choose	[n]=x_choose(items,titulo [,botão])
x_choices	r=x_choices(titulo,items)
x_matrix	[r]=x_matrix(label,matriz-inicial)

Entrada de Dados no Scilab: input




```
-->n=input("Forneça o número de interações: ")
Forneça o número de interações: -->4
 n  =

    4.

-->var=input("Onde estuda?","string")
Onde estuda?-->Universidade Federal de Uberlândia
 var  =

 Universidade Federal de Uberlândia
```

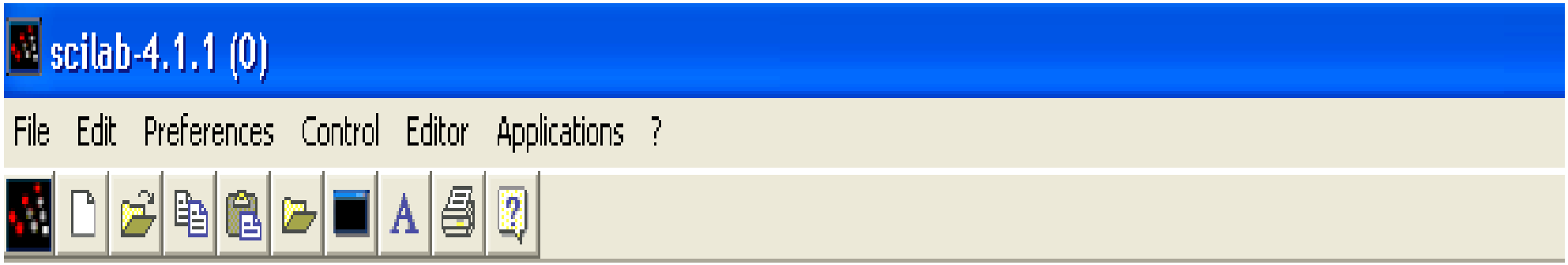

```
-->A=input('Forneça a Matriz A:')
Forneça a Matriz A:-->[1,2;3,4]
 A  =

    1.    2.
    3.    4.

-->nome=input('Seu nome é:')
Seu nome é:-->'Scilábio da Silva'
 nome  =

 Scilábio da Silva
```

Entrada de Dados: getvalue

[ok,x1,..,x14]=getvalue(Tit_janela,Tit_var,tipo,inicial)

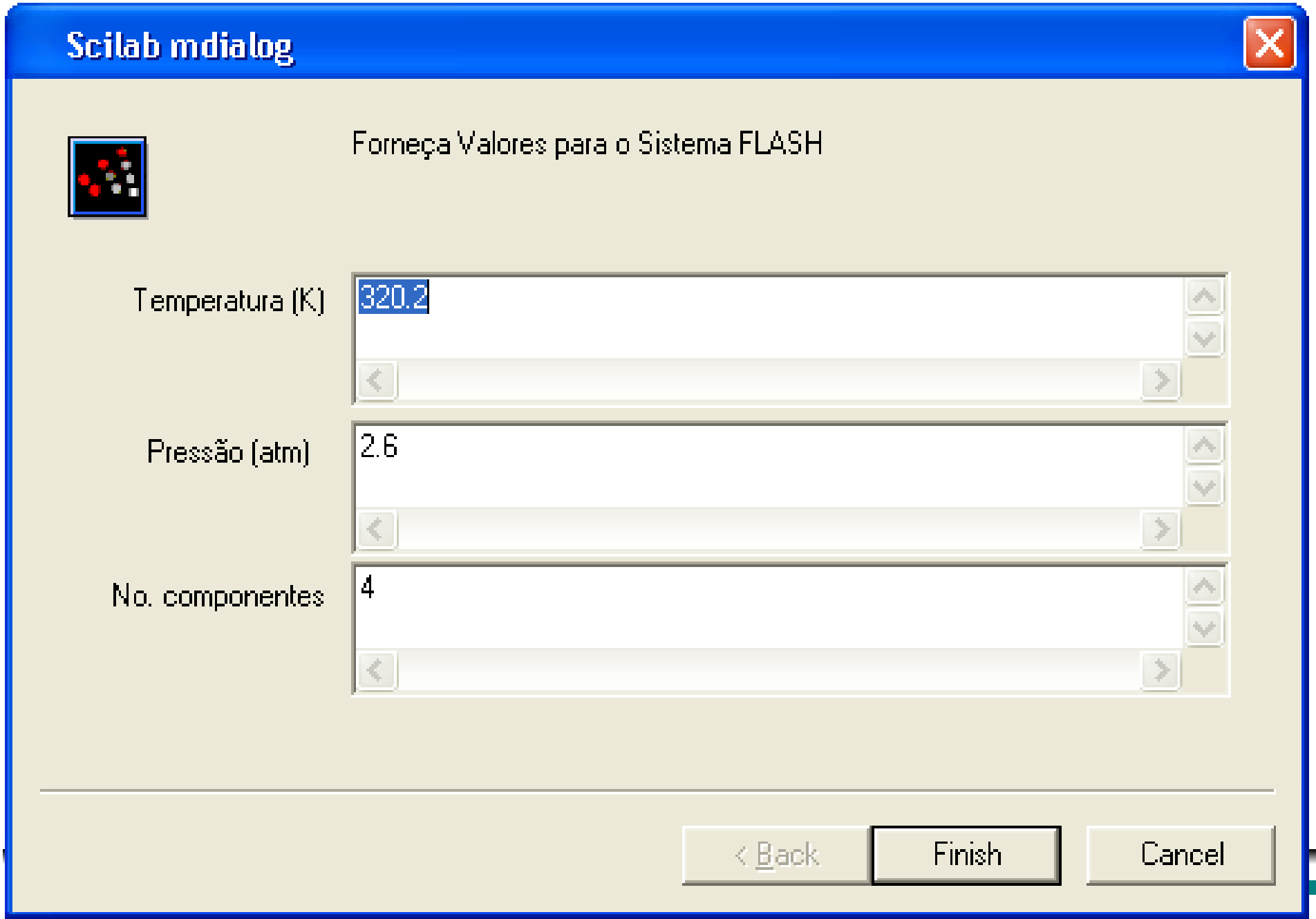
- Tit_janela: Título para a entrada de dados, vetor coluna
- Tit_var: Título para cada variável, vetor coluna
- Tipo: list(tip_1,dim_1,..,tip_n,dim_n)
  "mat": matriz; "col" : vetor coluna; "row" : vetor linha
  "vec" : vetor; "str" : string; "lis": lista
- inicial: vetor coluna de strings com valores padrão
- ok: Se ok=%t: entrada normal; ok=%f entrada cancelada
- xi: valor recebido da variável tip_i

Exemplo de getvalue

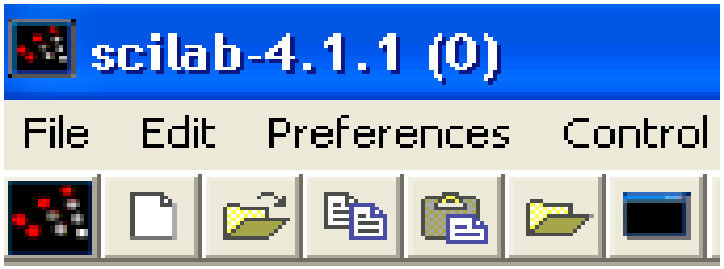



```
-->Tit_var=["Temperatura (K)";"Pressão (atm)  ";"No. componentes"];

-->[ok,T,P,Ncomp]=getvalue("Forneça Valores para o Sistema FLASH",Tit_var,...
-->        list("vec",1,"vec",1,"vec",1),["320.2";"2.6";"4"])
```

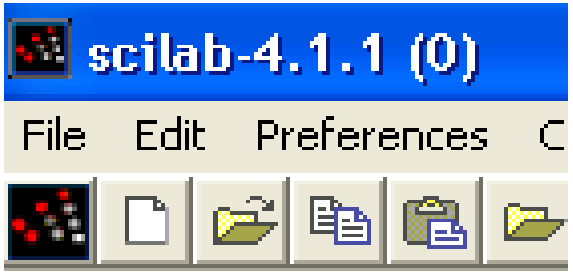
Exemplo de getvalue



Execução de getvalue




```
Ncomp  =

    4.
P   =

    2.6
T   =

    320.2
ok  =

 T
```




```
Ncomp  =

    []
P   =

    []
T   =

    []
ok  =

 F
```

Entrada em formato string: x_dialog, x_mdialog, x_matrix

- x_dialog: janela gráfica
  resultado=x_dialog(labels,valores_iniciais)
  Ex.: G=x_dialog('Valor de G ?','0.54')
  Retorna G string
  Ex.: G=evstr(x_dialog('Valor de G ?','0.54'))
  Retorna G numérico
- evstr: avalia expressões
  H=evstr(Z)
  Ex.: a=evstr('2.4')
  Retorna a=2.4 (numérico)

Podem-se fornecer matrizes

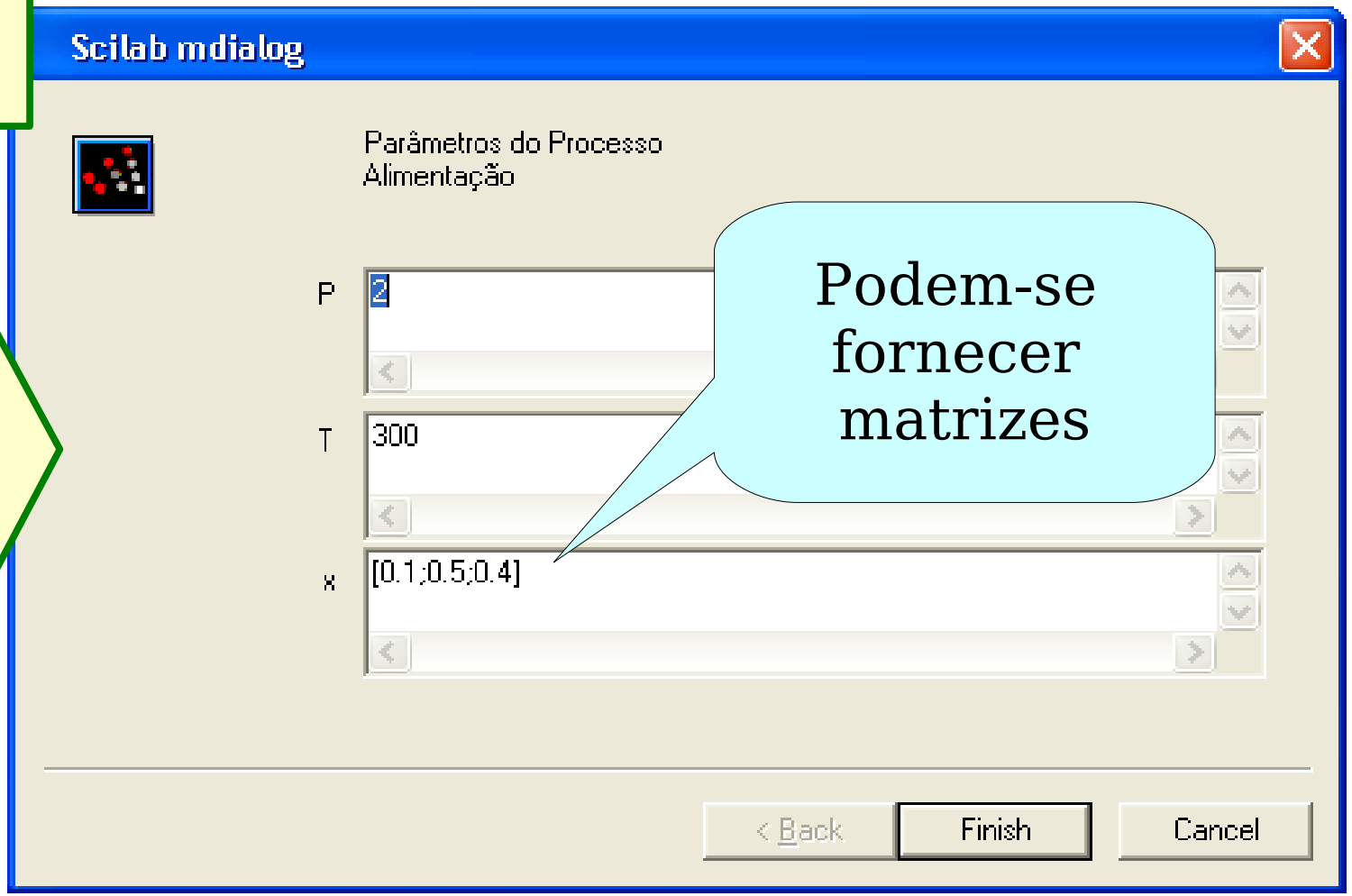
Figura

Entrada em formato string: x_dialog, x_mdialog, x_matrix

- x_mdialog:
- resultado=x_mdialog(título,label_l,label_c,matrix_default)

```
-->resp=x_mdialog(['Parâmetros do Processo';'Alimentação'],...
-->                ['P';'T';'x'],['2';'300';'[0.1;0.5;0.4]'])
```

Resultado é *string*

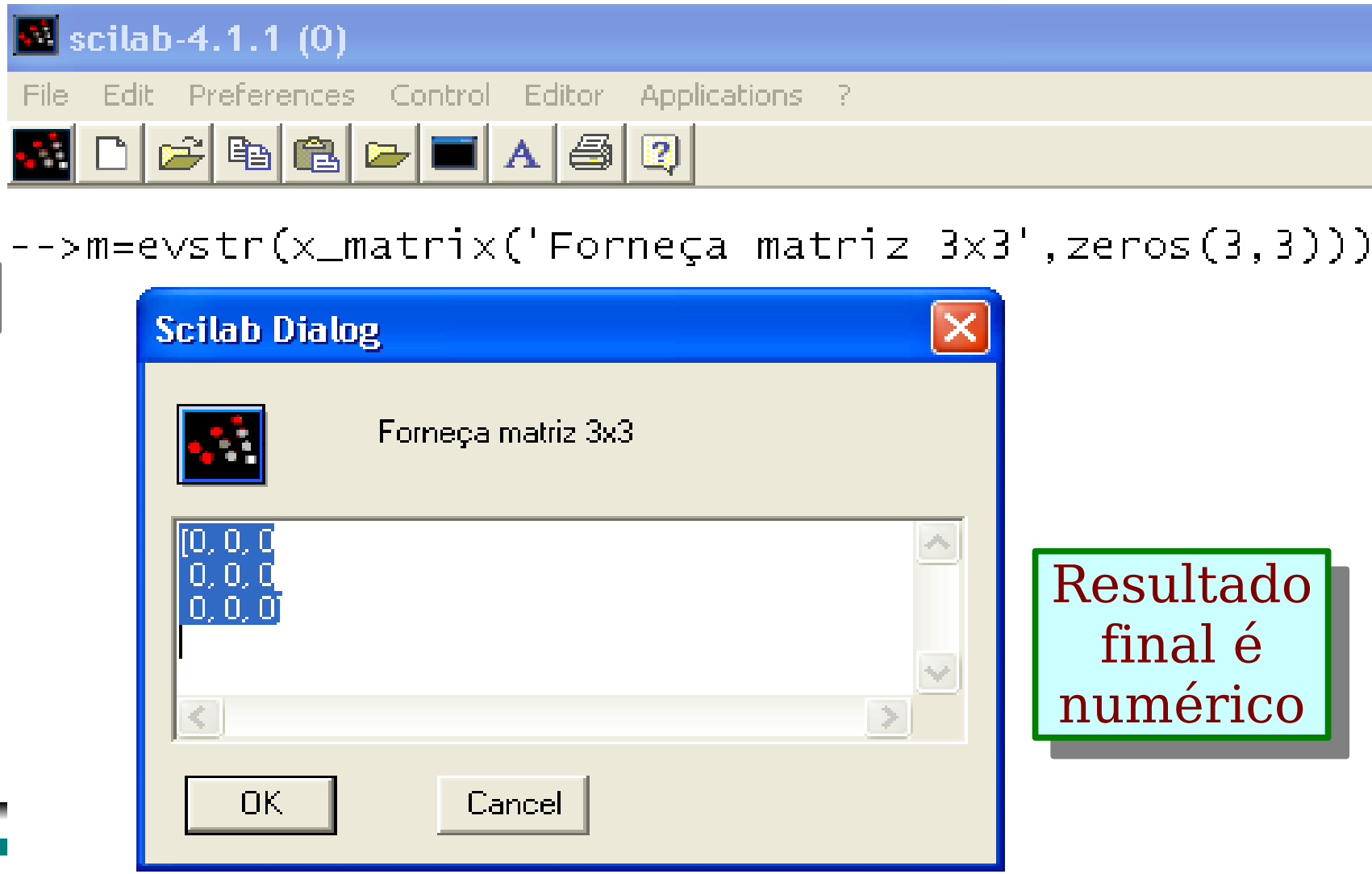


Entrada em formato string: x_dialog, x_mdialog, x_matrix

- x_matrix:
- resp=x_matrix(Título,matrix_default)

Resultado é *string*

evstr



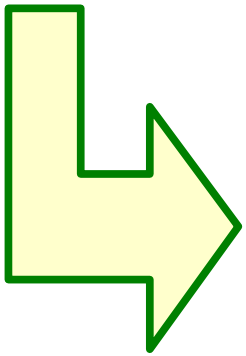
Resultado final é numérico

Entrada em formato matricial: editvar

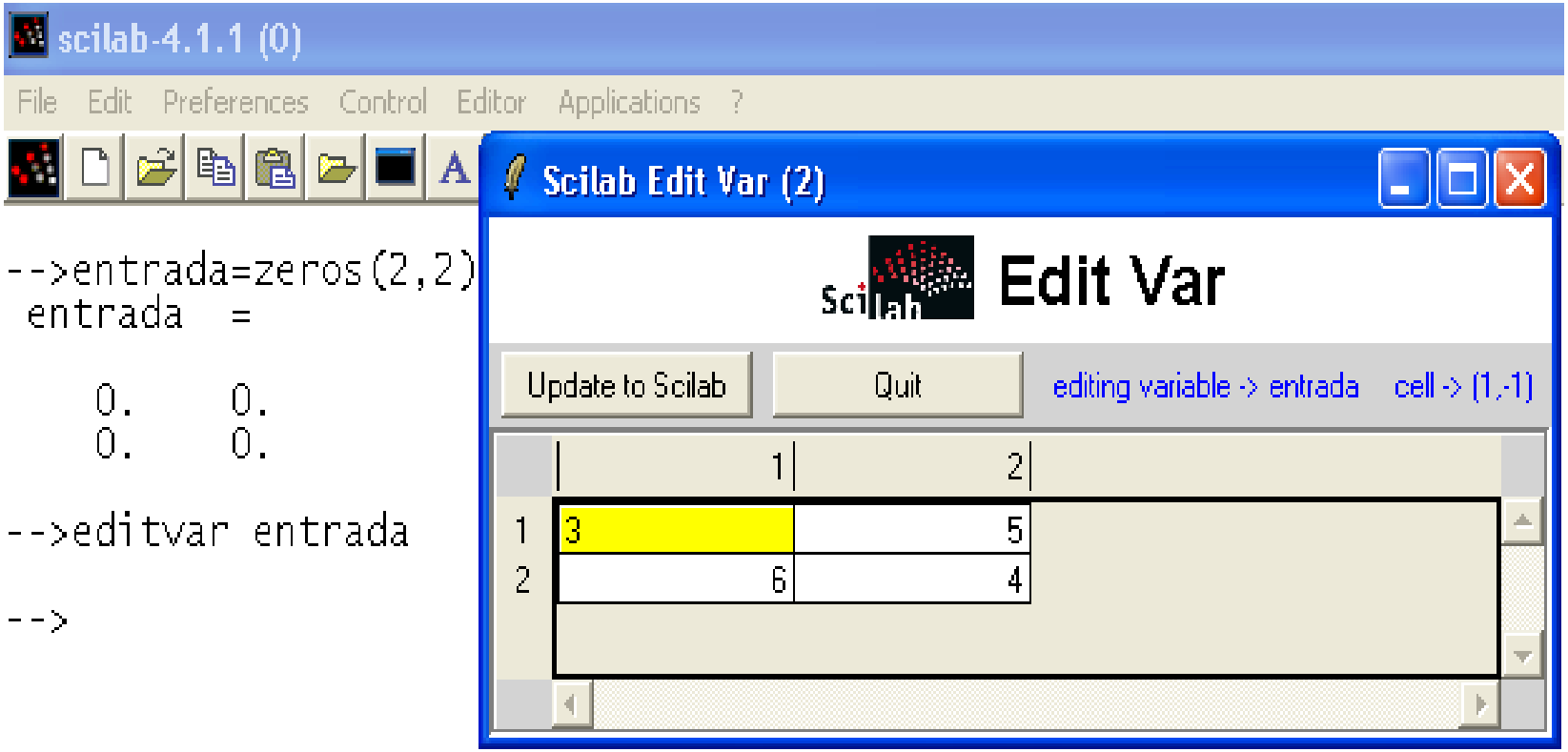
- editvar

  Cria-se uma matriz que receberá os valores (ex. A) e usa-se:

  editvar A

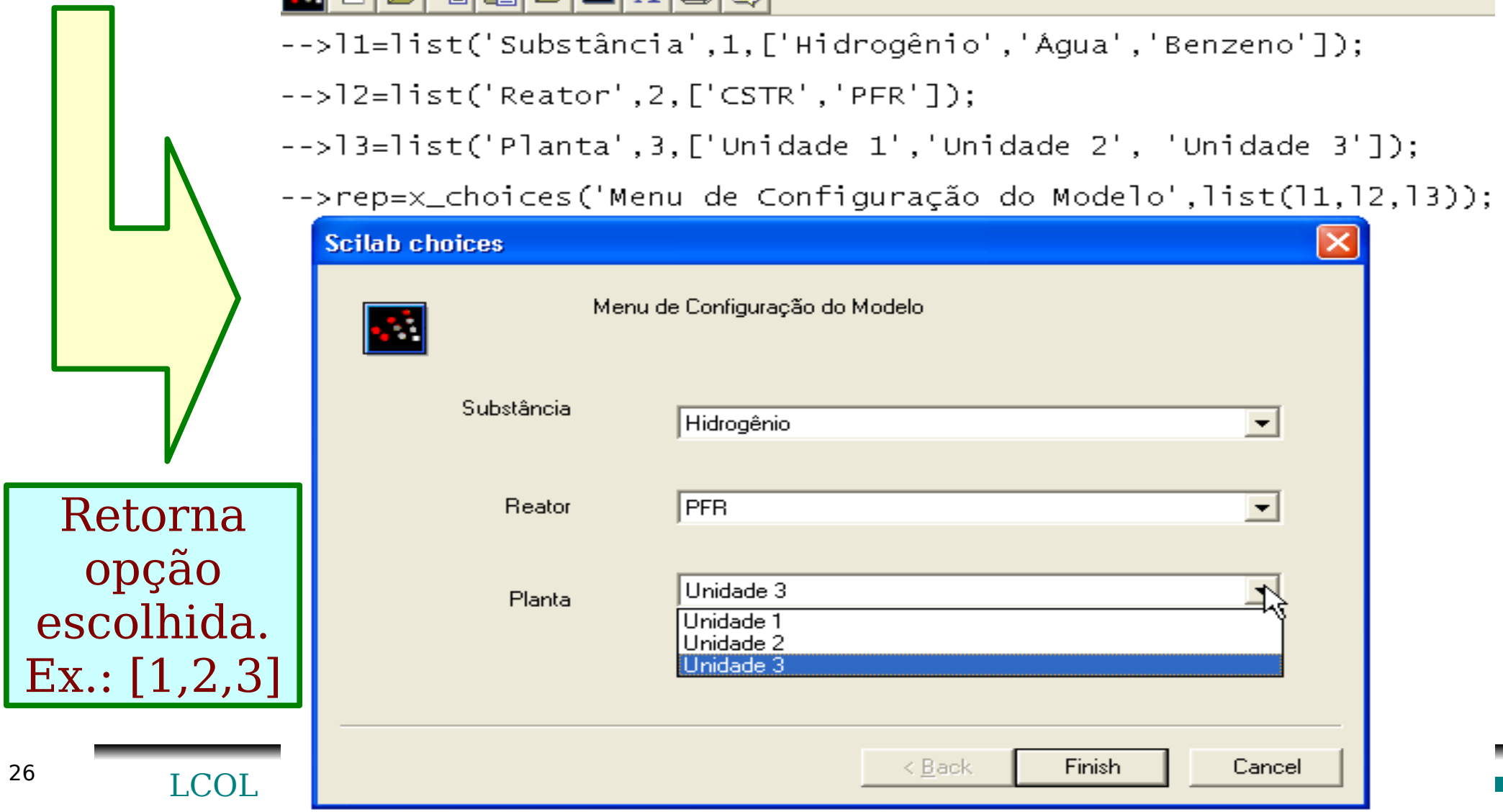


1) Update
2) Quit



Lista de opções: x_choices, x_choose

x_choices: rep=x_choices(Título,Lista_de_itens)
item=list('Texto',escolha_default,opções)



Retorna opção escolhida. Ex.: [1,2,3]

Entrada por lista de opções: x_choices, x_choose

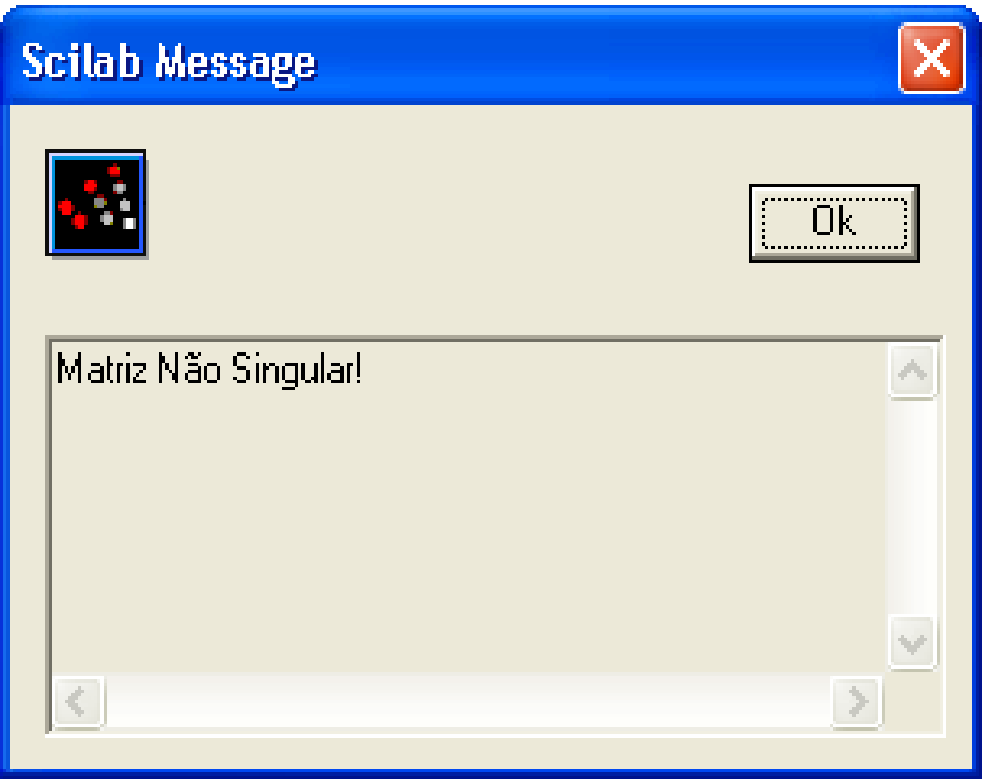
x_choose: [num]=x_choose(itens,Título [,texto para butão])

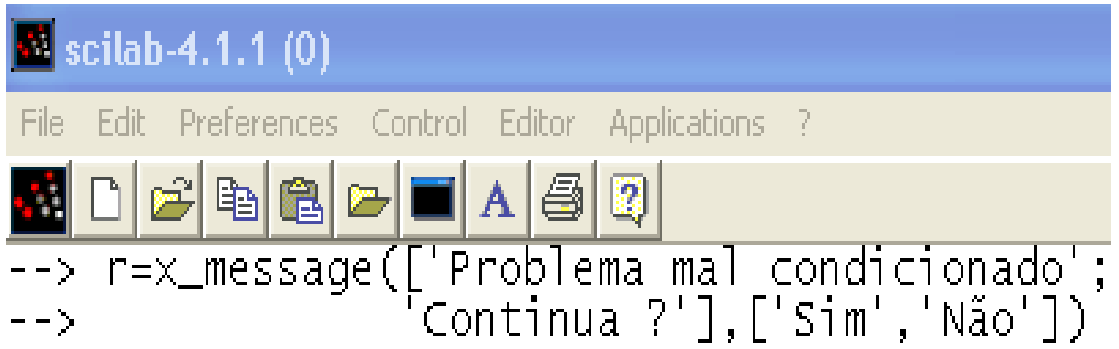

```
-->n=x_choose(['item1';'item2';'item3'],...
-->['Selecione uma opção'],'Cancela')
```

Retorna opção escolhida. Ex.: 2

Mensagens em modo Gráfico: I/O

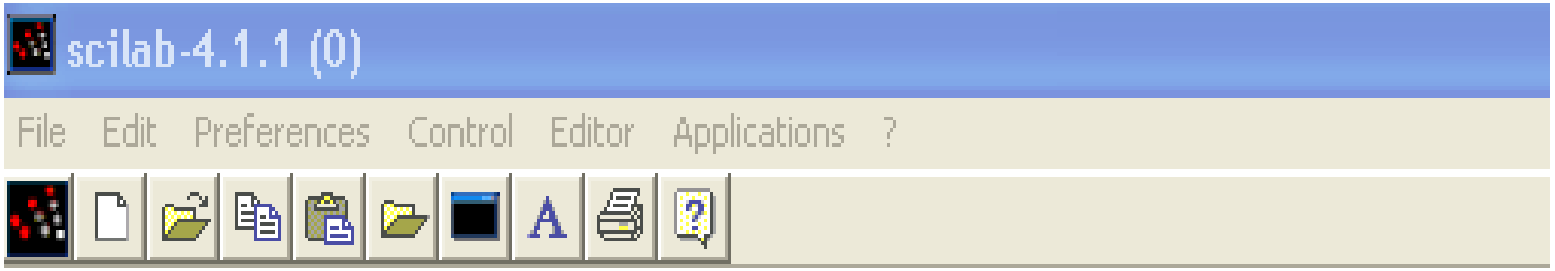
- x_message: mensagem em janela gráfica
  [num]=x_message(strings [,butões])
- Ex.: x_message(['Matriz Não Singular!'])





Retorna opção escolhida. Ex.: 1 ou 2

Mensagens em modo Gráfico: I/O

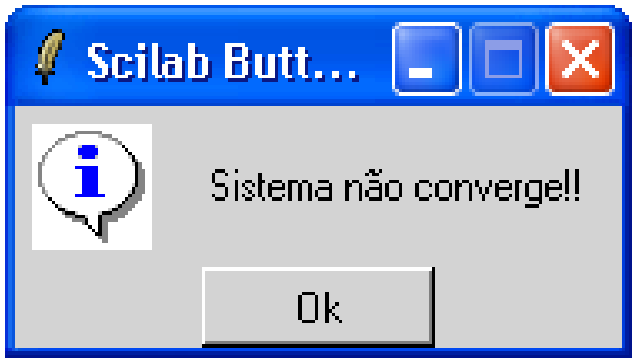



```
-->T=430.25;

-->x_message('Temperatura do reator é :'+string(T)+' K')
```

Uso da função *string* para apresentar valor em modo gráfico

Outras funções - exemplos...

```
-->answ=buttondialog("Continua?","Sim|Não","question");

-->// icone pode ser error, hourglass, info, passwd, question e warning

-->answ=buttondialog("Sistema não converge!!","Ok","info");
```



icone default

Saída de Resultados

Comando	finalidade
;	inibe apresentação de resultado após execução de instrução
disp	mostra objeto na tela do Scilab
file	seleciona unidade lógica e gerencia arquivo
write(%io(2),....)	escreve resultados formatados na janela do Scilab
print(%io(2),....)	escreve resultados formatados na janela do Scilab
printf(formato,valor1,..,valorn)	emula a função **printf** da linguagem C
mprintf(formato,a1,...,an);	converte, formata e escreve dados na janela do Scilab
mfprintf(fd,formato,a1,...,an);	converte, formata e escreve dados para arquivo
str=msprintf(formato,a1,...,an);	converte, formata e escreve dados em uma *string*

- format([tipo][,tamanho])
- %io(2) - tela do scilab

```
-->x=[%pi 100];
-->format('v',10);x
 x   =

    3.1415927    100.
-->format('e',10);x
 x   =

    3.142D+00    1.000D+02
```

Exemplos de saída na tela

scilab-4.1.1 (0)

File Edit Preferences Control Editor Applications

```
-->a=%pi
 a   =

    3.1415927

-->b=2.4;

-->disp(a*b)

    7.5398224

-->disp(a*b);

    7.5398224

-->print(%io(2),a)
 a   =

    3.1415927

-->write(%io(2),a)
  3.1415926536D+00

-->write(%io(2),a,'(1(f8.4))')
  3.1416
```

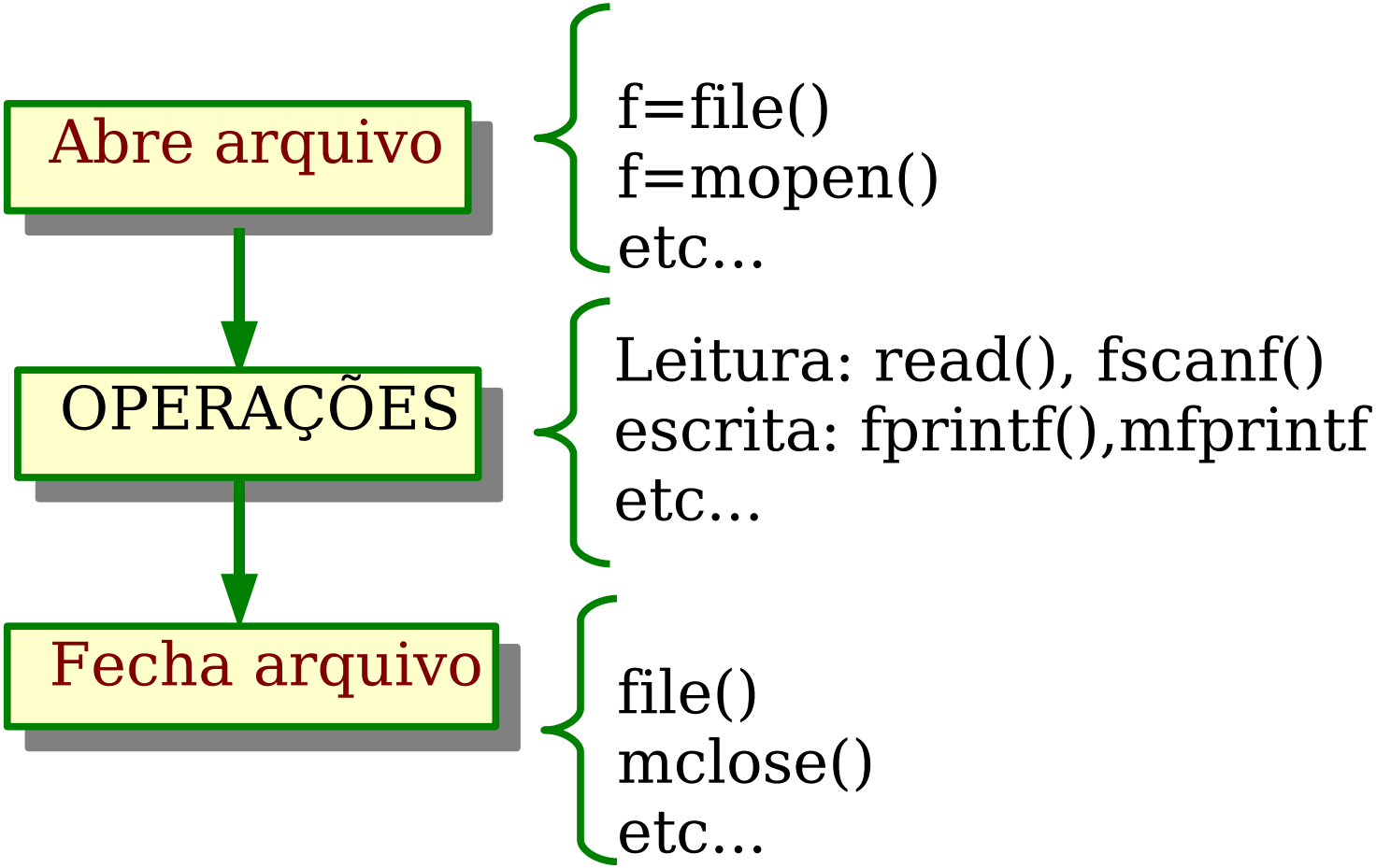
scilab-4.1.1 (0)

File Edit Preferences Control Editor Applications ?

```
-->printf('Resultado:\n a=%f",a)
Resultado:
 a=3.141593
-->mprintf('Resultados:\n a=%f  b=%f',a,b)
Resultados:
 a=3.141593  b=2.400000
-->mprintf('Resultados:\n a=%5.3f  b=%5.2f',a,b)
Resultados:
 a=3.142  b= 2.40
-->msprintf('Resultados:\n a=%f  b=%f',a,b)
 ans  =

!Resultados:                !
!                           !
! a=3.141593  b=2.400000    !
```

write--> formato Fortran
printf--> formato C

Trabalhando com arquivos



Entrada de Dados via arquivo

Sintaxe	Função
[fd,err]=**mopen**(file [, mode, swap])	abre arquivo, compatível com fopen da linguagem C
err=**mclose**([fd]) ou **mclose**('all')	fecha arquivo aberto com mopen
read(arquivo,m,n,[formato])	leitura de matriz de dados linha após linha
fscanf(arquivo,formato)	leitura de dados de arquivo

- mopen -----> mode
  - r ou rb: Abre um arquivo para leitura
  - w ou wb: Cria novo arquivo para escita
  - a ou ab: Abre arquivo para adicionar escrita
  - r+ ou r+b: Abre arquivo para atualizar (ler e escrever)
  - w+ ou w+b: Cria arquivo para atualizar
  - a+ ou a+b: Abre para atualizar com escrita no final do arquivo

Saída de Dados via arquivo

Comando	finalidade
file	seleciona unidade lógica e gerencia arquivo
write(fd,....)	escreve resultados formatados na janela do Scilab
print(fd,....)	escreve resultados formatados na janela do Scilab
fprintf(fd,formato,a1,...,an);	converte, formata e escreve dados para arquivo
mfprintf(fd,formato,a1,...,an);	converte, formata e escreve dados para arquivo

- [unidade [,err]]=file('open', arq [,status] [,acesso [,recl]] [,formt])
- file(ação,unidade)
- [unidade [,tipo [,nomes [,mod [,swap]]]]] = file([unidade])

string	significado
"close"	fecha o(s) arquivo(s) representado(s) pela unidade lógica associada
"rewind"	coloca o ponteiro para o início do arquivo
"backspace"	coloca o ponteiro para o início do último registro
"last"	coloca o ponteiro após o último registro

O Comando *file*

- [unidade [,err]]=file('open', arq [,status] [,acesso [,recl]] [,formt])
- file(ação,unidade)
- [unidade [,tipo [,nomes [,mod [,swap]]]]] = file([unidade])

status

string	significado
"close"	fecha o(s) arquivo(s) representado(s) pela unidade lógica associada
"rewind"	coloca o ponteiro para o início do arquivo
"backspace"	coloca o ponteiro para o início do último registro
"last"	coloca o ponteiro após o último registro

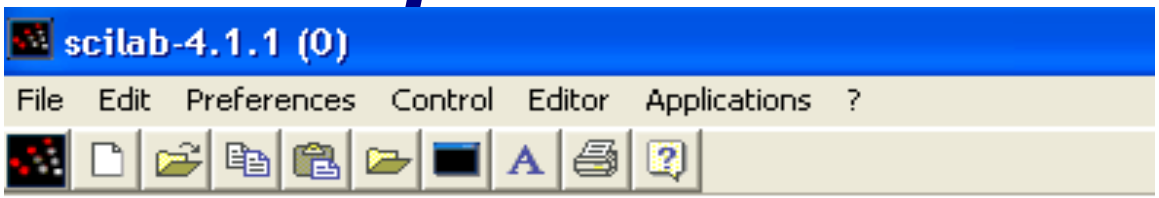
ação

string	significado
"new"	arquivo não pode existir (*default*)
"old"	arquivo deve existir
"unknown"	*status* desconhecido
"scratch"	arquivo a ser apagado no final da seção

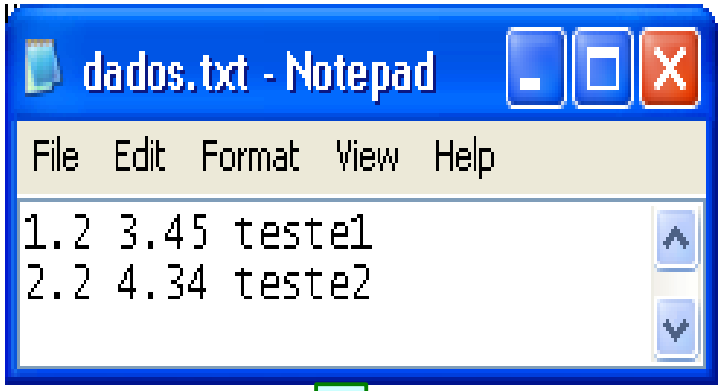
acesso

acesso	significado
"sequential"	acesso seqüencial (*default*)
"direct"	acesso direto

Exemplos : Leitura de Dados




```
-->u1=file('open','C:\dados.txt','unknown');

-->[v1,v2,v3]=fscanf(u1,'%g %g %s')
 v3  =

 teste1
 v2  =

    3.45
 v1  =

    1.2

-->[v4,v5,v6]=fscanf(u1,'%g %g %s')
 v6  =

 teste2
 v5  =

    4.34
 v4  =

    2.2

-->file('close',u1);
```



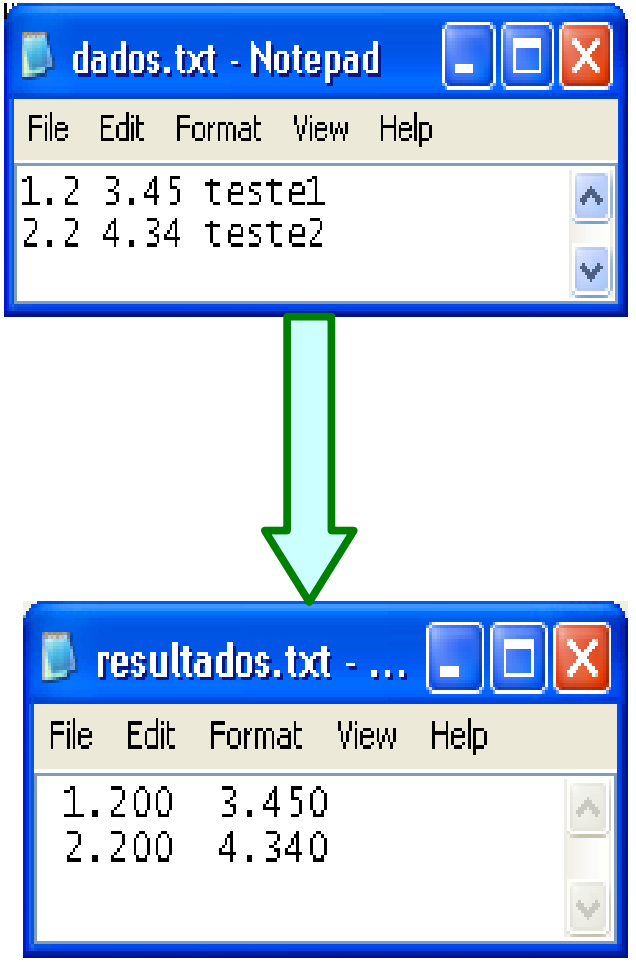
Memória e apresentação na tela

Exemplos: I/O arquivos

scilab-4.1.1 (0)

File Edit Preferences Control Editor Applications ?

```
-->u1=file('open','C:\dados.txt','unknown');
-->u2=mopen('C:\resultados.txt','w');
-->// Leitura de u1
-->[v1,v2,v3]=fscanf(u1,'%g %g %s');
-->[v4,v5,v6]=fscanf(u1,'%g %g %s');
-->// escrita em u2
-->fprintf(u2,'%6.3f %6.3f\n',v1,v2);
-->fprintf(u2,'%6.3f %6.3f\n',v4,v5);
-->file('close',u1);
-->mclose(u2);
```



I/O via arquivo - Funções Especiais

fscanfMat e fprintfMat

- Leitura de Matrizes em arquivos
  - M=fscanfMat(arquivo);
  - [M,texto]=fscanfMat(arquivo);
- Escrita de Matrizes para arquivo
  - fprintfMat(arquivo,M [,formato,texto]);
- Com:
  - arquivo: string com caminho/nome do arquivo.
  - M : Matriz real.
  - formato : Formato tipo C, default é "%f"
  - texto : Matriz de string para início de arquivo

As funções fscanMat e fprintfMat:
- Abrem arquivo
- Lêem informações de Matrizes
- Fecham arquivo

Usando fscanfMat e fprintfMat

scilab-4.1.1 (0)

File Edit Preferences Control Editor Applications ?

```
-->texto=['****Texto Explicativo ****';
-->       ' Continua texto....';
-->       ' Finaliza texto.'];

-->M=rand(6,6);

-->fprintfMat('C:\saida.txt',M,'%5.2f',texto);

-->[N,texto2]=fscanfMat('C:\saida.txt')
 texto2  =

!****Texto Explicativo ****  !
!                            !
! Continua texto....         !
!                            !
! Finaliza texto.            !
 N  =

    0.23    0.39    0.12    0.65    0.85    0.82
    0.63    0.83    0.29    0.99    0.06    0.06
    0.76    0.59    0.86    0.05    0.83    0.56
    0.05    0.48    0.85    0.75    0.93    0.12
    0.67    0.22    0.53    0.41    0.57    0.73
    0.2     0.84    0.99    0.61    0.57    0.27
```

saida.txt - Notepad

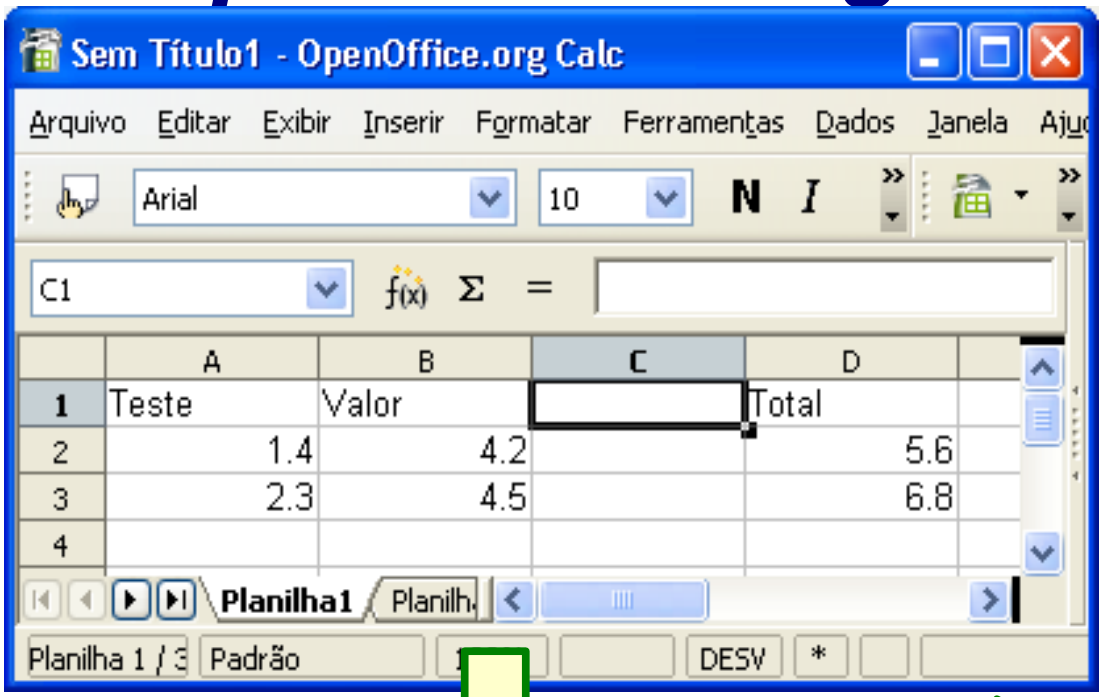
File Edit Format View Help

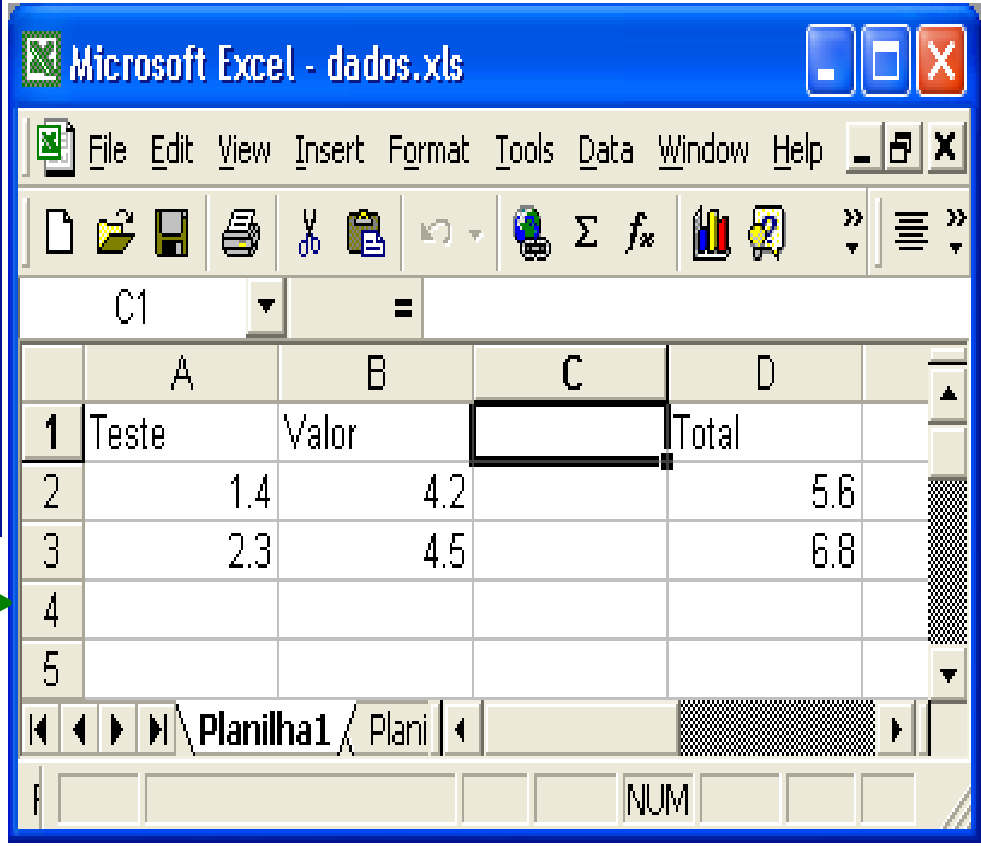
```
****Texto Explicativo ****
 Continua texto....
 Finaliza texto.
 0.23  0.39  0.12  0.65  0.85  0.82
 0.63  0.83  0.29  0.99  0.06  0.06
 0.76  0.59  0.86  0.05  0.83  0.56
 0.05  0.48  0.85  0.75  0.93  0.12
 0.67  0.22  0.53  0.41  0.57  0.73
 0.20  0.84  0.99  0.61  0.57  0.27
```

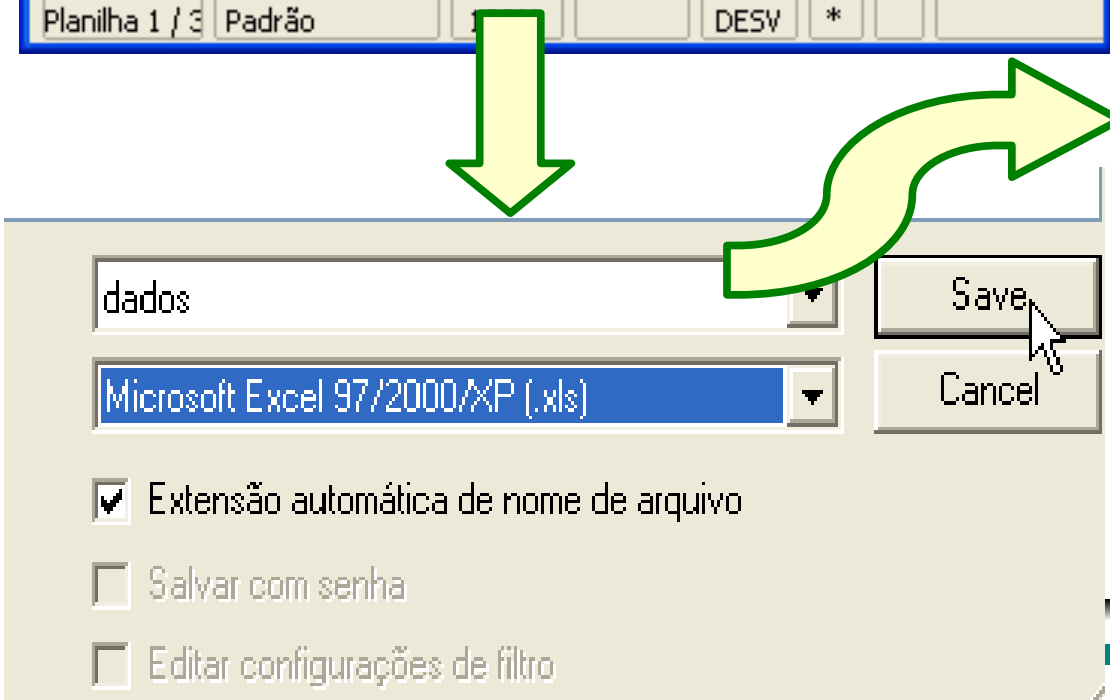
I/O via arquivo - Funções Especiais
xls_open e xls_read

- Leitura de arquivos Excel
  - Abertura do arquivo
  - Leitura do Arquivo
- Abertura do arquivo
  - [fd,SST,NomeFolha,PosFolha] = xls_open(arquivo)
- Leitura do arquivo
  - [M,TextInd] = xls_read(fd,PosFolha)
- Com
  - arquivo : string com caminho/nome arquivo Excel.
  - fd : Unidade lógica.
  - SST : Vetor strings da Folha Excel.
  - NomeFolha: Vetor de strings com nomes das folhas.
  - PosFolha: Vetor de números com posição das folhas.
  - M: Matriz de números, NaN se vazia ou texto.
  - TextInd : Matrix de índices.

Preparando arquivos Excel com OpenOffice.org Calc







Excel **NÃO** é necessário!!!

Leitura de Arquivo Excel

scilab-4.1.1 (0)

File Edit Preferences Control Editor Applications ?

```
-->[fd,SST,NomesFolha,PosFolha] = xls_open('C:\dados.xls')
 PosFolha  =

    1471.    2059.    2455.
 NomesFolha  =

!Planilha1  Planilha2  Planilha3  !
 SST  =

!Teste  Valor  Total  !
 fd  =

    1.

-->[M,TextInd] = xls_read(fd,PosFolha)
 TextInd  =

    1.    2.    0.    3.
    0.    0.    0.    0.
    0.    0.    0.    0.
 M  =

   Nan    Nan    Nan    Nan
    1.4    4.2   Nan     6.
    2.3    4.5   Nan     8.
```

Abertura arquivo Excel

Leitura de arquivo

Gráficos no Scilab

- Plot(): gráficos 2D
  plot(x1,y1,[Esp_L1>][,...xN,yN][,Espec_LN][,Prop1][,..PropM])

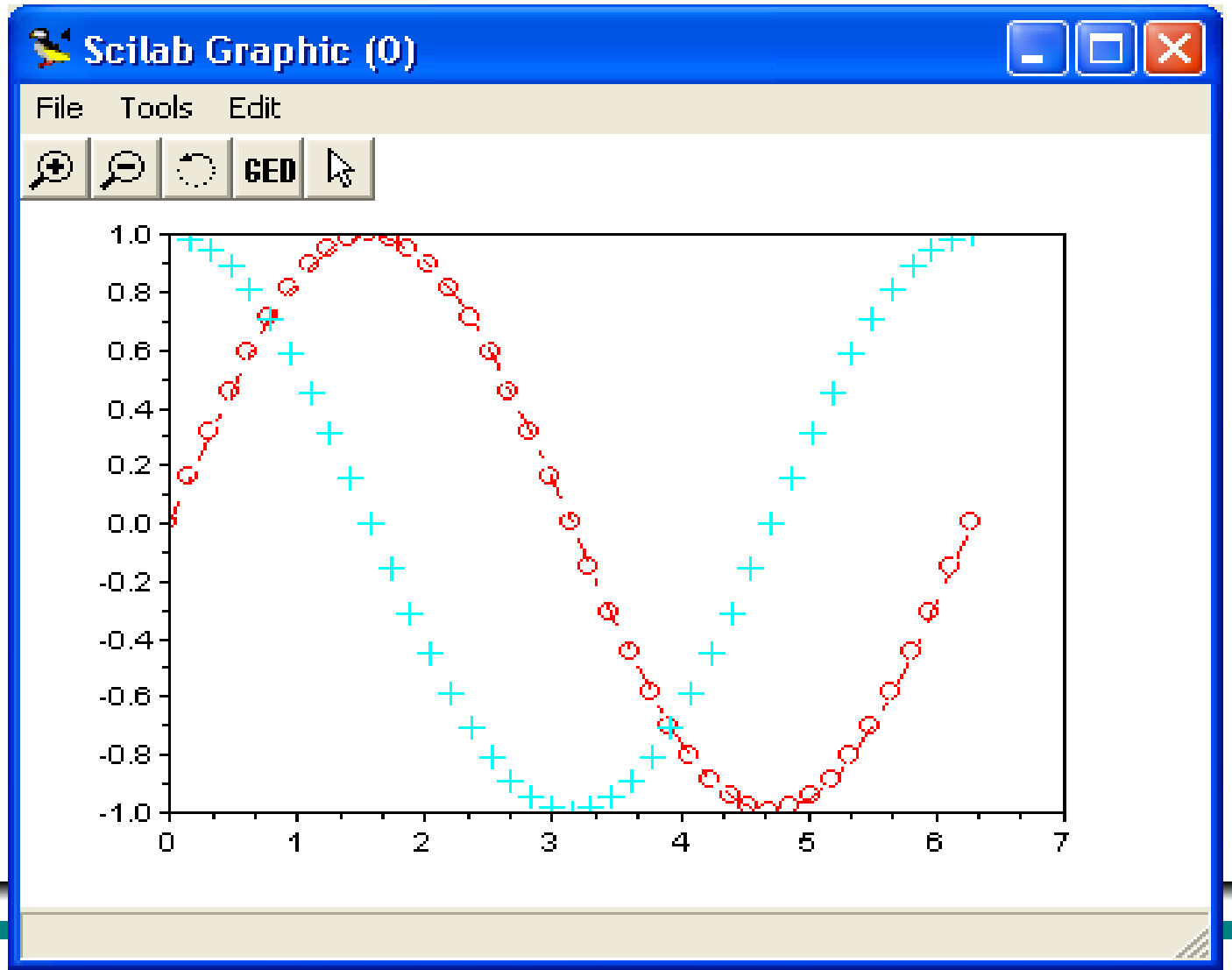
Esp_L
-
--
:
-.

cor
r
g
b
c
m
y
k
w

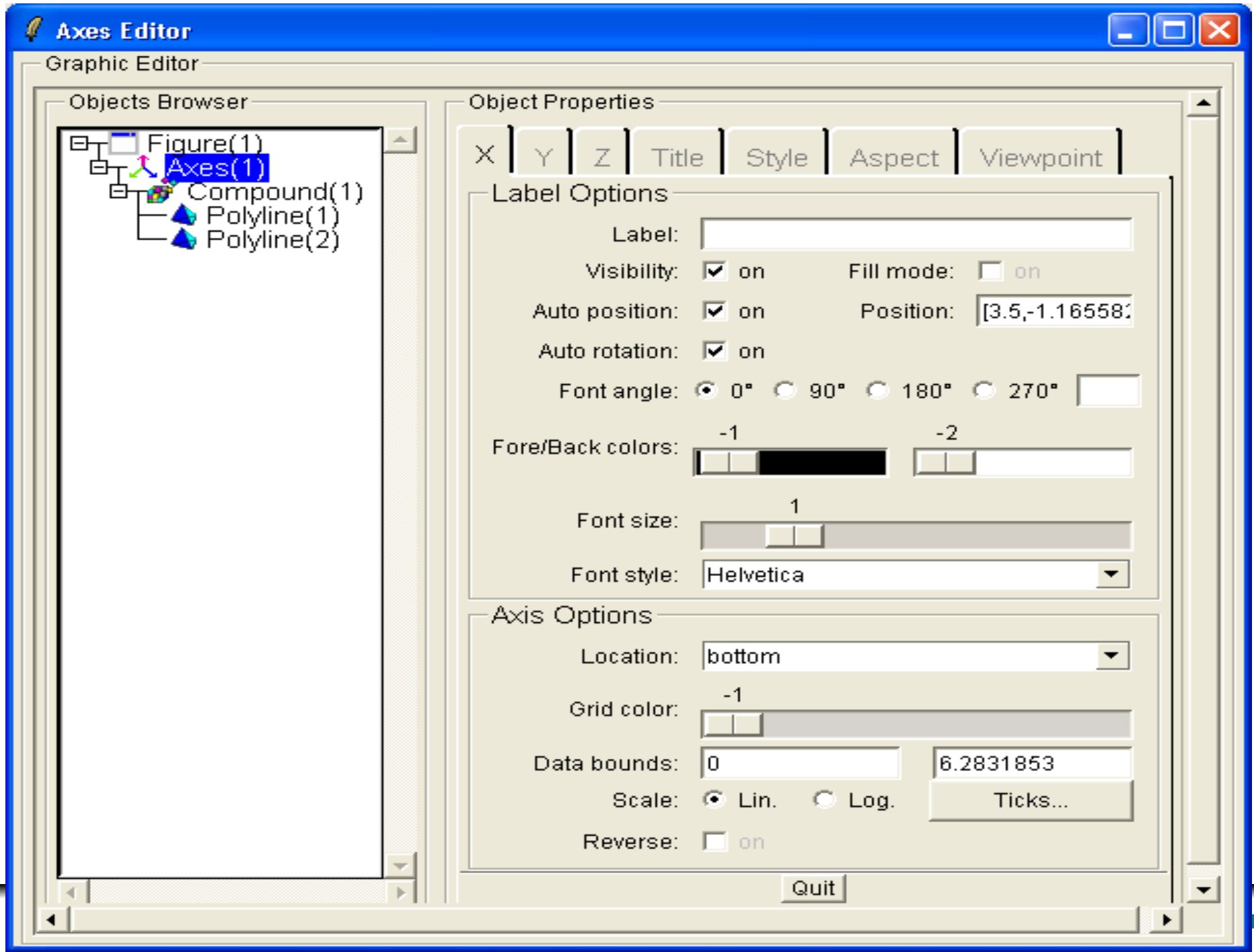
pontos
+
o
*
.
x
'square' or 's'
'diamond' or 'd'
^
v
>
<
'pentagram'
'none'

Exemplo plot

```
-->t=0:%pi/20:2*%pi;

-->plot(t,sin(t),'ro-.',t,cos(t),'cya+')
```



Formatando Gráfico



Funções Gráficas no Scilab

- clf(): limpa janela gráfica ativa
- subplot(): divide janela gráfica
- xtitle() : coloca títulos no gráfico e eixos.
- xgrid() : adiciona malha
- get() ..... etc

Exercícios,....

Funções no Scilab...

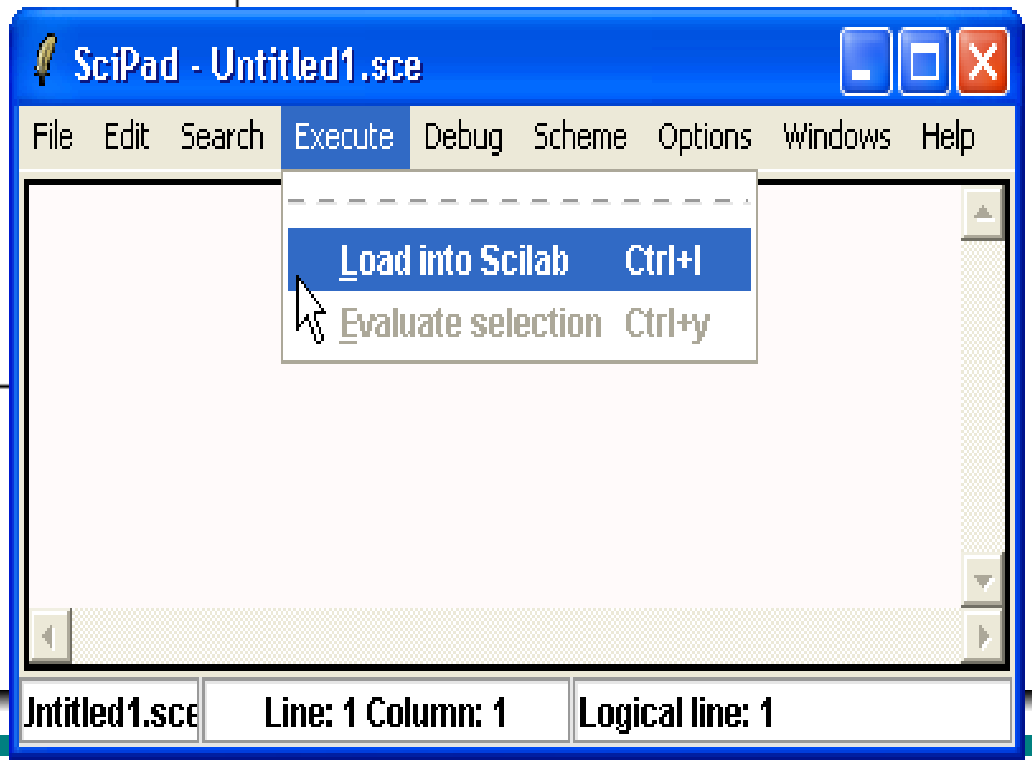
funções...

Programando Funções

[// Apresentação_do_Algoritmo]
[// informações sobre o código e sua utilização]
[// Detalhes do seu desenvolvimento]
[//Pré-processamento]
    Especificação de variáveis internas do Scilab
    Carregamento de bibliotecas para a memória
[//Funções]
    Declaração das funções
[// Início]
    Corpo do algoritmo
[//Fim]

Pré-processamento:

- clc
- clear
- mode(-1) ou mode(0)
- lines(0)
- ieee(mod),0 ou 1 ou 2



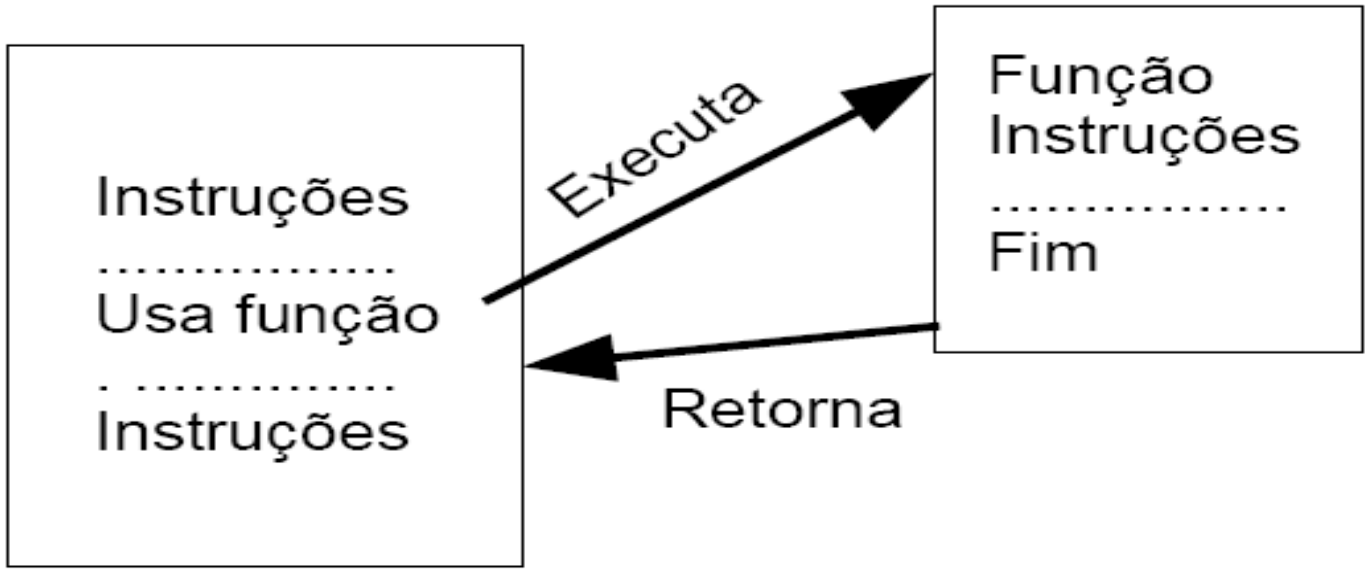
Funções no Scilab

```
function [y1,y2,...,yn]=nome_func(x1,x2,...,xm)
        .........

endfunction
```

ou

deff() - definição em linha

extensão	uso
.sce	arquivo para execução direta
.sci	arquivo de funções para carregar em memória



Variáveis:
- LOCAIS
- GLOBAIS

Exercícios,....

Referências sobre este documento

- O mesmo pode ser encontrado para download no site
- http://www.moodle.ufu.br/file.php/270/Material_Complementar/
- Material sobre Scilab ...
- Outros,...
- Agradecimentos: A todos que promovem a idéia do SL
- **Contato:** lcol@ufu.br

Obrigado!!