

Introdução ao Scilab 3.0

Parte 4

Paulo S. Motta Pires
`pmotta@dca.ufrn.br`

Departamento de Engenharia de Computação e Automação
Universidade Federal do Rio Grande do Norte
NATAL - RN

Endereços e Créditos

Prof. [Paulo S. Motta](#) Pires

- e-mail: pmotta@dca.ufrn.br
- homepage : www.dca.ufrn.br/~pmotta

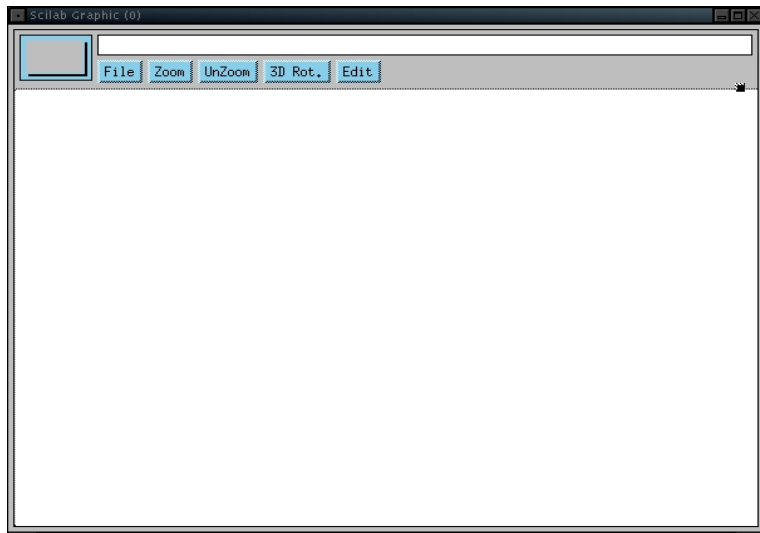
Este material pode ser copiado livremente, mantidos os créditos.

Agenda

- Parte 1
 - Introdução: Computação Numérica
 - O Ambiente Scilab
- Parte 2
 - Operações Básicas
 - Polinômios, Vetores, Matrizes e Listas
- Parte 3
 - Programação
- Parte 4
 - Gráficos
 - Considerações Finais

Agenda Parte 4 - Detalhes

- Janela Gráfica - Opções
- Gráficos Bi-dimensionais
- Gráficos Tri-dimensionais
- Conclusões



Comando plot2d([x], y)

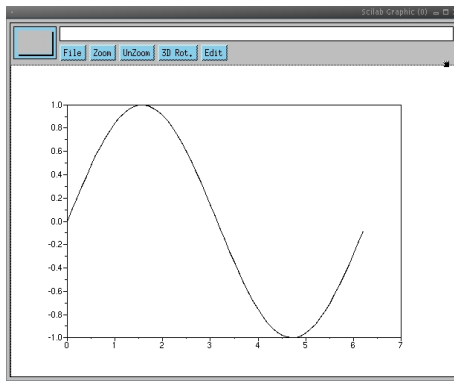
1 - Se **x** e **y** são vetores, a função **plot2d(x,y)** permite traçar o gráfico de **y** em função de **x**. É importante observar que os dois vetores devem ter o mesmo número de elementos

```
-->// 0 vetor das abscissas, x
-->x = [0:0.1:2*pi]; // Intervalo [0, 2pi]
-->// 0 vetor das ordenadas, y
-->y = sin(x);
--> // Vetores devem ter a mesma dimensao

-->size(x)
ans =
!   1.   63. !

-->size(y)
ans =
!   1.   63. !

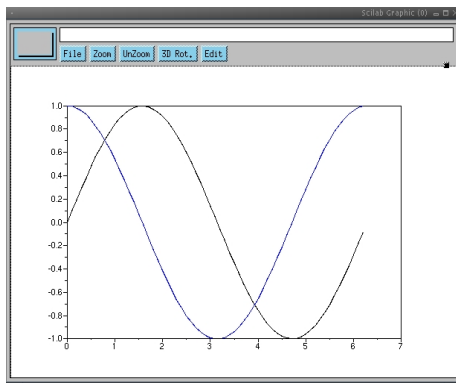
-->plot2d(x,y)
```



Comando plot2d([x], y)

2 - Se x é um vetor e y é uma matriz, a função **plot2d(x,y)** permite traçar o gráfico de cada coluna da matriz y em função do vetor x . Neste caso, o número de elementos das colunas da matriz y deve ser igual ao número de elementos do vetor x

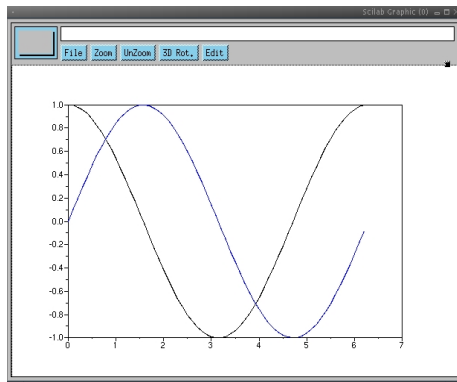
```
-->Y = [sin(x)' cos(x)']; // matriz Y  
--> // Y possui 63 elementos em cada coluna  
-->size(Y)  
ans =  
! 63. 2. !  
-->plot2d(x,Y)
```



Comando plot2d([x], y)

3 - Se **x** e **y** são matrizes, a função **plot2d(x,y)** permite traçar o gráfico de cada coluna da matriz **y** em função de cada coluna da matriz **x**. Neste caso, as matrizes devem ter as mesmas dimensões

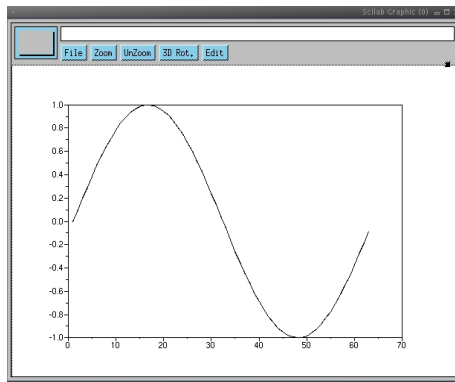
```
-->t = [0:0.1:2*%pi]; // Variavel auxiliar
-->X = [t' t'];        // Criando a matriz X
-->size(X) // X: 63 elementos em cada coluna
ans =
!   63.    2. !
-->Y = [cos(t)' sin(t)'];
-->size(Y) // Y: 63 elementos em cada coluna
ans =
!   63.    2. !
-->plot2d(X,Y)
```



Comando `plot2d([x], y)`

4 - Se **y** é um vetor, a função **plot2d(y)** permite traçar o gráfico do vetor **y** em função do vetor **[1:size(y)]**

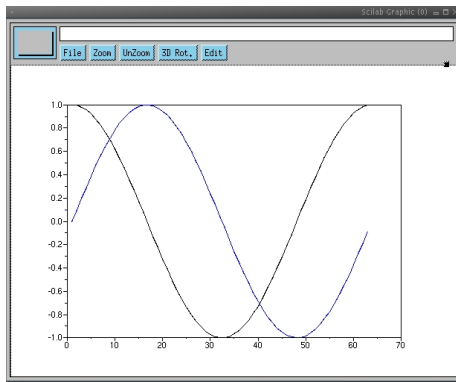
```
-->plot2d(sin(x))
```



Comando plot2d([x], y)

5 - Se **y** é uma matriz, a função **plot2d(y)** permite traçar o gráfico de cada coluna da matriz **y** em função do vetor **[1:size(y)]**

-->plot2d(Y)



Script

```
// Script para gerar a Figura 2 do capitulo 6
// Graficos - Scilab 3.0
```

```
// Definindo o vetor das abcissas, x
```

```
x = [0:0.1:2*%pi];
```

```
// Item 1 - y vetor
```

```
y = sin(x);
```

```
// xsetech[abcissa, ordenada, largura, altura]
// do grafico
```

```
xsetech([0, 0, 0.3, 0.5]);
xtitle("Item 1"); plot2d(x,y)
```

```
// Item 2 - y matriz
```

```
Y = [sin(x)' cos(x)']; // Definindo a matriz Y
```

```
xsetech([0.35, 0, 0.3, 0.5]);
xtitle("Item2"); plot2d(x,Y)
```

```
// Item 3 - x e y sao matrizes
// Definindo uma variavel auxiliar
```

```
t = [0:0.1:2*%pi];
```

```
// Criando a matriz X
```

```
X = [t' t'];
```

```
// Criando a matriz Y
```

```
Y = [cos(t)' sin(t)'];
```

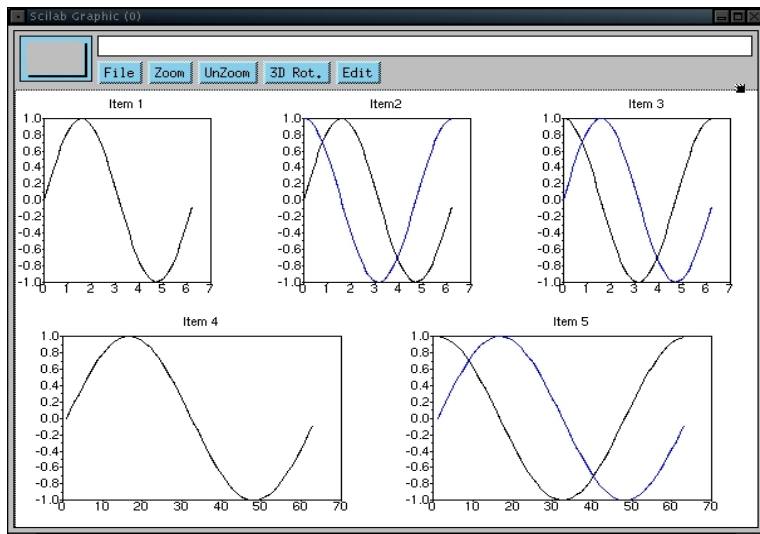
```
xsetech([0.70, 0, 0.3, 0.5]);
xtitle("Item 3"); plot2d(X,Y)
```

```
// Item 4 - y vetor
```

```
xsetech([0, 0.5, 0.5, 0.5]);
xtitle("Item 4"); plot2d(sin(x))
```

```
// Item 5 - Y matriz
```

```
xsetech([0.5, 0.5, 0.5, 0.5]);
xtitle("Item 5"); plot2d(Y)
```



```
plot2d([x],y,<opt_args>)
```

A forma geral para da função **plot2d()** inclui um terceiro argumento, **<opt_args>**,

```
plot2d([x],y,<opt_args>)
```

onde **<opt_args>** é uma seqüência de opções que determinam as características de um gráfico bi-dimensional

Parâmetros Adicionais

- **style** - é utilizada para especificar o padrão para a curva (ou curvas) que estão sendo traçadas. O valor associado à essa opção deve ser um vetor com valores inteiros positivos ou negativos. Se o valor associado for positivo, a curva é contínua. Nesse caso, o valor associado à opção define, também, a cor da curva que está sendo traçada. Se o valor associado à opção for negativo ou zero, a curva será desenhada usando marcadores.

Parâmetros Adicionais

- **style** - é utilizada para especificar o padrão para a curva (ou curvas) que estão sendo traçadas. O valor associado à essa opção deve ser um vetor com valores inteiros positivos ou negativos. Se o valor associado for positivo, a curva é contínua. Nesse caso, o valor associado à opção define, também, a cor da curva que está sendo traçada. Se o valor associado à opção for negativo ou zero, a curva será desenhada usando marcadores.
- **logflag** - define a escala, logarítmica ou linear, a ser utilizada nos eixos **x** e **y** do gráfico. Os valores associados à essa opção são *strings*, “nn”, “nl”, “ln” e “ll”, onde l indica a escala logarítmica, n indica escala normal e a segunda letra indica o tipo de graduação dos eixos (normal ou logarítmica). O valor *default* desta opção é “nn”, isto é, escala normal com graduação normal dos eixos;

Parâmetros Adicionais

- **style** - é utilizada para especificar o padrão para a curva (ou curvas) que estão sendo traçadas. O valor associado à essa opção deve ser um vetor com valores inteiros positivos ou negativos. Se o valor associado for positivo, a curva é contínua. Nesse caso, o valor associado à opção define, também, a cor da curva que está sendo traçada. Se o valor associado à opção for negativo ou zero, a curva será desenhada usando marcadores.
- **logflag** - define a escala, logarítmica ou linear, a ser utilizada nos eixos **x** e **y** do gráfico. Os valores associados à essa opção são *strings*, "nn", "nl", "ln" e "ll", onde l indica a escala logarítmica, n indica escala normal e a segunda letra indica o tipo de graduação dos eixos (normal ou logarítmica). O valor *default* desta opção é "nn", isto é, escala normal com graduação normal dos eixos;
- **rect** - é utilizada para estabelecer os limites do gráfico. O valor associado à essa opção é um vetor real com quatro entradas, [xmin, ymin, xmax, ymax], onde (xmin, xmax) e (ymin, ymax) indicam os valores mínimo e máximo para os eixos **x** e **y**, respectivamente;

Parâmetros Adicionais

- **frameflag** - É utilizada para controlar a escala dos eixos coordenados. O valor associado à essa opção é um número inteiro no intervalo 0 e 8, inclusive;

Parâmetros Adicionais

- **frameflag** - É utilizada para controlar a escala dos eixos coordenados. O valor associado à essa opção é um número inteiro no intervalo 0 e 8, inclusive;
- **axesflag** - especifica como os eixos serão traçados. O valor associado à essa opção é um número inteiro variando entre 0 e 5, inclusive;

Parâmetros Adicionais

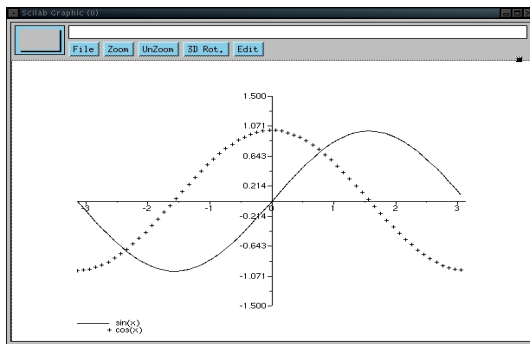
- **frameflag** - É utilizada para controlar a escala dos eixos coordenados. O valor associado à essa opção é um número inteiro no intervalo 0 e 8, inclusive;
- **axesflag** - especifica como os eixos serão traçados. O valor associado à essa opção é um número inteiro variando entre 0 e 5, inclusive;
- **nax** - permite definir os nomes e as marcas nos eixos **x** e **y**. O valor associado à essa opção, válido apenas quando a opção **axesflag=1**, é um vetor com quatro entradas inteiras, [nx, Nx, ny, Ny]. O parâmetro, Nx, é o número de marcações principais (*tics*) utilizadas no eixo **x**; nx é o número de divisões (*subtics*) entre as marcações principais do eixo **x**; Ny e ny têm significados semelhantes, tratando-se do eixo **y**;

Parâmetros Adicionais

- **frameflag** - É utilizada para controlar a escala dos eixos coordenados. O valor associado à essa opção é um número inteiro no intervalo 0 e 8, inclusive;
- **axesflag** - especifica como os eixos serão traçados. O valor associado à essa opção é um número inteiro variando entre 0 e 5, inclusive;
- **nax** - permite definir os nomes e as marcas nos eixos **x** e **y**. O valor associado à essa opção, válido apenas quando a opção **axesflag=1**, é um vetor com quatro entradas inteiras, [nx, Nx, ny, Ny]. O parâmetro, Nx, é o número de marcações principais (*tics*) utilizadas no eixo **x**; nx é o número de divisões (*subtics*) entre as marcações principais do eixo **x**; Ny e ny têm significados semelhantes, tratando-se do eixo **y**;
- **leg** - permite definir as legendas das curvas. O valor associado à esse parâmetro é uma *string* de caracteres para cada gráfico traçado.

Exemplo

```
-->x = [-%pi:0.1:%pi];  
-->y = [sin(x)' cos(x)'];  
-->plot2d(x,y, style=[1, -1], rect=[-%pi, -1.5, %pi, 1.5],axesflag=5, ...  
-->leg = "sin(x)@cos(x)")  
-->
```



Variações

- O comando **plot2d()** apresenta algumas variações

Comando	Tipo de Gráfico
plot2d2()	gráficos 2-D linearizados
plot2d3()	gráficos 2-D com barras verticais
plot2d4()	gráficos 2-D com setas

A sub-opção Graphics da opção Demos apresenta exemplos de utilização da função `plot2d()` e de suas variações.

- É importante lembrar que o *demo* de uma função gráfica também pode ser ativado através da chamada da função. Por exemplo, para ativar o *demo* da função gráfica `histplot`, que plota um histograma, basta fazer:

```
-->histplot()
```

Outros Comandos

- `xgrid` - coloca uma grade em um gráfico bi-dimensional
- `xtitle` - coloca títulos em gráficos 2-D ou 3-D
- `titlepage` - coloca um título no meio de uma janela gráfica
- `subplot(m,n,p)` - permite dividir a janela gráfica do Scilab em uma matriz $m \times n$. Em cada um dos elementos da “matriz”, especificado por p , pode ser colocado um gráfico.

Usando subplot()

```
-->subplot(221)
```

```
-->champ      // chamada do demo da funcao champ
Demo of champ
champ(1:10,1:10,rand(10,10),rand(10,10),1.0);
```

```
-->subplot(222)
```

```
-->histplot  // chamada do demo da funcao histplot
histplot([-6:0.2:6],rand(1,2000,'n'),[1,-1], '011', ' ', [-6,0,6,0.5],[2,12,2,11]);
deff(' [y]=f(x)', 'y=exp(-x.*x/2)/sqrt(2*%pi);');
x=-6:0.1:6;x=x';plot2d(x,f(x),1,"000");
titre= 'macro histplot : Histogram plot';
xtitle(titre,'Classes','N(C)/Nmax');
```

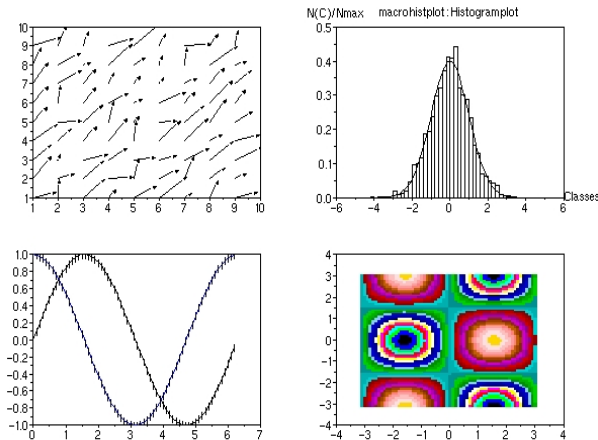
```
-->subplot(223)
```

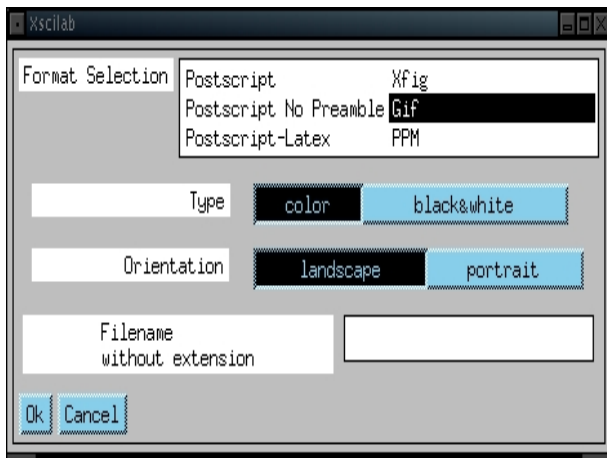
```
-->errbar    // chamada do demo da funcao errbar
x=0:0.1:2*%pi;
y=[sin(x);cos(x)]';x=[x;x]';
plot2d(x,y);
errbar(x,y,0.05*ones(x),0.03*ones(x));
```

```
-->subplot(224)
```

```
-->grayplot  // chamada do demo da funcao grayplot
Demo of grayplot
t=-%pi:0.1:%pi;m=sin(t)*cos(t);grayplot(t,t,m);
```

Observar que essa a Figura não foi capturada pelo **GIMP**. Ela foi armazenada em um arquivo através da sub-opção Export da opção **File** da janela gráfica na qual o gráfico foi gerado.





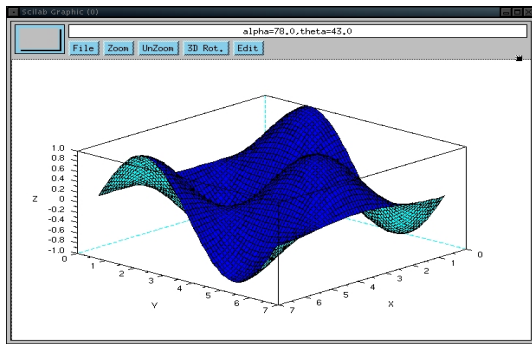
Gráficos 2D Especiais

- `bode` - permite traçar o gráfico de módulo e fase da resposta em frequência de um sistema linear;
- `gainplot` - permite traçar o gráfico do módulo da resposta em frequência de um sistema linear;
- `nyquist` - permite traçar o gráfico da parte imaginária versus parte real da resposta em frequência de um sistema linear;
- `m_cicle` - gráfico M-círculo usado com o gráfico de Nyquist;
- `chart` - permite traçar a diagrama de Nichols;
- `black` - permite traçar o diagrama de Black para um sistema linear;
- `evans` - permite traçar o o lugar das raízes pelo método de Evans;
- `plzr` - permite traçar o diagrama de polos e zeros.

O `help` do Scilab fornece informações mais detalhadas sobre a utilização dessas funções.

Gráficos 3D

```
-->x = [0:0.1:2*pi]';  
-->y = x;  
-->z = cos(x) * sin(y');  
-->plot3d(x, x, z)
```



3D Definidos por Funções

- `fplot3d` - que permite traçar gráficos de superfícies definidas por funções, como no *script* mostrado no exemplo:

```
deff('z=f(x,y)', 'z=x^4-y^4')  
x=-3:0.2:3 ;y=x ;  
clf() ;fplot3d(x,y,f,alpha=5,theta=31)
```

- `fplot3d1` - que permite traçar gráficos de superfícies definidas por funções, com o no caso anterior. As superfícies são apresentadas em escala cinza ou com uma graduação de cores.

Gráficos 3D Especiais

- `param3d` - permite traçar curvas paramétricas;
- `hist3d` - permite traçar histogramas 3-D;
- `contour` - permite traçar curvas de nível para uma função 3-D.

subplot()

```
-->subplot(121);
```

```
-->param3d
```

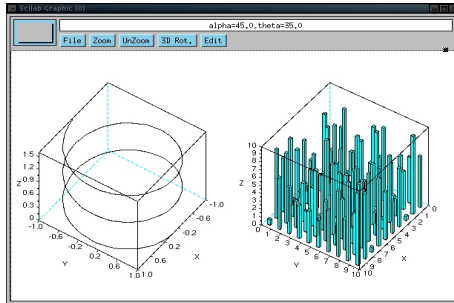
```
Demo of param3d
```

```
t=0:0.1:5*pi;param3d(sin(t),cos(t),t/10,35,45,'X@Y@Z',[2,4]);
```

```
-->subplot(122)
```

```
-->hist3d
```

```
hist3d(10*rand(10,10));
```



Scilab - Considerações Finais

- `www.scilab.org`

Contatos

Prof. Paulo Motta - DCA/UFRN

- `pmotta@dca.ufrn.br`
- `www.dca.ufrn.br/~pmotta`