



Curso de Scilab



Luís Cláudio OLIVEIRA-LOPES

Parte III

08 de Outubro de 2007

Uberlândia, Minas Gerais, Brasil

SUMÁRIO



Mascote
(*puffin*)

- Resolvendo problemas no Scilab
 - Sistemas Eq. Lineares
 - Sistemas Eq. Não Lineares
 - EDOs
 - Otimização
 - Regressão Numérica
- Estudo de Casos
 - Balanço de Massa
 - Sistema Químico/ Eq. de Estado
 - Sistemas Dinâmicos
 - Ajuste de Dados
- Exercícios





Programando no Scilab...



Programando...

```
// Estudo de funções no Scilab
// Exemplo 1
// Autor: Scilábio da Silva
// 06/10/2007
```

```
//-----
// pré-processamento
```

```
clc
clear
mode(-1)
lines(0)
```

```
//-----
// Funções
```

```
//-----
// Programa Principal
```

```
//-----
// Entrada de Dados
```

```
// Processamento
```

```
// Saída de Resultados
```

```
//Fim de Programa
disp('* FIM *')
```

Template:
***.sce**

pré-processamento

Funções

Entrada de Dados

processamento

Funções

```
SciPad - template_func.sce
File Edit Search Execute Debug Scheme Options Windows Help

function [sai_1,sai_2,...]=nome_funcao(ent_1,ent_2,...)
//-----
// Template de função no Scilab
// Autor: Scilábio da Silva
// 08/10/2007
// Uso:
// [sai_1,sai_2,...]=nome_funcao(ent_1,ent_2,...)
//-----
comando_a;
comando_b;
....
sai_1=comando_1;
sai_2=comando_2;
....
endfunction
//Fim da função
```

Template de função: *.sci

Parâmetros entrada

Parâmetros saída

Line: 8 Column: 50

Line 7 in nome_funcao

Definindo funções em linha: *deff*()

- `deff('[s1,s2,...]=nome_funcao(e1,e2,...)',texto [,opt])`
- **com:**
 - `e1,e2,...` : variáveis entrada
 - `s1,s2,...` : variáveis saída.
 - `texto` : matriz de comandos em *strings*
 - `opt` : string opcional
 - `'c'` : função "processada previamente pela scilab" (padrão)
 - `'n'` : função "não processada previamente pela scilab"

Exemplo:

```
deff('[x]=transforma(y,z)', ['a=3*y+4'; 'x=a*z+y-3'])
```

```
--> deff('[x]=transforma(y,z)', ['a=3*y+4'; 'x=a*z+y-3'])
```

```
--> c=transforma(2,3)
```

```
c =
```

```
29.
```



Resolvendo Problemas: Eq. Algébricas Lineares...



**Eq. Alg.
Lineares...**

Resolução de Eq. Alg. Lineares

- Usando inversa: $x = \text{inv}(A) * b$
- Usando divisão à esquerda: $x = A \setminus b$
- Usando função `linsolve()`
- Usando `rref()` - Eliminação de Gauss

Sintaxe de `linsolve()`

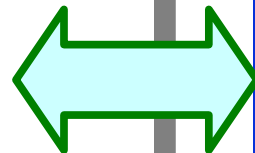
$$[x_0, \text{ker}A] = \text{linsolve}(A, c, [x])$$

$$Ax + c = 0 \rightarrow Ax = b = -c$$

x_0	solução particular do sistema (se existir)
$\text{ker}A$	<i>nullspace</i> de A .

Resolvendo $Ax=b$

$$\begin{cases} 2x + 3y - 5z = -7 \\ 6x - 2y + z = 5 \\ x + 3y - z = 4 \end{cases}$$



```
scilab-4.1.1 (0)
File Edit Preferences Control Editor Applications ?
-->A=[2 3 -5; 6 -2 1; 1 3 -1]
A =
    2.  3.  -5.
    6.  -2.  1.
    1.  3.  -1.
-->b=[-7; 5; 4]
b =
   -7.
    5.
    4.
```

SOLUÇÃO

```
scilab-4.1.1 (0)
File Edit Preferences Control Ed
-->w=linsolve(A, -b)
w =
    1.
    2.
    3.
```

```
-->w=inv(A)*b
w =
    1.
    2.
    3.
```

```
-->w = A\b
w =
    1.
    2.
    3.
```

Resolução de Eq. Alg. Lineares

Sintaxe de rref()

$$B = \text{rref}([A \ b])$$

```
scilab-4.1.1 (0)
File Edit Preferences Control Editor Applications ?
-->sol=rref([A b])
sol =
    1.    0.    0.    1.
    0.    1.    0.
    0.    0.    1.
-->w=sol(:, $)
w =
    1.    2.    3.
```



Resolvendo Problemas: Eq. Algébricas Não Lineares...



Eq. Alg. Não
Lineares...

Resolução de Eq. Alg. Não Lineares

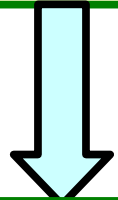
$$[x, v, info] = fsolve(x0, fct, fjac, tol)$$

parâmetros		descrição
x0		vetor de estimativa inicial
fct		função ou lista de <i>string</i> (external)
fjac		função da matriz jacobiana ou lista de <i>string</i> (external)
tol		tolerância, <i>default</i> : tol=1.e-10
x		vetor com valor final estimado
v		vetor com valor da função em x
info		indicador de término
	0	parâmetros de entrada inadequados
	1	erro relativo é no máximo igual a tol
	2	número de chamadas a função excedido
	3	tol é muito pequeno. Não se pode melhorar solução
	4	interação não leva a convergência

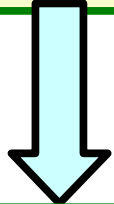
Solução ok!

Resolvendo: $f(x)=0$

Definir
função



Carregar função
na memória

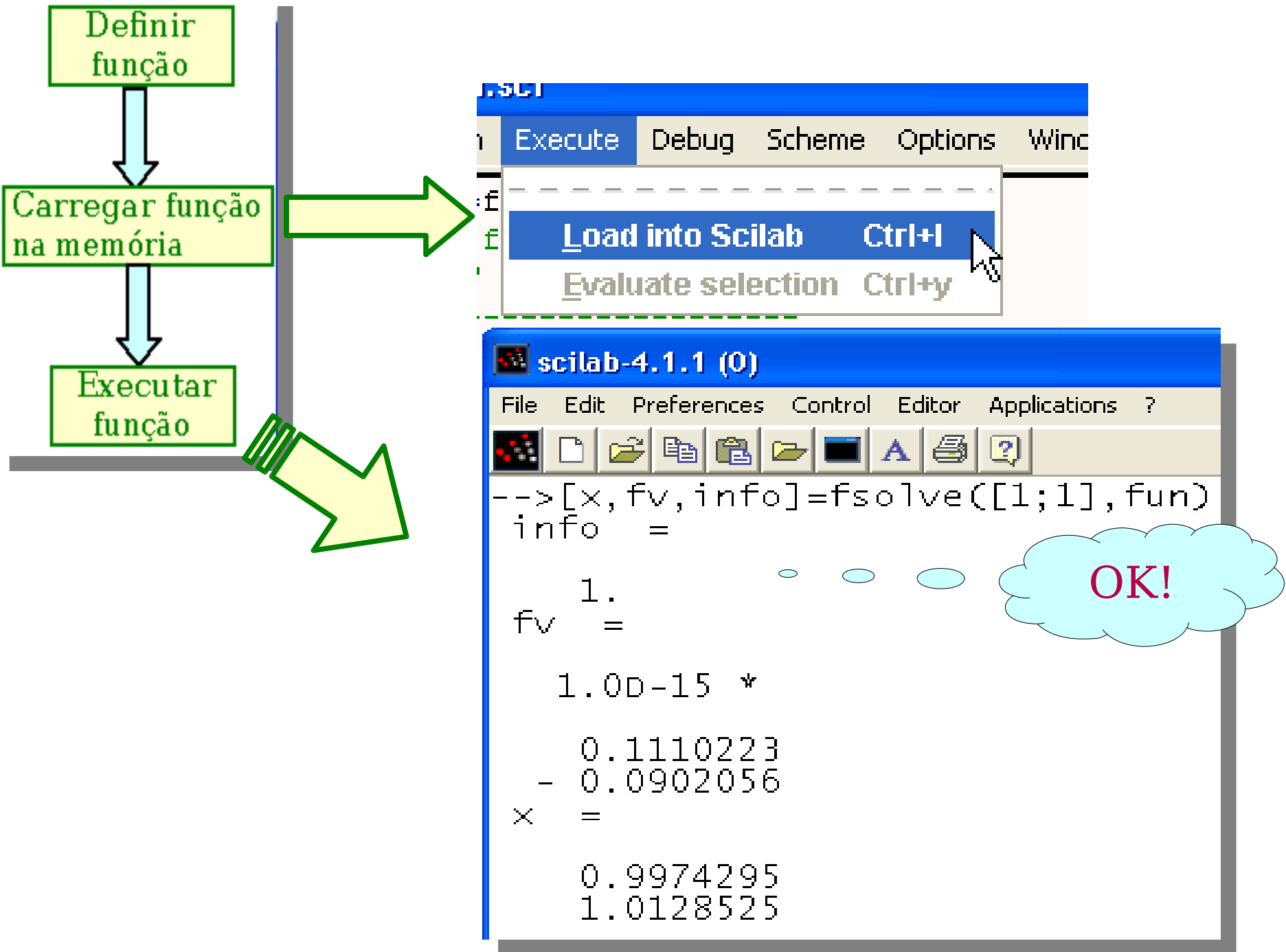


Executar
função

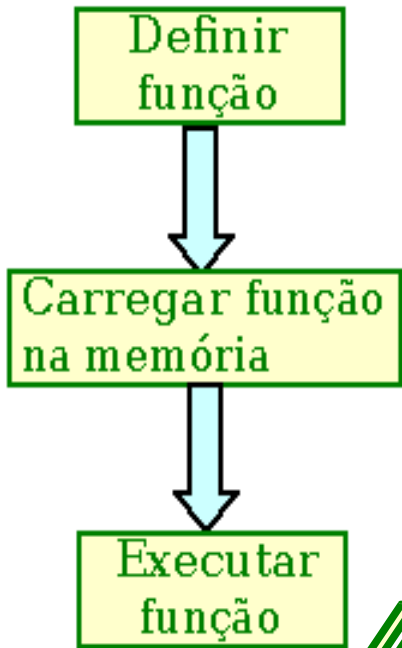
```
SciPad - fun.sci
File Edit Search Execute Debug Scheme Options Windows Help

function [f]=fun(x)
// Estudo de fsolve no Scilab
// 08/10/2007
//-----
B=1; g=1; t=50; de=10; dh=10;
f(1)=1-x(1)*(1+t*exp(-de/x(2)));
f(2)=(1-x(2))+B*(g-x(2))+dh*x(1)*t*exp(-de/x(2));
endfunction
```

Line: 7 Column: 2 Line 7 in fun



Mas,...



```
-->[x, fv, info]=fsolve([0.5;2], fun)
info =
    1.
fv =
    1.0D-15 *
    - 0.2220446
    0.2220446
x =
    0.8418288
    1.7908561
```



```
-->[x, fv, info]=fsolve([0;6], fun)
info =
    1.
fv =
    1.0D-15 *
    0.3330669
    0.
x =
    0.1116045
    5.4419775
```



Multiplicidade de Soluções Estacionárias

Passando parâmetros para *fsolve()*

Uso de Lista
-Sintaxe-

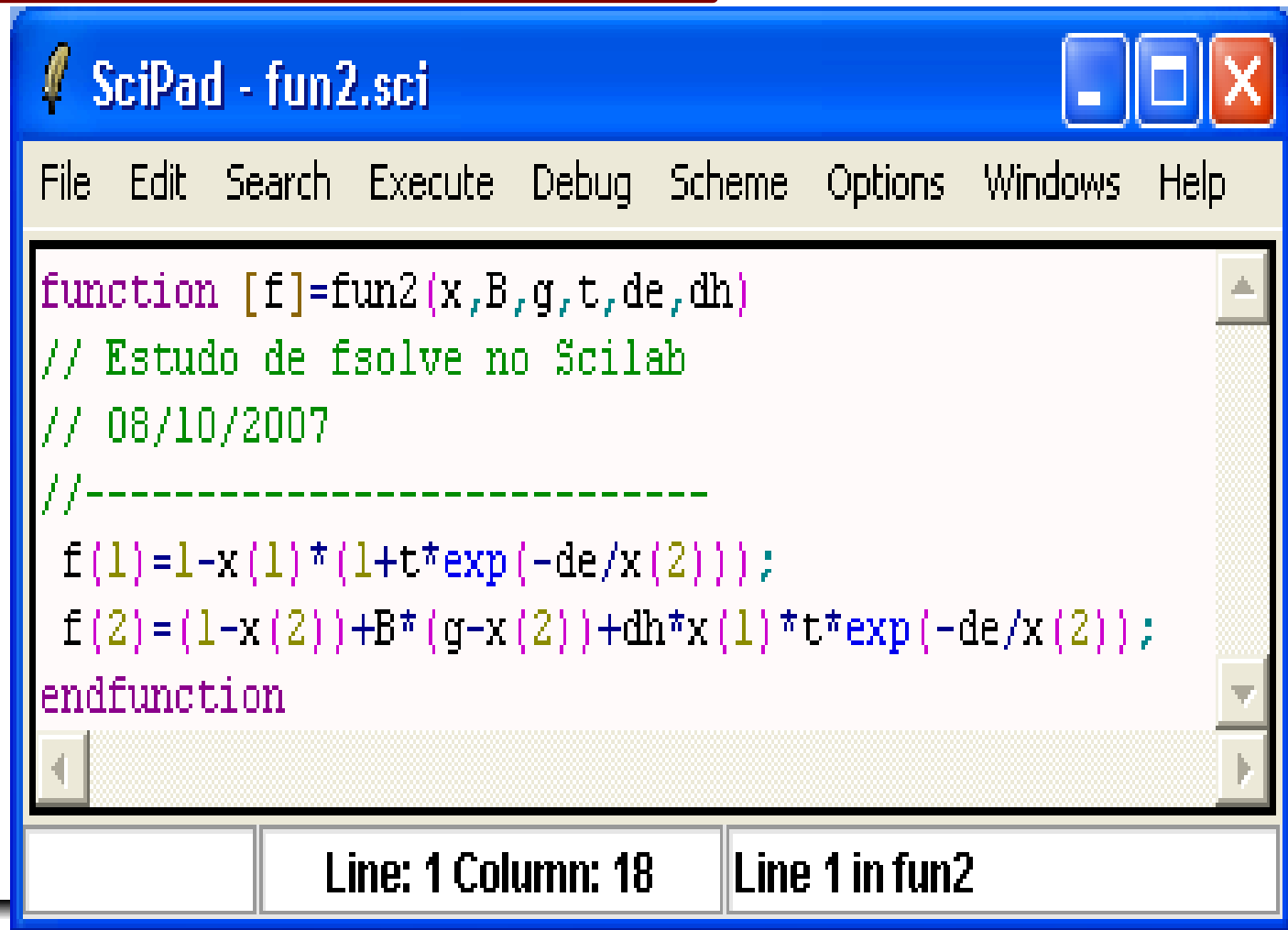
```
lista=list(nome_func,p_1,p_2,p3,...)  
[x,fv,info]=fsolve(x0,lista)
```

```
function [f]=funcao(x,a,b,c)  
    .....  
endfunction  
.....  
// Programa  
.....  
flist=list(funcao,a,b,c);  
[x,fv,iflag]=fsolve(x0,flist);
```


Passando parâmetros para *fsolve()*

Uso de Lista
-Sintaxe-

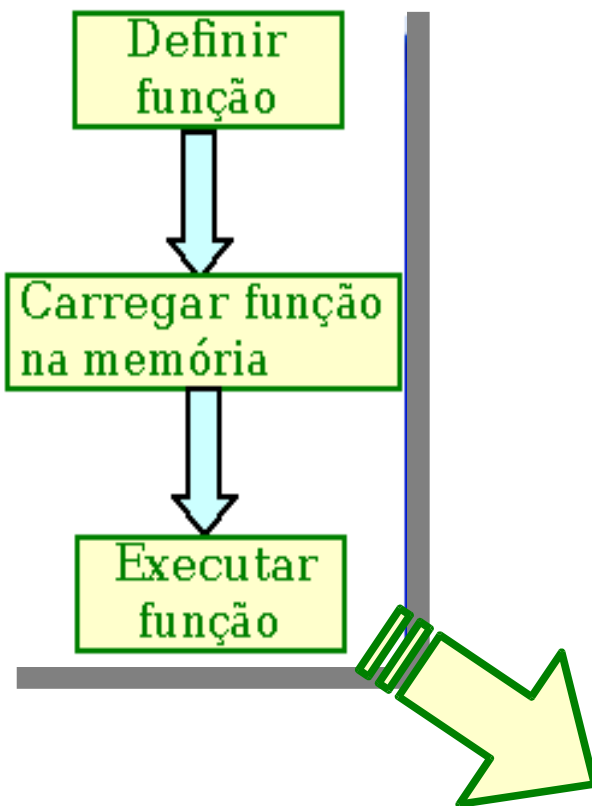
```
lista=list(nome_func,p_1,p_2,p3,...)  
[x,fv,info]=fsolve(x0,lista)
```



```
function [f]=fun2(x,B,g,t,de,dh)  
// Estudo de fsolve no Scilab  
// 08/10/2007  
//-----  
f(1)=1-x(1)*(1+t*exp(-de/x(2)));  
f(2)=(1-x(2))+B*(g-x(2))+dh*x(1)*t*exp(-de/x(2));  
endfunction
```

Line: 1 Column: 18 Line 1 in fun2

Passando parâmetros para *fsolve()*



Uso de Lista
-Sintaxe-

```
lista=list(nome_func,p_1,p_2,p3,...)  
[x,fv,info]=fsolve(x0,lista)
```

```
scilab-4.1.1 (0)  
File Edit Preferences Control Editor Applications ?  
-->B=1; g=1; t=50; de=10; dh=10;  
-->lista=list(fun2,B,g,t,de,dh);  
-->[x,fv,info]=fsolve([0.5;2],lista)  
info =  
  
1.  
fv =  
  
1.0D-15 *  
  
- 0.2220446  
0.2220446  
x =  
  
0.8418288  
1.7908561
```



Resolvendo Problemas: Eq. Diferenciais Ordinárias...



EDOs...

Resolução de EDOs:

PVI

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0$$

```
[y,rd,w,iw]=ode(tipo,y0,t0,t [,rtol [,atol]],f [,jac],ng,g [,w,iw])
```

Sintaxe de função:

```
function dydt=f(t,y)
.....
endfunction
```

Ex.:

```
// dy/dt=y^2-y sin(t)+cos(t), y(0)=0
function dydt=f(t,y),dydt=y^2-y*sin(t)+cos(t),endfunction
y0=0;t0=0;t=0:0.1:%pi;
y=ode(y0,t0,t,f)
plot(t,y)
```

Opções do Comando `ode()`

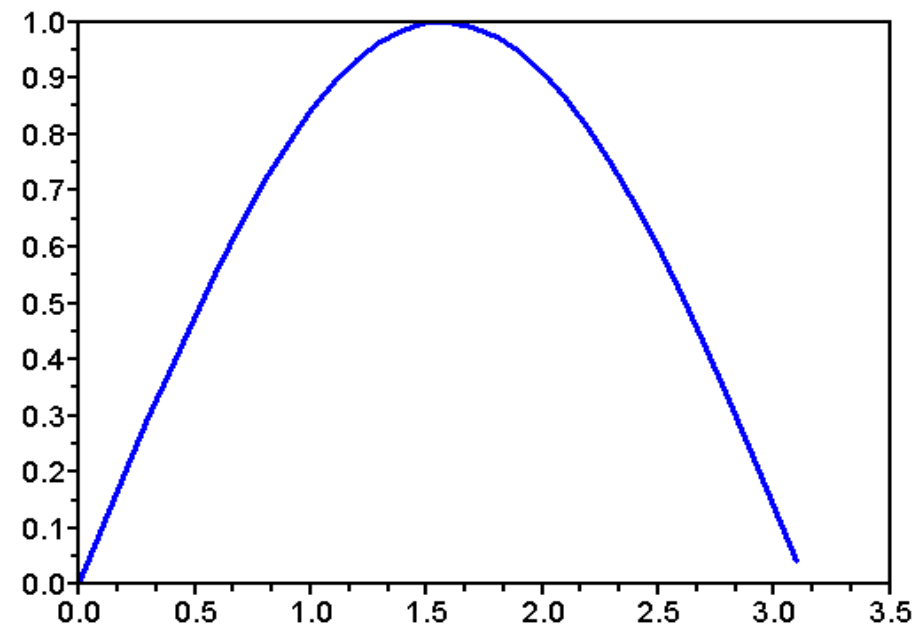
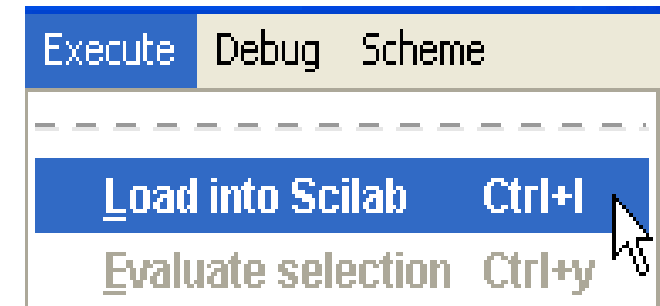
<code>y0</code>	vetor ou matriz real com condições iniciais
<code>t0</code>	valor do tempo inicial
<code>t</code>	vetor real com os instantes em que a solução é calculada
<code>f</code>	função, <code>list</code> ou <code>string</code> de caracteres
<code>tipo</code>	uma das seguintes opções: "adams", "stiff", "rk", "rkf", "fix", "discrete", "roots" Obs: <code>lsoda</code> é o pacote de integração <i>default</i>
<code>rtol,atol</code>	constantes ou vetor com tolerância relativa e absoluta de mesma dimensão de <code>y</code> . valores <i>default</i> : <code>rtol=1.e-5</code> e <code>atol=1.e-7</code> valores <i>default</i> para "rkf" e "fix": <code>rtol=1.e-3</code> e <code>atol=1.e-4</code>
<code>jac</code>	função, <code>list</code> ou <code>string</code> de caracteres
<code>w,iw</code>	vetores.
<code>ng</code>	inteiro.
<code>g</code>	função, <code>list</code> ou <code>string</code> de caracteres
<code>k0</code>	inteiro, tempo inicial
<code>kvect</code>	inteiro (vetor)

Exemplo de *ode()*

```
SciPad - f.sci
File Edit Search Execute Debug Scheme
Options Windows Help

// Programa para estudo de ode()
// UFU, 08/Out/2007
// pré-processamento
clc,clear,mode(-1),list(0);
// Função
function dydt=f(t,y),
//dy/dt=y^2-y sin(t)+cos(t), y(0)=0
dydt=y^2-y*sin(t)+cos(t);
endfunction
// Programa Principal
y0=0;t0=0; // C.I.
t=0:0.1:%pi;
y=ode(y0,t0,t,f);
plot(t,y)
disp('* FIM *')
// FIM

Line: 3 Column: 4 Logical line: 3
```



Exercícios



Otimização e Regressão com o Scilab...



Otimização e
Regressão...

Resolução de Problemas de Otimização/Regressão com o Scilab

Exercícios



Usando o Scicos...



Scicos...

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!!