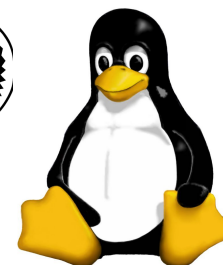


1º InstallFest do LEF

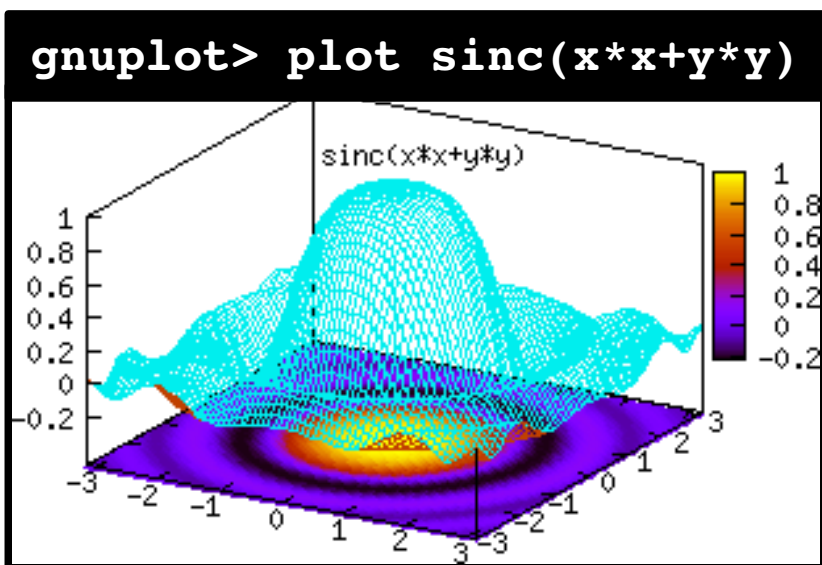
minicurso: gnuplot>



Dias 6 e 7 de novembro de 2007
16h às 18h Sala 206 – LEF – IFSC

Gerson J. Ferreira
gersonjr@ifsc.usp.br

gnuplot>
linguagem de script para gerar
gráficos em 2D e 3D pela linha
de comando;



Primeira parte:

- **Sobre o gnuplot**
 - x Pra que serve? Quando usar?
 - x Origin vs gnuplot – e outras alternativas
- **Comandos básicos**
 - x plot (2D) e splot (3D)
 - x funções analíticas
 - x arquivo de dados, barras de erro;
- **Personalizando o gráfico**
 - x cores, legendas, títulos, eixos, escalas, etc...
- **Usando scripts**
 - x para simplificar a edição de um gráfico
 - x diversos gráficos semelhantes

Segunda parte:

- **Multiplot**
 - x diversos gráficos numa janela
- **Salvando em arquivos**
 - x formatos: JPEG, GIF, PNG, EPS
- **Ajustando curvas a dados experimentais**
 - x funções de ajuste personalizáveis
- **Animações**
 - x acompanhamento dos resultados
 - x simulações dinâmicas

Apoio



UNIVERSIDADE
DE SÃO PAULO

Instituto de Física de São Carlos



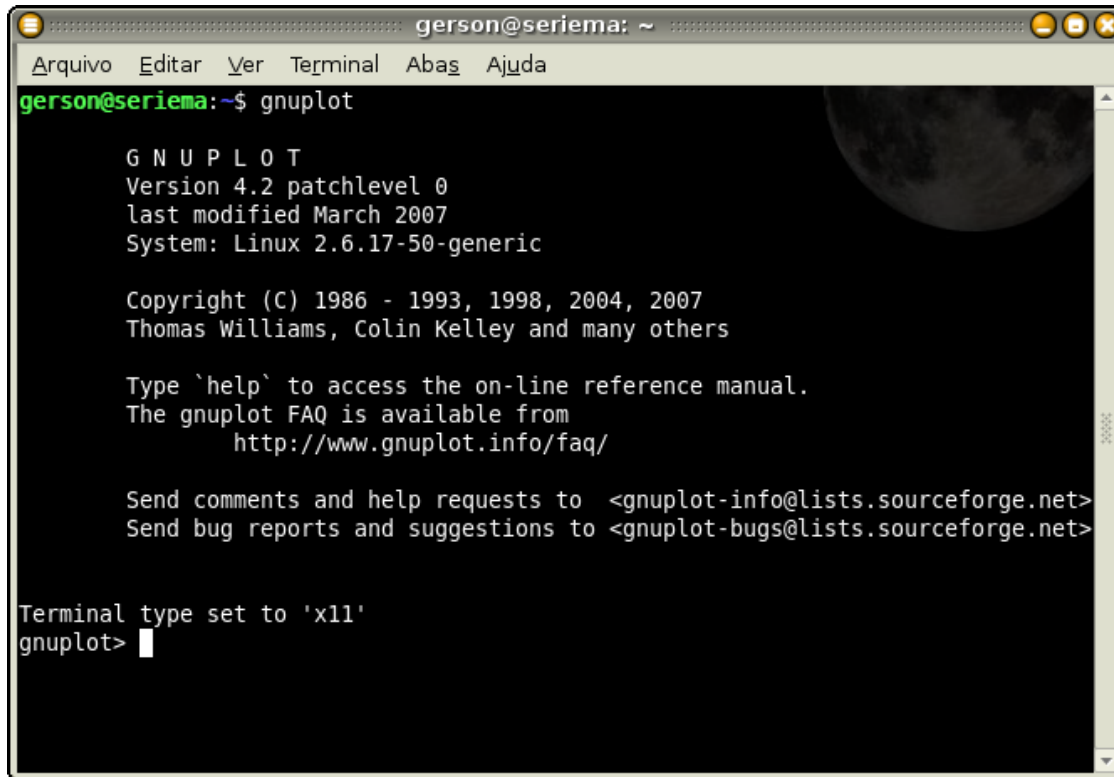
LINUX NEW MEDIA
The Pulse of Open Source



<http://www.lef.ifsc.usp.br/installfest>

Sobre o gnuplot e o Origin

gnuplot não é exatamente um substituto do Origin (*opinião pessoal*);



```
gereson@seriema: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
gereson@seriema:~$ gnuplot

G N U P L O T
Version 4.2 patchlevel 0
last modified March 2007
System: Linux 2.6.17-50-generic

Copyright (C) 1986 - 1993, 1998, 2004, 2007
Thomas Williams, Colin Kelley and many others

Type `help` to access the on-line reference manual.
The gnuplot FAQ is available from
    http://www.gnuplot.info/faq/

Send comments and help requests to <gnuplot-info@lists.sourceforge.net>
Send bug reports and suggestions to <gnuplot-bugs@lists.sourceforge.net>

Terminal type set to 'x11'
gnuplot>
```

E quanto ao Origin?

- WYSWYG, gerenciado pelo mouse, cansativo se repetitivo;
- figuras preparadas individualmente;
- não dá para acoplar a simulações

Vantagens do gnuplot

- é linguagem de script para gráficos
- elimina o uso do mouse (90%)
- formatação via scripts
 - x vários gráficos com a mesma formatação;
- facilmente integrável a simulações
 - x acompanhamento de resultados
 - x animações;

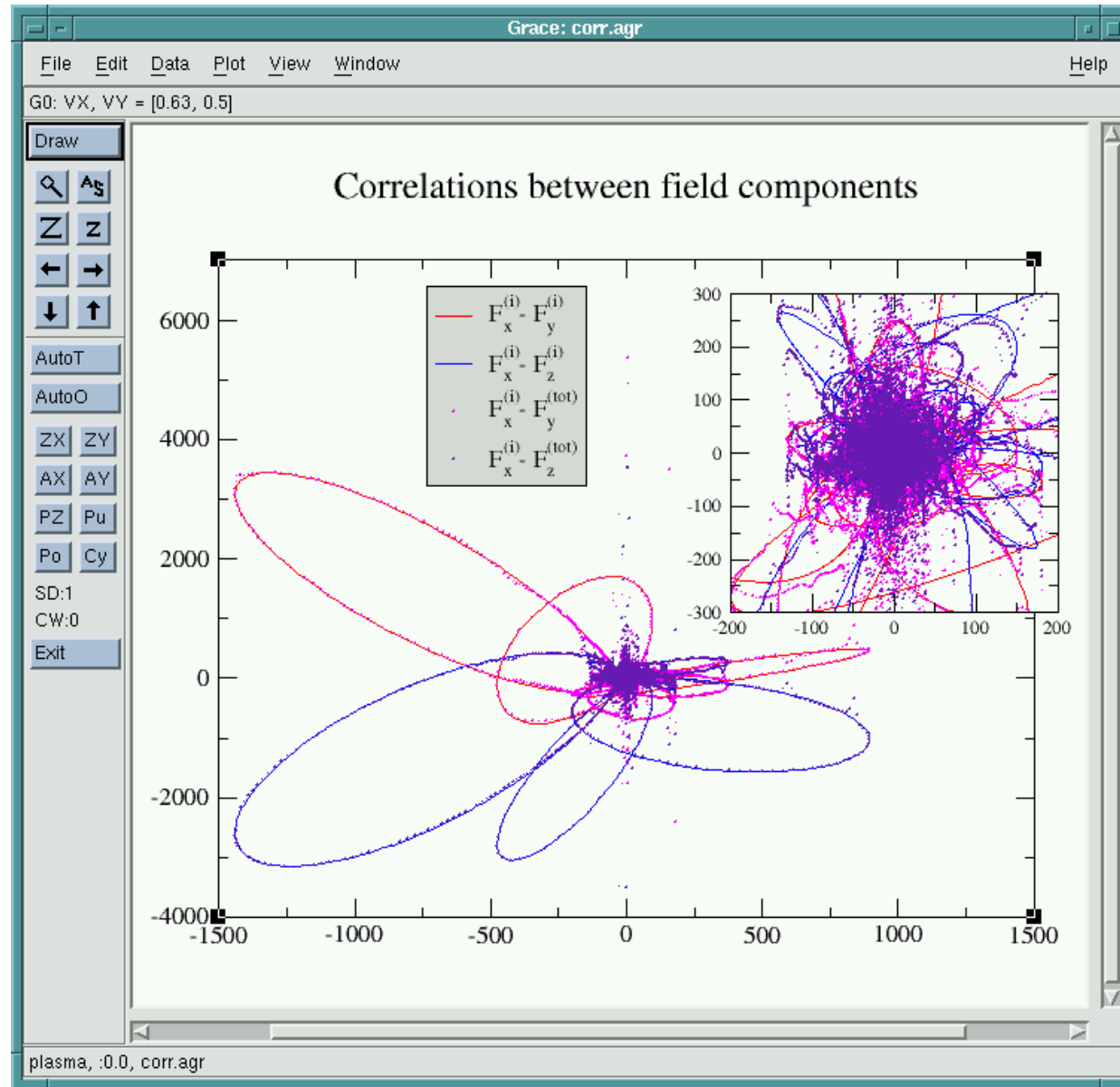
Desvantagens do gnuplot

- figuras complexas são difíceis de formatar;
- recursos avançados como ajuste de múltiplos picos não são automatizados (requer script extra);
- faltam recursos de planilha eletrônica;

Outras alternativas

Grace

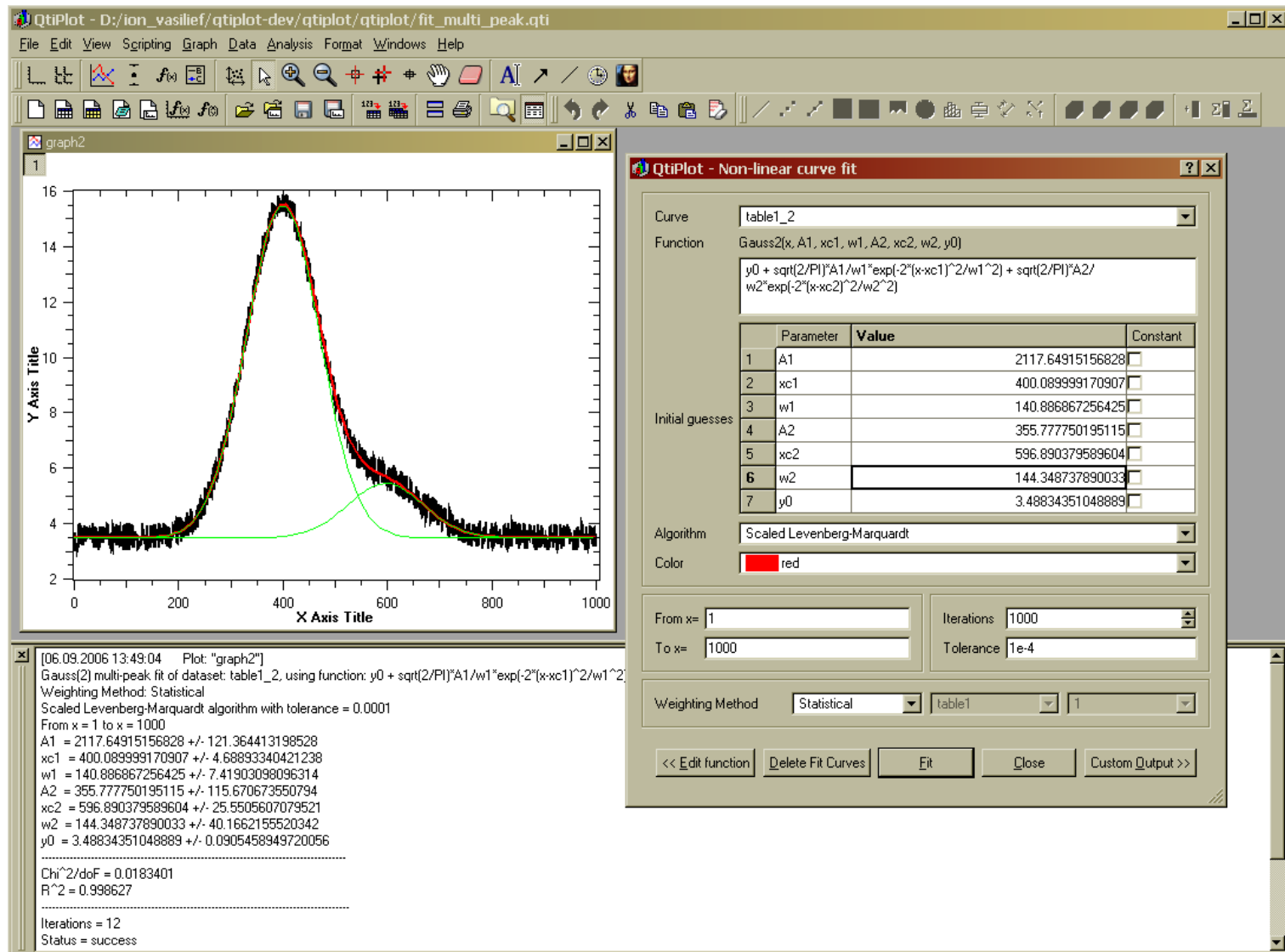
<http://plasma-gate.weizmann.ac.il/Grace/>



Outras alternativas

qtiPlot

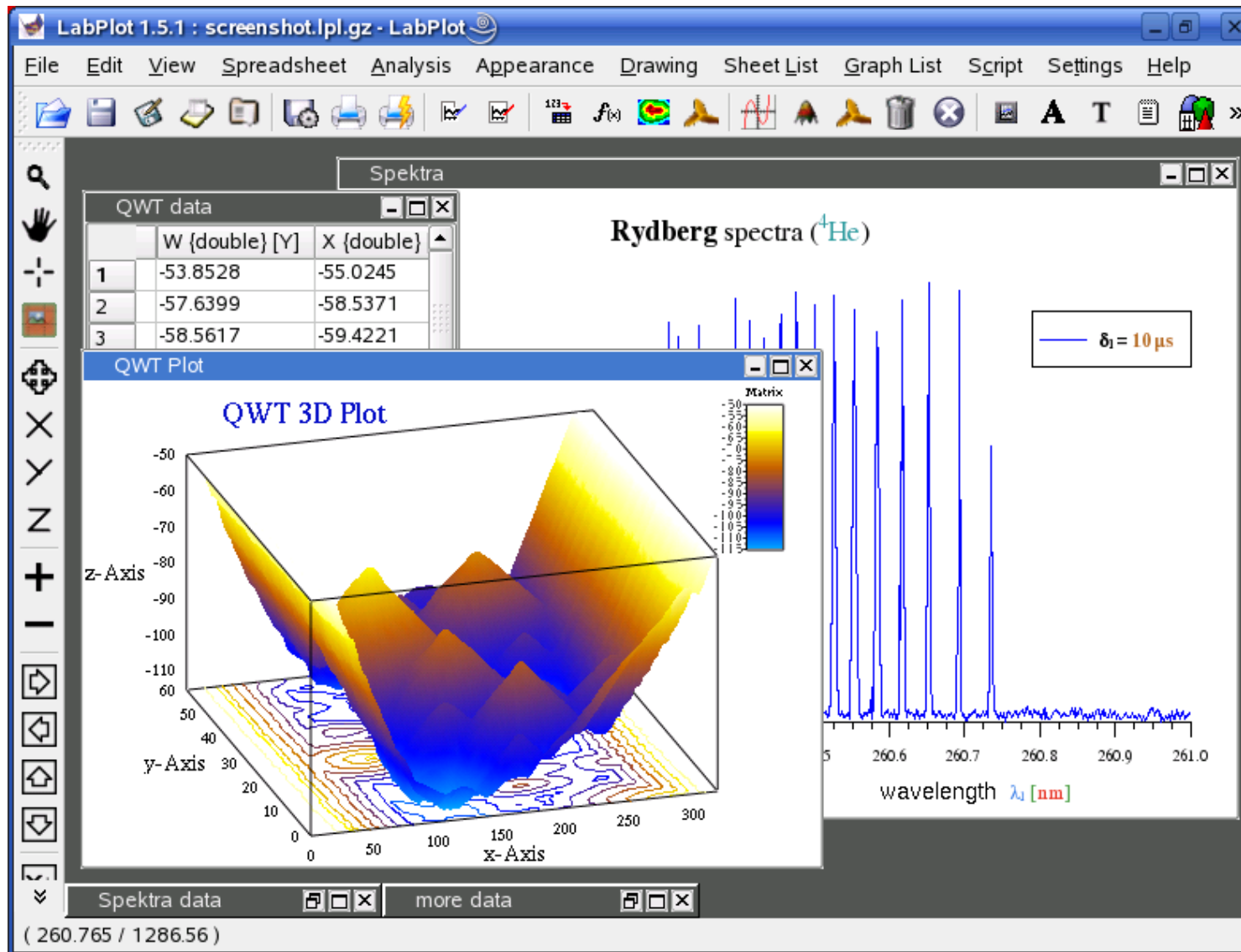
<http://soft.proindependent.com/qtiplot.html>



Outras alternativas

labPlot

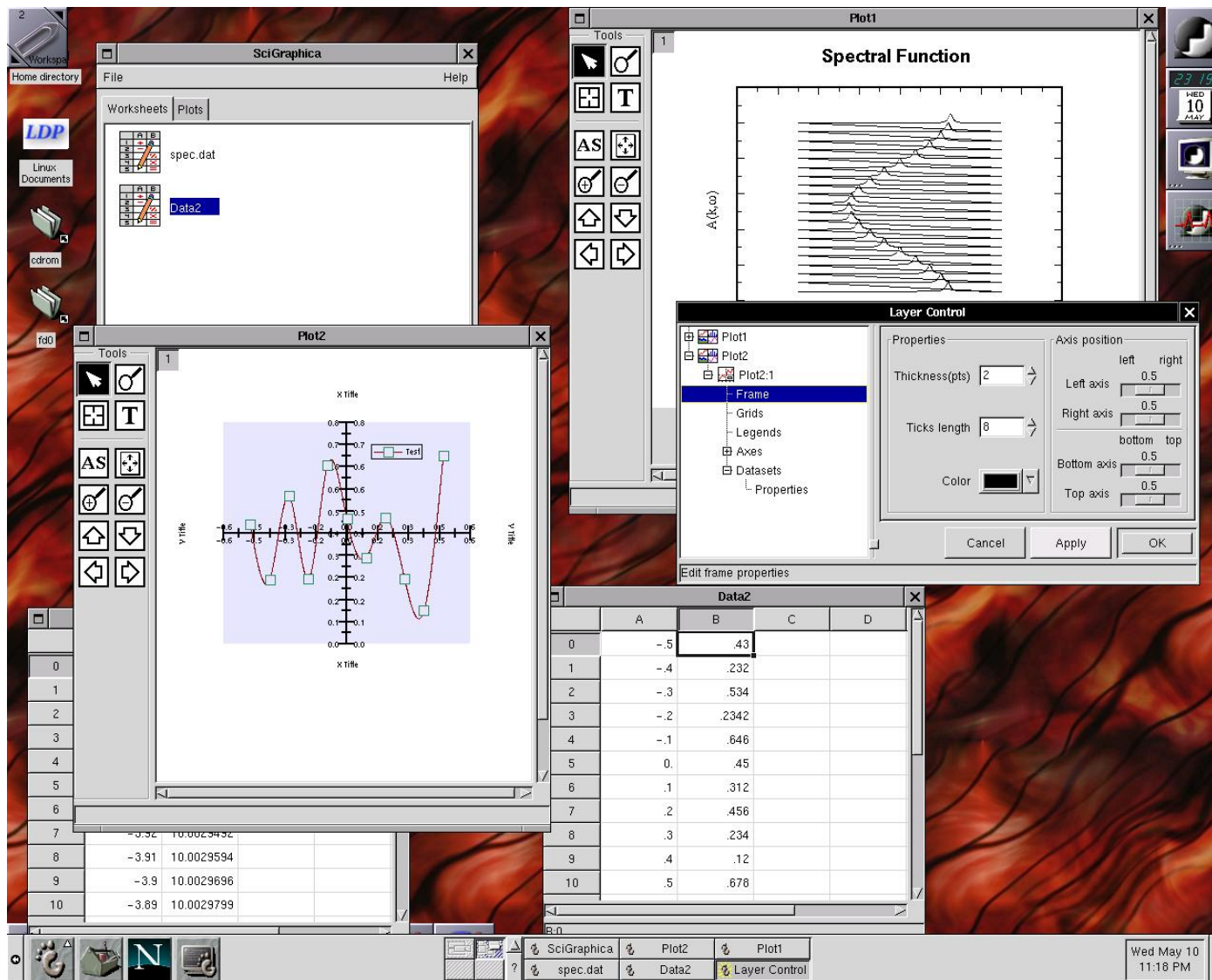
<http://labplot.sourceforge.net/>



Outras alternativas

SciGraphica

scigraphica.sourceforge.net/



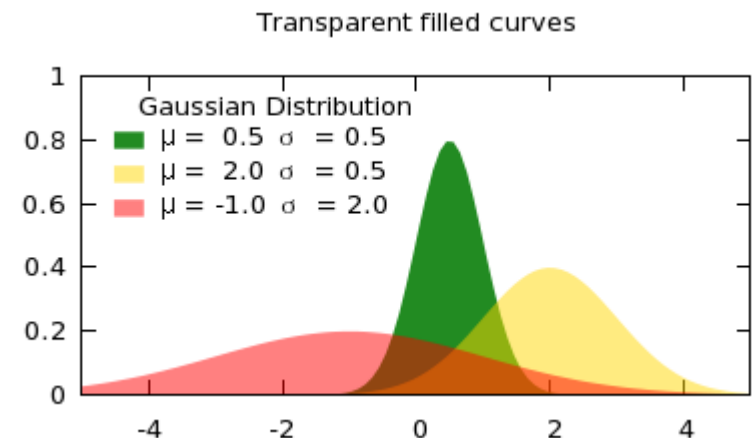
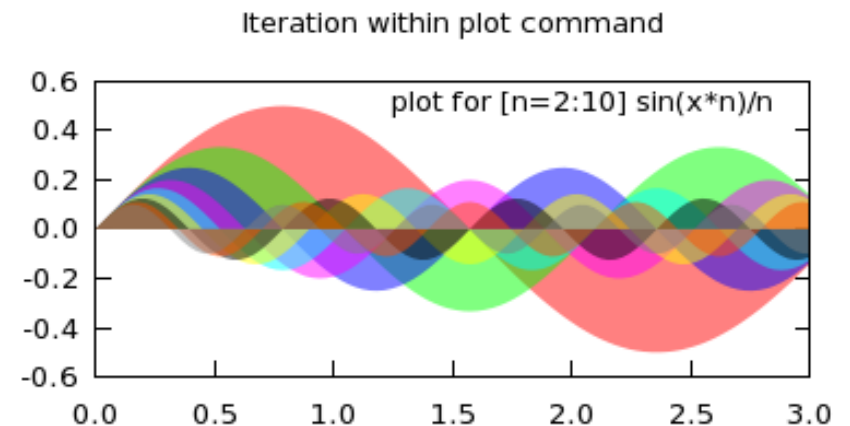
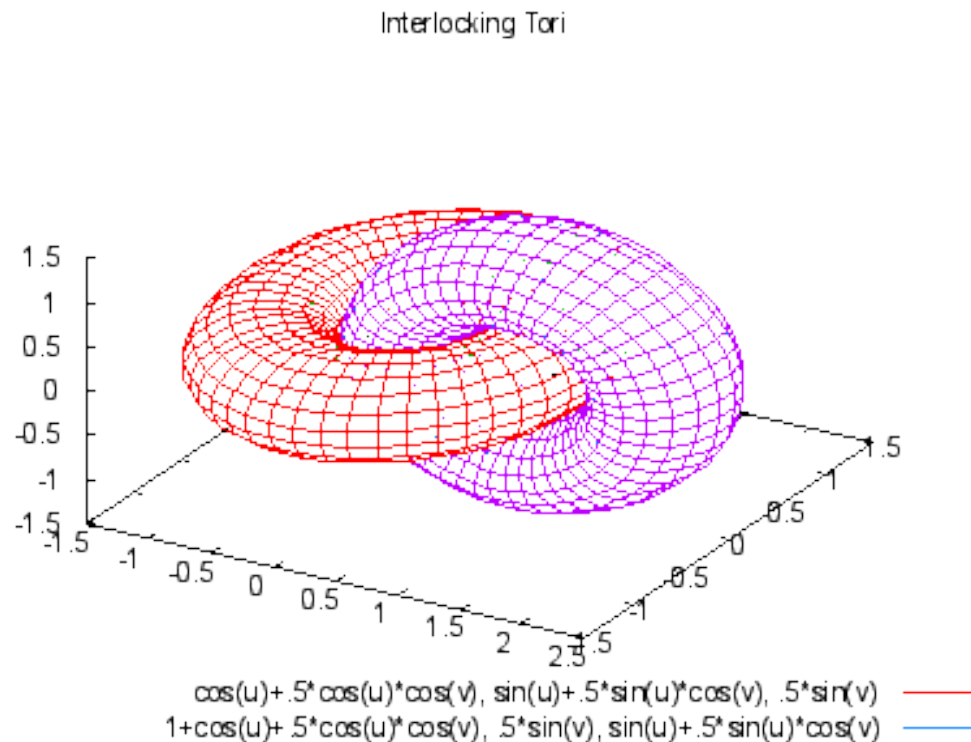
Visitem estes endereços...

Página oficial:

<http://www.gnuplot.info>

Tutorial gnuplot com comentários:

<http://t16web.lanl.gov/Kawano/gnuplot/index-e.html>



Comandos básicos - plot (2D)

Syntax:

```
plot {<ranges>}  
      {<function> | {"<datafile>" {datafile-modifiers}}}  
      {axes <axes>} {<title-spec>} {with <style>}  
      {, {definitions,} <function> ...}
```

opções

range: intervalo dos eixos

datafile: arquivo de dados

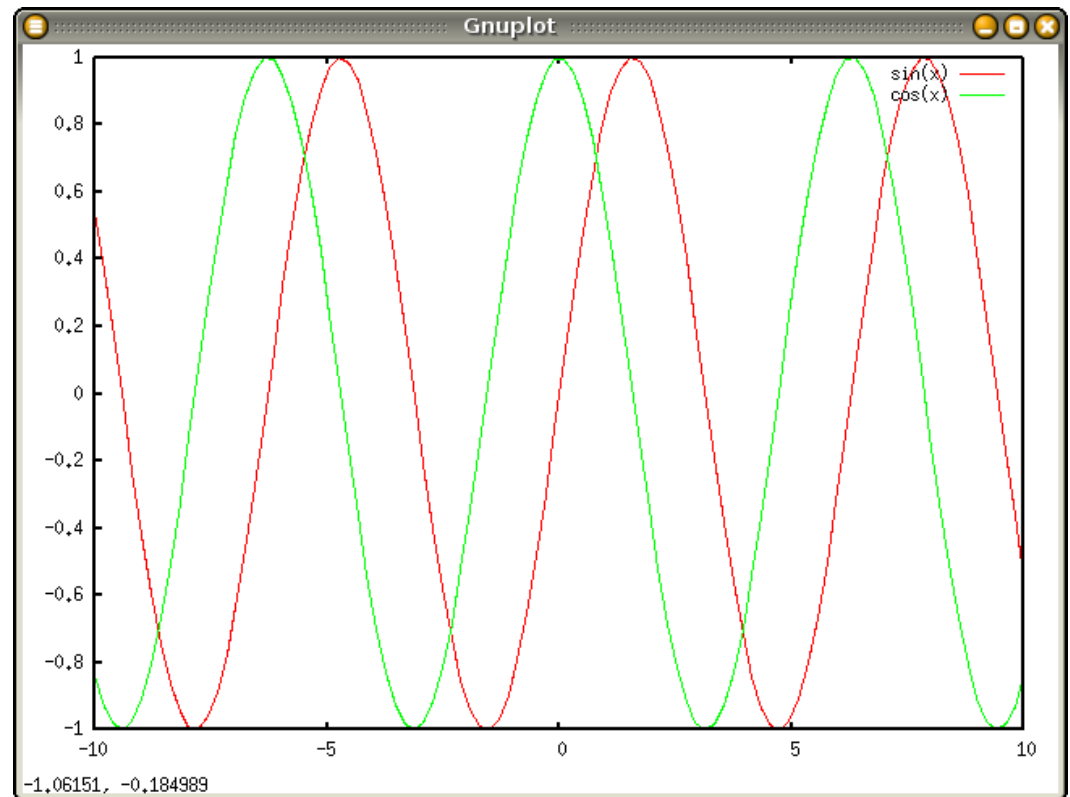
axes: eixos (x1, x2, y1, y2)

title-spec: legenda

with <style>: personalização

definitions: muitas opções...

plot sin(x), cos(x)



Comandos básicos - splot (3d)

Syntax:

```
splot {<ranges>}  
      <function> | "<datafile>" {datafile-modifiers}  
      {<title-spec>} {with <style>}  
      {, {definitions,} <function> ...}
```

opções

range: intervalo dos eixos

datafile: arquivo de dados

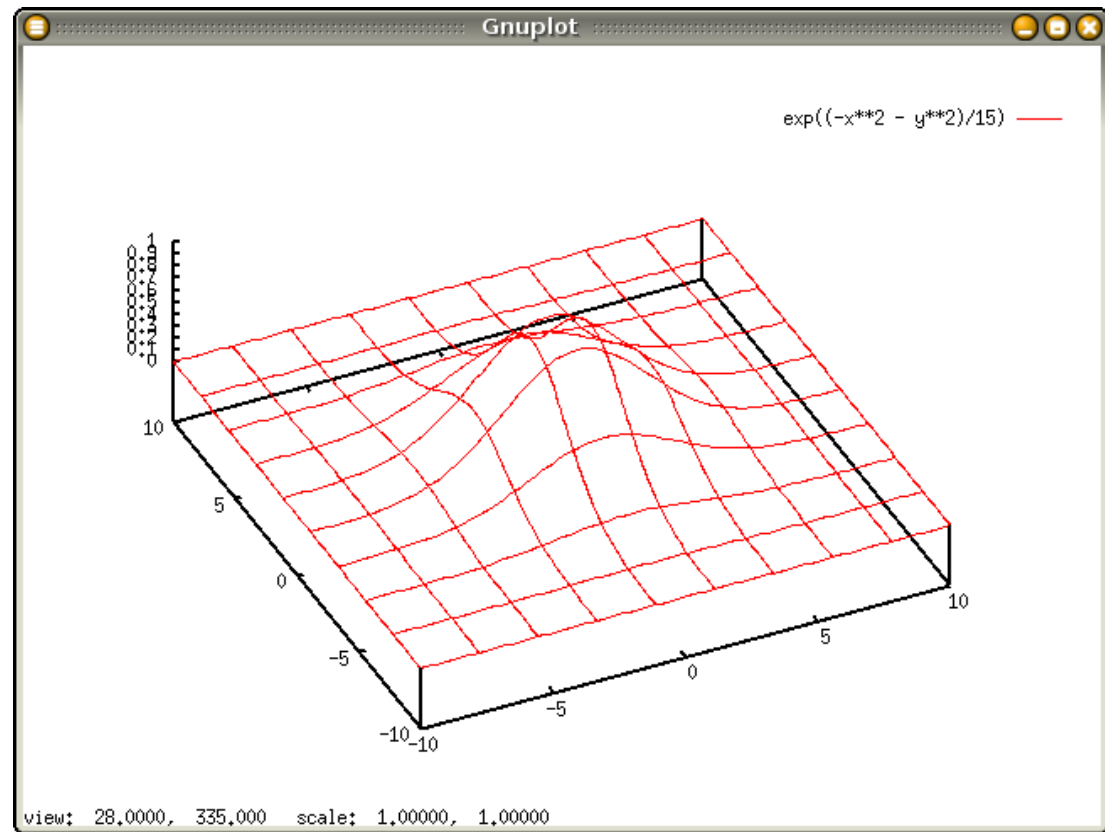
axes: eixos (x1, x2, y1, y2)

title-spec: legenda

with <style>: personalização

definitions: muitas opções...

splot exp((-x2-y**2)/15)**



Funções matemáticas

principais

abs(x)	módulo
acos(x)	arco-cosseno
asin(x)	arco-seno
atan(x)	arco-tangente
cos(x)	cosseno
cosh(x)	cosseno hiperbólico
erf(x)	função erro
exp(x)	exponencial
log(x)	log na base e
log10(x)	log na base 10
sgn(x)	sinal de x
sin(x)	seno
sinh(x)	seno hiperbólico
sqrt(x)	raiz quadrada
tan(x)	tangente
tanh(x)	tangente hiperbólica

help expressions functions

abs	acos	acosh	arg
asin	asinh	atan	atan2
atanh	besj0	besj1	besy0
besy1	ceil	column	cos
cosh	defined	erf	erfc
exists	exp	floor	gamma
gprintf	ibeta	igamma	imag
int	inverf	invnorm	lambertw
lgamma	log	log10	norm
rand	real	sgn	sin
sinh	sprintf	sqrt	stringcolumn
strlen	strstrt	substr	system
tan	tanh	timecolumn	valid
word	words		

**exponenciação
(Fortran-like)**

$$x^2 = x ** 2$$

Para definir uma função e parâmetros:

```
f(x,w) = sin(w*x)**2;  
plot f(x,1), f(x,3)
```

a parametrização será usada para ajustar curvas usando o comando fit (help fit)

Aprendendo com exemplos

```
plot {<ranges>}  
      {<function> | {"<datafile>" {datafile-modifiers}}}  
      {axes <axes>} {<title-spec>} {with <style>}  
      {, {definitions,} <function> ...}
```

Introdução ao plot (2D)

comandos para introduzir: eixos, legendas, título, cores, etc...

```
> reset;                                     exemplo01.gnuplot  
> f(x) = sin(x)/x  
> plot f(x)  
> set xrange [-15:15]  
> set yrange [-0.4:1]; replot  
> plot f(x) title "funcao sinc(x)" lt 3 lw 4  
> set xlabel "eixo x"  
> set ylabel "eixo y"; replot  
> set key bottom left; replot  
> replot 0 title "" lt 1 lw 2;  
> set title "Meu primeiro grafico"; replot  
> set arrow 1 from -8,0.4 to -3.1,0.01 head; replot  
> set label 1 "zero da funcao" at -9,0.4 right; replot
```

consulte também...

```
> set key x,y  
> set [no]log (x|y)  
> set autoscale (x|y)  
> test
```

detalhes...

```
lt = line type  
lw = line width  
title = "" : não aparece na legenda
```

Aprendendo com exemplos

Introdução ao plot (2D)

```
plot {<ranges>}  
    {<function> | {"<datafile>" {datafile-modifiers}}}  
    {axes <axes>} {<title-spec>} {with <style>}  
    {, {definitions,} <function> ...}
```

gráficos com pontos e outros detalhes...

```
> reset  
> set xrange [1:2.2]; set yrange [-30:30]  
> set xlabel "eixo x"; set ylabel "eixo y"  
> set title "Using samples and functions"  
> f(x) = x**5  
> pi = 3.14159; sf = 4.5  
> plot (sf**x)*sin(f(x)*pi) notitle with linespoints  
> set samples 1000  
> set xtics ("Start" 1, "Middle" 1.6, "End" 2.2)  
> set ytics ("Oh crap!" -30, "Uh-oh" -15, "0" 0, "Uh-oh" 15, "Oh crap!" 30)  
> replot 15 notitle; replot -15 notitle
```

exemplo02.gnuplot

consulte também...

```
> set style function linespoints  
> help plotting styles
```

Aprendendo com exemplos

```
splot {<ranges>}  
      <function> | "<datafile>" {datafile-modifiers}}  
      {<title-spec>} {with <style>}  
      {, {definitions,} <function> ...}
```

Introdução ao splot (3D)

comandos para introduzir: eixos, legendas, título, cores, etc...

```
> sinc(t)=sin(t)/t                                     exemplo03.gnuplot  
> splot [-3:3][-3:3] sinc(x*x+y*y)  
> set isosamples 40,40; replot;  
> set hidden3d; replot;  
> set xlabel "eixo x"  
> set ylabel "eixo y"; replot  
> set pm3d at b; replot  
> unset hidden3d; replot  
> set pm3d at s; replot  
> set hidden3d; replot  
> set isosamples 10,10; replot  
> set isosamples 50,50; replot  
> set contour base; replot  
> set cntrparam levels 25; replot notitle  
> unset clabel; replot  
> set pm3d map; replot
```

Arquivos de dados

Formatos para arquivos... em colunas (plot 2d ou 3d)

Exemplos de comandos 2D:

exemplo04.gnuplot

```
> set style data points
> set pointsize 5
> plot "pt1.dat" u 1:2
> plot "pt1.dat" u 1:3
> plot "pt1.dat" u 2:1
> plot "pt1.dat" u 2:1:3 ps 0 with errorbars
> plot "pt1.dat" u 2:1:3 ps 0 w e, "" u 2:1 w l
> set style data lines
> plot "pt2.dat" u 1:2
```

Exemplos de comandos 3D:

```
> splot "pt2.dat" u 1:2:3
> splot "pt2.dat" w p lt 2, "" w l lt 1
```

Colunas:

- por padrão o plot usa as colunas 1 e 2 e o splot as colunas 1, 2 e 3;
- o número de colunas é livre, basta indicar quais você quer usar;

arquivo "pt1.dat"

coluna:	1	2	3
	1	1	0.1
	2	4	-0.1
	3	9	0.2
	4	16	0.1
	5	25	-0.1
	6	36	-0.2

arquivo "pt2.dat"

coluna:	1	2	3
	0	2	1
	1	1	2
	2	0	3
	1	-1	4
	0	-2	5
	-1	-1	6
	-2	0	7
	-1	1	8
	0	2	9

Arquivos de dados

Formatos para arquivos... em matriz (plot 3d)

```
                                exemplo05.gnuplot
Eixos indicam o elemento de matriz
> set style data linespoints
> splot "pt3.dat" matrix
> set xtics ("100" 0, "200" 1, "300" 2)
> set hidden3d; replot;
> set pm3d at s; replot;
> set contour base; replot
> set cntrparam levels 25; replot;
> unset clabel; replot;
```

Usando os mesmos dados no formato coluna

```
> splot "pt4.dat" u 1:2:3
```

- Neste plot 3d o formato coluna deve ser organizado de forma a ter o mesmo número de elementos em todos os blocos;
- **Uma única linha em branco (!!)** deve separar os blocos referentes a cada trecho do gráfico;

arquivo "pt3.dat"

0	1	4	9
1	2	5	10
4	5	8	13
9	10	13	18
16	17	20	25
25	26	29	34

arquivo
"pt4.dat"

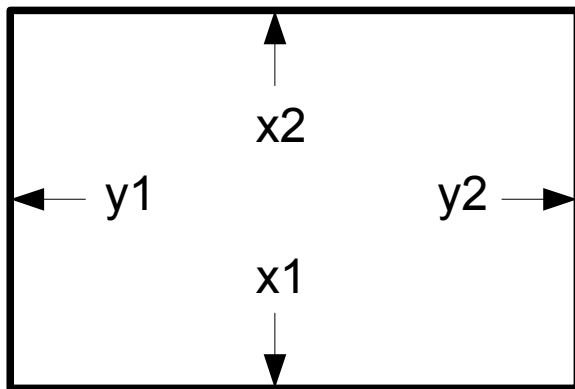
mesmo resultado
no formato
de coluna

0	0	0
0	1	1
0	2	4
0	3	9
0	4	16
0	5	25
1	0	1
1	1	2
1	2	5
1	3	10
1	4	17
1	5	26
2	0	4
2	1	5
2	2	8
2	3	13
2	4	20
2	5	29
3	0	9
3	1	10
3	2	13
3	3	18
3	4	25
3	5	34

Eixos secundários

usando mais de um eixo x (ou y):

```
> set y2tics  
> plot x lt 1 lw 2, x**2 lt 1 lw 2 axes x1y2  
> set arrow 1 from 0,0 to -5,0 head lt 1 lw 2 size 0.5,30 filled  
> set arrow 2 from 5,second 25 to 8,second 25 head lt 2 lw 2 size 0.5,30 filled  
> set key top center  
> replot
```



No comando plot : `> plot x axes x1y2`

Comandos que precisam de coordenadas:

`set arrow 1 from 5,second 25 to 8 second 25`

O padrão são os eixo x_1y_1 ;

Scripts

Simplificando a edição...

```
> cor=3                                script01.gnuplot
> tamanho=4

> set xrange [-15:15]
> set yrange [*:*]
> set xlabel "eixo x"
> set ylabel "eixo y"
> set title "funcao sinc(x) = sin(x)/x"
> f(x) = sin(x)/x

> set terminal jpeg
> set output "figural.jpg"

> plot f(x) title "sinc(x)"

> set terminal x11
> set output
```

para rodar o script:

```
$ gnuplot script01.gnuplot
    ou
> load "script01.gnuplot"
```

Scripts

Fazendo animações...

```
> max=2*pi
> f(x)=sin(x)
> dx = max/100
> set xrange[0:max]

> ix = ix + dx
> set object 1 rectangle at ix,f(ix) size 0.1,0.05;
> plot f(x)
> pause 0.1

> if (ix < max) reread
```

script02.gnuplot

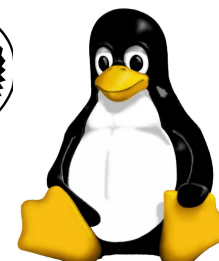
para rodar o script:

neste caso a variável "ix"
deve ser inicializada:

```
> ix = 0
> load "script02.gnuplot"
```

1º InstallFest do LEF

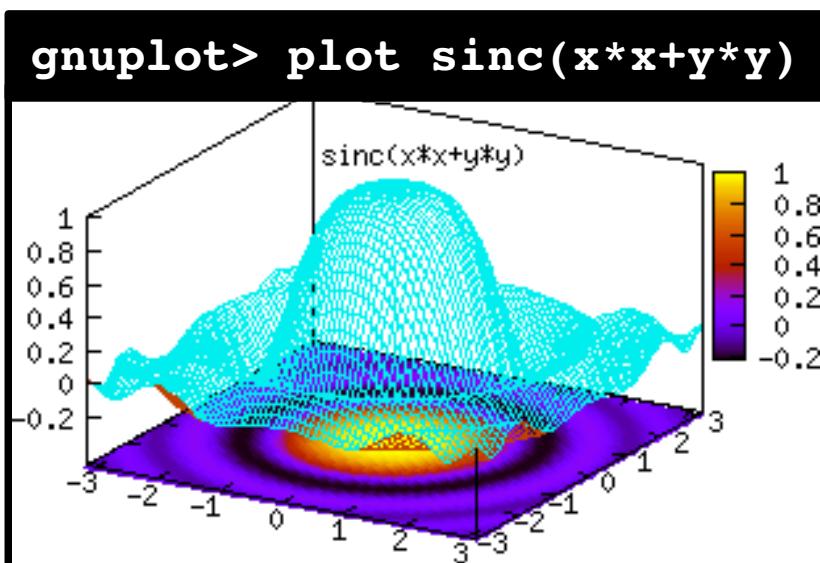
minicurso: gnuplot>



Dias 6 e 7 de novembro de 2007
16h às 18h Sala 206 – LEF – IFSC

Gerson J. Ferreira
gersonjr@ifsc.usp.br

gnuplot>
linguagem de script para gerar
gráficos em 2D e 3D pela linha
de comando;



Primeira parte:

- **Sobre o gnuplot**
 - x Pra que serve? Quando usar?
 - x Origin vs gnuplot – e outras alternativas
- **Comandos básicos**
 - x plot (2D) e splot (3D)
 - x funções analíticas
 - x arquivo de dados, barras de erro;
- **Personalizando o gráfico**
 - x cores, legendas, títulos, eixos, escalas, etc...
- **Usando scripts**
 - x para simplificar a edição de um gráfico
 - x diversos gráficos semelhantes

Segunda parte:

- **Multiplot**
 - x diversos gráficos numa janela
- **Salvando em arquivos**
 - x formatos: JPEG, GIF, PNG, EPS
- **Ajustando curvas a dados experimentais**
 - x funções de ajuste personalizáveis
- **Animações**
 - x acompanhamento dos resultados
 - x simulações dinâmicas

Apoio



UNIVERSIDADE
DE SÃO PAULO

Instituto de Física de São Carlos



LINUX NEW MEDIA
The Pulse of Open Source



<http://www.lef.ifsc.usp.br/installfest>

Multiplot

diversos quadros em uma figura...

Dois gráficos alinhados na vertical

```
> reset          exemplo06.gnuplot
> unset key

> set multiplot
> dx = 1.0/1
> dy = 1.0/2
> set size dx, dy
> f(t)=sin(t)/t

> set hidden3d
> unset colorbox
> set isosamples 80,80
> set view 40,45,1,1

> set origin 0*dx,0*dy
> set pm3d at s
> splot [-3:3][-3:3]f(x*x+y*y)

> set origin 0*dx,1*dy
> unset pm3d
> splot [-3:3][-3:3]f(x*x+y*y)

> unset multiplot
```

Dois plots (2d) ao lado de um splot (3d)

```
> reset          exemplo07.gnuplot
> unset key

> set multiplot
> dx = 1.0/2
> dy = 1.0/2
> set size dx, dy
> f(t)=t*sin(t)
> g(t)=exp(-t*t)
> set hidden3d

> set origin 0*dx,0*dy
> plot [-6:6] g(x)

> set origin 0*dx,1*dy
> plot [-10:10][-10:10] f(x)

> set origin 1*dx,0*dy
> set size dx, 2*dy
> set isosamples 100,100
> set pm3d at s
> unset colorbox
> set view 10,0,1,1
> splot [-10:10][-10:10] g(x-f(y))

> unset multiplot
```


Exportando os resultados

(x11, jpeg, png e eps)

outros formatos: help set terminal

```
> reset                                exemplo08.gnuplot

> f(x)=x*sin(x)
> set xrange [-20:20]
> set yrange [-20:20]
> set samples 300,300

> set terminal x11
> set output
> plot f(x)

> set output "figura08.jpg"
> set terminal jpeg size 800,600 crop
> plot f(x)

> set output "figura08.png"
> set terminal png transparent size 800,600 nocrop
> plot f(x)

> set output "figura08.eps"
> set terminal postscript eps color
> plot f(x)
```

versão 4.2

terminal padrão: **wxt**

*introduz uma interface
gráfica para alguns
comandos;*

animações: **terminal gif**

*eu prefiro fazer em formato
MPEG usando o mencoder
(mplayer) ou visualizar no
próprio terminal X11;*

Ajustando curvas (fitting) 2D e 3D

Ajuste a curva 2D

```
> reset                                exemplo09.gnuplot
> set key top left
> plot "pt5.dat" w p ps 3 lt 1 pt 6

> f(x) = fa*x + fb
> fit f(x) "pt5.dat" via fa,fb
> replot f(x) lt 2 lw 2 title "fit 01"

> g(x) = ga*x**2 + gb*x + gc
> fit g(x) "pt5.dat" via ga,gb,gc
> replot g(x) lt 3 lw 2 title "fit 02"
```

pt5.dat

0	1
1	0.6
2	3
3	10
4	14
5	25
6	34
7	47
8	68
9	85
10	90

final do "help fit"

**"Nonlinear fitting
is an art!"**

Ajuste a curva 3D

```
splot "pt6.dat" w p ps 3 pt 6          exemplo10.gnuplot

f(x,y)=a*x**2 + b*x*y + c*y**2 + d
fit f(x,y) "pt6.dat" u 1:2:3:(1) via a,b,c,d

set isosamples 30,30
replot f(x,y) w l lt 3

set isosamples 4,6; replot
```

pt6.dat

0	0	0
0	1	1
0	2	4
0	3	9
0	4	16
0	5	25
1	0	1
1	1	2
1	2	5
1	3	10
1	4	17
1	5	26
2	0	4
2	1	5
2	2	8
2	3	13
2	4	20
2	5	29
3	0	9
3	1	10
3	2	13
3	3	18
3	4	25
3	5	34

**não dá para fazer
fit 3D usando o
formato de matriz**

Animações: script no gnuplot

Via script gnuplot (reread): poucos recursos, não é a melhor opção

```
> max=2*pi
> f(x)=sin(x)
> dx = max/100
> set xrange[0:max]

> ix = ix + dx
> set object 1 rectangle at ix,f(ix) size 0.1,0.05;
> plot f(x)
> pause 0.1

> if (ix < max) reread
```

script02.gnuplot

para rodar o script:

neste caso a variável "ix"
deve ser inicializada:

```
> ix = 0
> load "script02.gnuplot"
```

**scripts BASH, PERL ou integração com
a saída de programas (C, Fortran) são
mais interessantes
(requer programação específica)**

- **reread**: roda novamente todo o script carregado;
- não há loops (while, for) implementados;
- variáveis devem ser inicializadas manualmente;

Animações: via pipes

Neste exemplo uso um script em BASH

```
#!/bin/bash

PI=3.1415926535897932384626433832795

MIN=0
MAX=`dc -e "10 k 2 $PI * p"``
DX=`dc -e "10 k $MAX $MIN - 100.0 / p"``

echo "reset;"
echo "f(x)=sin(x);"
echo "set xrange[$MIN:$MAX];"

x=$MIN
while [[ $x < $MAX ]]; do

    echo "set object 1 rectangle at $x,f($x) size 0.1,0.05;"
    echo "plot f(x)"
    echo "pause 0.1"

    x=`dc -e "10 k $x $DX + p"``
done
```

pipe_x11.sh
script BASH

Para executar:

```
$ ./pipe_x11.sh | gnuplot
```

neste exemplo uso o **terminal x11**,
mas podemos usar qualquer saída,
basta indicar no início do script.

Animações: formato GIF (p/ internet)

Formato GIF: script BASH adaptado (**mudanças em negrito**)

```
#!/bin/bash

PI=3.1415926535897932384626433832795

MIN=0
MAX=`dc -e "10 k 2 $PI * p"``
DX=`dc -e "10 k $MAX $MIN - 100.0 / p"``

echo "reset;"
echo "f(x)=sin(x);"
echo "set xrange[$MIN:$MAX];"
echo "set terminal gif animate delay 5 optimize crop"
echo "set output \"animacao.gif\""

x=$MIN
while [[ $x < $MAX ]]; do

    echo "set object 1 rectangle at $x,f($x) size 0.1,0.05;"
    echo "plot f(x)"
    echo "pause 0.1" : terminal gif controla o delay

    x=`dc -e "10 k $x $DX + p"``
done

echo "set output"
echo "set terminal x11"
```

pipe_gif.sh
script BASH

apenas
versão
4.2

Para executar:

\$./pipe_gif.sh | gnuplot

Animações: video MPEG

Formato MPEG: figuras codificadas via mplayer (mencoder)

```
#!/bin/bash

PI=3.1415926535897932384626433832795

MIN=0
MAX=`dc -e "10 k 2 $PI * p"``
DX=`dc -e "10 k $MAX $MIN - 100.0 / p"``

echo "reset;"
echo "f(x)=sin(x);"
echo "set xrange[$MIN:$MAX];"
echo "set terminal jpeg size 320,200 crop"

x=$MIN
i=10000
while [[ $x < $MAX ]]; do

    echo "set output \"mpeg-$i.jpg\""
    echo "set object 1 rectangle at $x,f($x) size 0.1,0.05;"
    echo "plot f(x)"

    x=`dc -e "10 k $x $DX + p"``
    i=`dc -e "$i 1 + p"``
done
```

```
pipe_mpeg.sh
script BASH
```

Para executar:

```
$ ./pipe_mpeg.sh | gnuplot
$ ./gera_video.sh
```

mencoder: codifica as imagens em formato de vídeo (pacote MPlayer)

```
$ mencoder mf://mpeg*.jpg -mf w=320:h=200:fps=25:type=jpg -ovc lavc
-lavcopts vcodec=mpeg4:mbd=2:trell -oac copy -o animacao.mpeg
```


Interface via linguagem C

```
#include <stdio.h>
#include <unistd.h>

#define gplot_path "/usr/local/bin/gnuplot"

int main(void) {
    FILE *gp;
    double xmin, xmax, dx, x;

    gp = popen(gplot_path, "w");

    xmin = 0;
    xmax = 2.0 * 3.141592;
    dx = (xmax-xmin)/100.0;

    fprintf(gp, "reset\n");
    fprintf(gp, "f(x)=sin(x)\n");
    fprintf(gp, "set xrange [%g:%g]\n", xmin, xmax);

    fflush(gp);
    for (x=xmin; x <= xmax; x+=dx) {

        fprintf(gp, "set object 1 rectangle at %f,f(%f) size 0.1,0.05\n", x,x);
        fprintf(gp, "plot f(x)\n");
        fflush(gp);

        usleep(50000);
    }

    pclose(gp);

    return(0);
}
```

iface.c

*exemplo simples de integração do gnuplot
com linguagem C:*

comando popen(...):

executa o comando indicado criando um
stream de saída integrado ao novo
comando via pipe

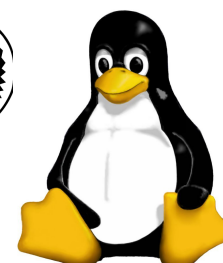
Para executar:

\$ gcc iface.c -o iface.out

\$./iface.out

1º InstallFest do LEF

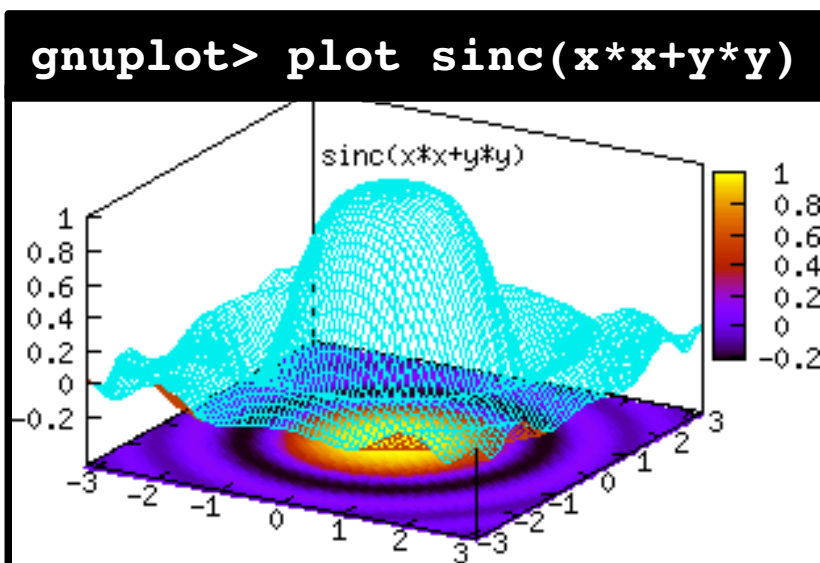
minicurso: gnuplot>



Dias 6 e 7 de novembro de 2007
16h às 18h Sala 206 – LEF – IFSC

Gerson J. Ferreira
gersonjr@ifsc.usp.br

gnuplot>
linguagem de script para gerar
gráficos em 2D e 3D pela linha
de comando;



Primeira parte:

- **Sobre o gnuplot**
 - x Pra que serve? Quando usar?
 - x Origin vs gnuplot – e outras alternativas
- **Comandos básicos**
 - x plot (2D) e splot (3D)
 - x funções analíticas
 - x arquivo de dados, barras de erro;
- **Personalizando o gráfico**
 - x cores, legendas, títulos, eixos, escalas, etc...
- **Usando scripts**
 - x para simplificar a edição de um gráfico
 - x diversos gráficos semelhantes

Segunda parte:

- **Multiplot**
 - x diversos gráficos numa janela
- **Salvando em arquivos**
 - x formatos: JPEG, GIF, PNG, EPS
- **Ajustando curvas a dados experimentais**
 - x funções de ajuste personalizáveis
- **Animações**
 - x acompanhamento dos resultados
 - x simulações dinâmicas

Apoio



UNIVERSIDADE
DE SÃO PAULO

Instituto de Física de São Carlos



LINUX NEW MEDIA
The Pulse of Open Source



<http://www.lef.ifsc.usp.br/installfest>

Resumo dos principais comandos

Comandos estudados neste curso introdutório:

- **fit.....**: para ajustar funções parametrizadas a dados de um arquivo. Pode ser usado para ajustar funções 2D e 3D, mas no caso 3D os dados não podem estar no formato de matriz;
- **help.....**: sistema de ajuda bastante completo, mas deve ser usado apenas como consulta de sintaxe – não é fácil achar os comandos sem saber o que esta procurando;
- **pause....**: interrompe a execução pelo tempo indicado e continua na sequência – usado em animações;
- **plot.....**: comando básico para fazer gráficos 2D;
- **replot...**: faz um novo **plot** sem apagar o anterior;
- **reset....**: limpa todas as definições modificadas pelo usuário/script;
- **set.....**: define características de outros comandos (**mais detalhes adiante**);
- **splot....**: comando básico para fazer gráficos 3D;
- **test.....**: mostra uma janela com exemplos de cores/comandos que podem ser usados (bom para escolher a cor do gráfico e forma dos símbolos);
- **unset....**: desfaz os ajustes do comando **set**;

Use o comando **help** para ver mais detalhes de cada um dos comandos acima (e.g., "> **help set**").

Resumo dos principais comandos

Detalhes do comando: set

- **set arrow:** para definir setas;
- **set autoscale:** define escala automática para as variáveis x , y e z ; cada dimensão pode ser ajustada individualmente;
- **set clabel:** define o formato da legenda em gráficos 3D com contornos (**set contour**);
- **set cntrparam:** controle dos detalhes dos contornos em gráficos 3D (**splot**);
- **set colorbox:** define posição, tamanho, etc da barra da escala de cores;
- **set contour:** ativa e define a exibição de contornos (**splot**); ver **cntrparam**;
- **set hidden3d:** esconde o que esta atrás da superfície 3D (**splot**);
- **set isosamples:** número de linhas no gráfico 3D (**grade**);
- **set key:** controle da legenda: tipo, tamanho, cor, etc;
- **set label:** para escrever textos no gráfico;
- **set log:** escala logaritmica;
- **set multiplot:** inicia o modo de vários gráficos (lado a lado, inset, etc);
- **set object:** exhibe um objeto no gráfico 2D (**plot**) – apenas na versão 4.2, por enquanto apenas o retângulo esta implementado;
- **set origin:** (**multiplot**) indica a origem do gráfico no modo **multiplot**;

Resumo dos principais comandos

Detalhes do comando: set

- **set output:** define o arquivo para saída de dados (deixar em branco se for usar o terminal X11 ou WXT);
- **set pm3d:** ativa modo de mapa de cores para gráficos 3D e 4D;
- **set pointsize:** define o tamanho do ponto nos gráficos (na linha do comando plot pode ser redefinido pelo parâmetro "ps <n>");
- **set samples:** número de pontos a ser usado nas curvas do plot (para splot veja **set isosamples**);
- **set size:** define o tamanho do plot com relação a janela (multiplot);
- **set style:** define o estilo dos plots: **set style data** para arquivos; e **set style function** para expressões analíticas (entre outros...);
- **set terminal:** interface de saída (X11 e WXT: janela gráfica; jpeg, png, gif, postscript: arquivos);
- **set title:** título do gráfico;
- **set view:** ângulo de visão e dimensões para visualização de gráficos 3D (splot);
- **set xlabel:** texto do eixo x (idem para y e z);
- **set xrange:** intervalo do eixo x (idem para y e z);
- **set xtics:** numeração da escala do eixo x (idem para y e z);

Outros comandos...

Comandos importantes que não tivemos tempo de ver no curso

- **cbrange, cblabel, cbtics, etc:** controle dos parametros do **ColorBox**;
- **palette:** escala de cores para pm3d;
- **parametric:** para curvas paramétricas (e.g., $x^2+y^2=R^2$);
- **plot "dado.dat" index <n>:** grupos de dados podem ser separados em um arquivo por duas linhas em branco, o parâmetro **index** indica qual destes blocos será usado ($n \geq 0$) – é útil para fazer animações;
- **plot "dado.dat" every <...>:** semelhante ao **index**, mas permite o uso de mais de um grupo de dados simultaneamente;
- **coordinates:** definição do sistema de coordenadas usado em comandos que indicam uma posição (e.g., **set arrow** – veja abaixo);

Demonstrações

Na pasta “demo” vocês podem acessar vários scripts com exemplos de comandos do gnuplot.

Para acessá-los abra o gnuplot nesta pasta e use o comando “load” para carregar os arquivos com extensão .dem.

Note o exemplo all.dem. Este script executa todos os outros scripts da pasta, é interessante rodar este exemplo para ter uma visão geral.

Compilação do gnuplot

Não é difícil compilar o gnuplot mais novo a partir do código-fonte, mas é importante incluir algumas bibliotecas de desenvolvimento. No debian/ubuntu os pacotes que devem ser instalados são (pelo menos estas):

libreadline5-dev
libcairo2-dev
libgd2-xpm-dev
libpango1.0-dev

Quem tiver dificuldades para compilar pode me procurar.